

Федеральное агентство по образованию
Государственное образовательное учреждение высшего
профессионального образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ
(ТУСУР)**

Кафедра компьютерных систем в управлении и проектировании (КСУП)

В. М. Зюзьков, Н. Ю. Хабибулина, Е.А. Потапова

Основы алгоритмизации и программирование на языке Паскаль
(методические указания к лабораторным и самостоятельным работам
для студентов специальностей 220201 – Управление и информатика в
технических системах и 230104 -Системы автоматизированного
проектирования)

Учебно- методическое пособие

2007

Рецензент:

Зав. каф. Вычислительной математики и компьютерного моделирования
ТГУ, профессор, д. ф.-м. наук, Старченко А.В.

Зюзьков В. М., Хабибулина Н. Ю., Потапова Е.А.

Основы алгоритмизации и программирование на языке Паскаль (методические указания к лабораторным и самостоятельным работам для студентов специальностей 220201 – Управление и информатика в технических системах и 230104 -Системы автоматизированного проектирования): Учебно-методическое пособие / В.М. Зюзьков, Н.Ю. Хабибулина, Е.А. Потапова. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2007. – 166 с.

Пособие содержит требования к выполнению лабораторных работ, примеры написания программ, задания по дисциплине «Программирование и основы алгоритмизации» для специальности 220201 «Информатика и управление в технических системах» и «Программирование на языках высокого уровня» для специальности 230104 «Системы автоматизированного проектирования»

В качестве источника для получения теоретических знаний рекомендуется использовать лекции Зюзькова В. М. «Программирование на языке высокого уровня: Учебное пособие».

© Зюзьков В.М, Хабибулина Н.Ю.,

Потапова Е.А., 2007

© кафедра КСУП , ТУСУР, 2007

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА	5
ЛАБОРАТОРНАЯ РАБОТА 1(простые типы данных).....	6
Примеры решения некоторых задач.....	6
Задания на лабораторную работу 1	19
ЛАБОРАТОРНАЯ РАБОТА 2 (простейшие программы).....	29
Задания на лабораторную работу 2.	29
ЛАБОРАТОРНАЯ РАБОТА 3 (одномерные массивы, матрицы).....	43
Примеры решения некоторых задач.....	43
ОДНОМЕРНЫЕ МАССИВЫ	43
МНОГОМЕРНЫЕ МАССИВЫ	47
Задания на лабораторную работу 3	55
ЛАБОРАТОРНАЯ РАБОТА 4 (символьный и строковый типы)	68
Примеры решения некоторых задач.....	68
Задания на лабораторную работу 4	79
ЛАБОРАТОРНАЯ РАБОТА 5 (процедуры и функции)	84
Примеры решения некоторых задач.....	86
Задания на лабораторную работу 5	91
ЛАБОРАТОРНАЯ РАБОТА 6 (множества)	99
Примеры решения некоторых задач.....	100
Задания на лабораторную работу 6	104
ЛАБОРАТОРНАЯ РАБОТА 7 (сортировка)	110
Задания на лабораторную работу 7	112
ЛАБОРАТОРНАЯ РАБОТА 8 (рекурсия).....	116
Примеры решения некоторых задач.....	116
Задания на лабораторную работу 8	118
ЛАБОРАТОРНАЯ РАБОТА 9 (работа с файлами)	125
Примеры решения некоторых задач.....	125
Задания на лабораторную работу 9.	129
ЛАБОРАТОРНАЯ РАБОТА 10(динамические переменные)	135
Примеры решения некоторых задач.....	135
Задания на лабораторную работу 10	149
ЛАБОРАТОРНАЯ РАБОТА 11(графика).....	154
Примеры решения некоторых задач.....	154
Задания на лабораторную работу 11	160
ПРИЛОЖЕНИЕ А (форма титульного листа к лабораторной работе).....	165
ПРИЛОЖЕНИЕ Б (пример оформления содержания).....	166

ВВЕДЕНИЕ

Дисциплина "Программирование на языках высокого уровня" для студентов специальности 230104 – Системы автоматизированного проектирования изучается на протяжении двух семестров. В процессе обучения дисциплине в первом семестре студент должен выполнить четыре лабораторных работы. Первая, вторая и третья лабораторные работы содержат по четыре задачи, а четвертая лабораторная работа содержит две задачи. Каждое задание требует создания программы на языке Паскаль, предполагается, что каждый студент получит задание индивидуально. В методических указаниях к лабораторным работам приведены примеры решения некоторых задач, а так же задания на каждую лабораторную работу. Чтобы получить индивидуальное задание нужно обратиться к преподавателю

Во втором семестре студент должен выполнить семь лабораторных работ, каждая работа содержит по две задачи. Номера задач выдаются преподавателем индивидуально для каждого студента.

Дисциплина «Программирование и основы алгоритмизации» для студентов специальности 220201 – Управление и информатика в технических системах изучается на протяжении одного семестра. В процессе обучения дисциплине студент должен выполнить восемь лабораторных работ. Каждая работа содержит по две задачи. Номера задач выдаются преподавателем индивидуально для каждого студента.

К каждой лабораторной работе студент должен составить отчет. Требования к содержанию и оформлению отчета смотрите в соответствующем разделе методического пособия.

По мере написания и тестирования программ студент должен сдать их преподавателю, преподаватель может задать студенту вопросы по теме лабораторной работы.

Лабораторная работа считается сданной, если написаны и сданы преподавателю все предложенные задачи, а также отчет по лабораторной работе.

ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Отчет к лабораторной работе должен быть выполнен на бумаге формата А4 с одной стороны листа одним из следующих способов:

- Машинописным - через 1.5 – 2 межстрочных интервала. Шрифт основного текста Times New Roman, 12-14 пунктов. Шрифт листинга программ – Courier New 12-14 пунктов.
- Рукописным – четким разборчивым почерком, с высотой букв и цифр не менее 2.5 мм. Текст должен быть написан тушью, чернилами или пастой черного, синего, фиолетового цвета. Расстояние между строками 8-10 мм.

Отчет к лабораторной работе должен содержать:

- Титульный лист
- Содержание
- Введение
- Основная часть
- Заключение

Пример оформления титульного листа смотри в приложении.

Пример оформления содержания смотри в приложении.

Введение содержит краткое описание теоретического раздела, которому посвящена лабораторная работа.

Основная часть – для каждого задания содержит:

- Точную формулировку задания
- Описание алгоритма решения задачи
- Описание используемых переменных и обоснование выбора типа для всех переменных
- Блок – схему алгоритма
- Тестирование программы.
- Текст программы с комментариями

ЛАБОРАТОРНАЯ РАБОТА 1(простые типы данных)

Для решения задач этой лабораторной работы следует использовать простые типы данных паскаля, такие как `integer`, `real`, `boolean`, `char`. Не допускается использование массивов для хранения данных. Если по условию задачи дана последовательность чисел, то рекомендуется использовать пошаговый ввод данных, то есть на каждом шаге цикла вводится новое значение, которое обрабатывается в теле цикла (смотрите примеры выполнения некоторых задач).

Примеры решения некоторых задач.

1. Дано натуральное N число. Разложить его на простые множители.

```
var i, n, f, j : integer;
```

```
begin
```

```
  repeat
```

```
    write('Введите натуральное число N= ');
```

```
    readln(n);
```

```
  until n>0;
```

```
  write (N:6, '=1');
```

```
  f:=0; j:=n;
```

```
  for i:=2 to n div 2 do
```

```
    begin
```

```
      if j mod i = 0 then begin
```

```
        f:=1;
```

{цикл определяют, сколько таких множителей i в

нашем числе n}

```
      while j mod i=0 do
```

```
        begin
```

```
          write('*', i);
```

```

        j:=j div i;
    end;
        {f определяет, были ли найдены простые множители, которые
больше единицы}
        if f=0 then writeln('*', n);
        end;
    end;
    writeln
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим натуральное число n ;
- в цикле по переменной i начинаем порождение натуральных чисел, не превосходящих $n/2$;
- если такое число является делителем, то определяем, сколько таких делителей, уменьшая число n ;
- в зависимости от значения флага f выводим результат.

Переменные:

- n - исследуемое число;
- i, j - переменные циклов;
- f - вспомогательный флаг.

2. Дано натуральное число N . Определить, является ли оно автоморфным.

Аutomorphic number n is equal to the last digits of the square of this number: $5 \leftrightarrow 25$

$6 \leftrightarrow 36$ $25 \leftrightarrow 625$

```
var n, m, r : integer;
```

```
begin
```

```
    repeat
```

```

write('Введите натуральное число n=');
readln(n);
until n>0;
m:=n; r:=1;
while m>0 do
begin
    m:=m div 10;
    r:=r*10;
end;
if (n*n mod r) = n then writeln('Число ',n,' автоморфно')
    else writeln('Число ',n,' не автоморфно')
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим натуральное число n;
- определяем, сколько в нем разрядов, и формируем r;
- отделяем нужное количество разрядов от $n*n$ с помощью функции $n*n \bmod r$;
- выводим результат.

Переменные:

n - исследуемое число;

m, r - вспомогательные переменные.

3. Дано натуральное число n. Определить, является ли оно палиндромом.

Число-палиндром можно читать справа налево и слева направо: 4 88 121 767767

и т.д.

```
var n, r, m, k, i : integer;
```

```
    f : boolean;
```

```
begin
```

```
    repeat
```

```

write('Введите натуральное число n=');
readln(n);
until n>0;
m:=n; r:=1; k:=1; f:=false;
while (m div 10)>0 do
begin
    m:=m div 10;
    r:=r*10;
    k:=k+1;
end;
m:=n;
for i:=1 to k div 2 do
begin
    if (m div r) <> (m mod 10) then f:=true;
    m:=(m mod r) div 10;
    r:=r div 100
end;
if f then writeln('Число ',n,'не палиндром')
    else writeln('Число ',n,'палиндром')
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим натуральное число n;
- определяем, сколько в нем разрядов, и формируем r;
- в цикле по i отделяем по одному разряду справа и слева и сравниваем их, отделяя рассмотренные разряды;
- выводим результат.

Переменные:

n - исследуемое число;

10

m, r - вспомогательные переменные.

i - переменная цикла;

f - вспомогательный флаг;

k - количество разрядов.

4. Вычислить: $y = (2n-1)!! = 1*3*5* \dots *(2n-1)$, $n>0$;

```
var y : real;
```

```
  i, n : integer;
```

```
begin
```

```
  writeln('Введите размер n-?');
```

```
  readln(n);
```

```
  y:=1;
```

```
  for i:=1 to n do y:=y*(2*i-1);
```

```
  writeln('Полученное значение y= ', y)
```

```
end.
```

5. Вычислить: $y = \sin 1 + \sin 1.1 + \sin 1.2 + \dots + \sin 2$.

```
var y : real;
```

```
  i : integer;
```

```
begin
```

```
  writeln('Полученное значение расчета формулы y=sin1+sin1.1+sin1.2+ ...  
+sin2 = ');
```

```
  y:=0;
```

```
  for i:=0 to 10 do y:=y+sin(1+0.1*i);
```

```
  writeln(y);
```

```
end.
```

6. Если среди чисел $\sin(x^n)$ ($n = 1, 2, \dots, 30$) есть хотя бы одно отрицательное число, то логической переменной t присвоить значение true, а иначе - значение false.

а) используя оператор цикла с параметром и оператор перехода.

```

var y,x : real;
    n : integer;
    t : boolean;
label 15;
begin
    write('Введите значение x -');
    readln(x);
    t:=true; y:=1;           {y = x^n = x * x^{n-1}}
    for n:=1 to 30 do
        begin
            y:=x*y;
            if sin(y)<0 then goto 15
        end;
        t:=false;
    15:writeln(t);
end.

```

б) Не использовать оператор перехода.

```

var y,x : real;
    n : integer;
    t : boolean;
begin
    write('Введите значение x -');
    readln(x);
    y:=1; n:=0;
    repeat
        n:=n+1; y:=x*y; t:=sin(y)<0
    until t or (n=30);
    writeln(t);
end.

```

12

end.

7. Определить k - количество трехзначных натуральных чисел, сумма цифр которых равна n ($1 < n < 27$). Операции деления ($/$, div и mod) не использовать.

```
var d1, d2, d3, k, n : integer;
```

```
begin
```

```
  writeln('Введите число n, с которым будем сравнивать сумму цифр числа');
```

```
  readln(n);
```

```
  k:=0;
```

```
                                {d1 - левая, d2 - средняя, d3 -
```

```
правая цифры числа}
```

```
  for d1:=1 to 9 do
```

```
    for d2:=0 to 9 do
```

```
      for d3:=0 to 9 do
```

```
        if d1+d2+d3=n then begin
```

```
          k:=k+1; write(d1,d2,d3, ' ');
```

```
        end;
```

```
      writeln('Количество искомых чисел равно -', k);
```

```
end.
```

8. Найти сумму первых n членов ряда $y = 1 + x/2 + x^2/3 + x^3/4 + \dots$ при $|x| < 1$.

```
var x, y, z : real;
```

```
  n, i : integer;
```

```
begin
```

```
  repeat
```

```
    writeln('Введите переменную ряда x, |x|<1 x=');
```

```
    readln(x);
```

```
    write('Введите число членов ряда n=');
```

```
    readln(n);
```

```
  until (abs(x)<1) and (n>0);
```

```

y:=1; z:=1;
for i:=2 to n do
begin
    z:=z*x;
    y:=y+z/i;
end;
writeln('Сумма первых n членов ряда у =', y);
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим количество членов ряда n и переменную X;
- в цикле порождаем очередной член ряда и прибавляем его к сумме у;
- выводим результат.

Переменные:

- n - количество членов ряда;
- x - переменная ряда;
- z - вспомогательная переменная;
- i - переменная цикла;
- y - сумма ряда.

9. Вводится последовательность из N целых чисел. Найти сумму всех отрицательных чисел.

```

var n, x, sum, i : integer;
begin
    repeat
        write('Введите длину последовательности n=');
        readln(n);
    until n>0;
    sum:=0;

```

14

```
for i:=1 to n do
begin
    write('Введите x=');
    readln(x);
    if x<0 then sum:=sum+x;
end;
if sum=0 then writeln('Отрицательных чисел нет')
    else writeln('Сумма отрицательных чисел sum= ', sum);
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим длину последовательности n и устанавливаем начальное значение sum;
- последовательно считываем числа, и если число отрицательное, то прибавляем его к сумме sum;
- в зависимости от значения sum выводим результат.

Переменные:

- n - количество чисел;
- x - очередное число;
- i - переменная цикла;
- sum - сумма отрицательных чисел.

10. Вводится последовательность из N целых чисел. Найти наибольшее число.

```
var n, x, max, i : integer;
begin
    repeat
        write('Введите длину последовательности n=');
        readln(n);
    until n>0;
    write('Введите x=');
```

```

readln(x);
max:=x;
for i:=2 to n do
begin
  write('Введите x=');
  readln(x);
  if (x>max) then max:=x;
end;
writeln('Наибольшее из чисел max=', max);
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим длину последовательности n и устанавливаем начальное значение max по первому числу;
- последовательно считываем числа, и если очередное число x больше max, то переприсваиваем значение max := x;
- выводим результат.

Переменные:

- n - количество чисел;
- x - очередное число;
- i - переменная цикла;
- max - наибольшее число.

11. Вводится последовательность целых чисел, 0 -конец последовательности.

Найти два наименьших числа.

```

var x,min1,min2:integer;
begin
  write('Введите x=');
  readln(x);

```

```

min1:=x;
write('Введите x=');
readln(x);
min2:=x ;
                                                    {min1>=min2}

repeat
  if x<=min1 then
    begin
      min2:=min1;
      min1:=x;
    end
  else
    if (min1<x) and (x<min2) then min2:=x;
  write('Введите x=');
  readln(x);
until (x=0);
writeln( 'Два наименьших числа равны', min1, 'и', min2);
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- устанавливаем начальные значения $min1$ и $min2$ по двум первым числам;
- последовательно считываем числа и, если очередное число x меньше или равно $min1$ ($min1 < min2$), то переприсваиваем значение $min1$ и $min2$;
- если x попадает в интервал от $min1$ до $min2$, то переприсваиваем только $min2$;
- выводим результат.

Переменные:

x - очередное число;

$min1$ - первое наименьшее число;

$min2$ - второе наименьшее число ($min2 \geq min1$).

12. Вводится последовательность ненулевых чисел, 0 -конец последовательности. Определить, является ли последовательность возрастающей.

Решение данной задачи строится от противного. Математически для того, чтобы последовательность была возрастающей, для каждого очередного элемента `new` и предыдущего `old` должно выполняться условие `new > old`. Любое нарушение данного условия приводит к тому, что последовательность не может быть возрастающей.

```
var old, new : real;
    f : boolean;
begin
    write('Введите x=');
    readln(old);
    write('Введите x=');
    readln(new);
    f:=true;
    repeat
        if new<=old then f:=false;
        old:=new;
        write('Введите x=');
        readln(new);
    until new=0;
    if f then writeln( 'Последовательность возрастающая')
        else writeln( 'Последовательность не является возрастающей');
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим два первых числа как `old` и `new`, задаем начальное значение флага `f`;

- в цикле ищем нарушение свойства членов возрастающей последовательности;
- переписываем значение `old:=new` и вводим новое - `new`;
- в зависимости от флага выводим результат.

Переменные:

`old` - предыдущее число;

`new` - рассматриваемое число;

`f` - флаг.

13. В очереди за билетами стоят мужчины и женщины. Какое количество мужчин стоит в начале очереди.

```
var pol : char;
```

```
    i,k,n : integer;
```

```
begin
```

```
    repeat
```

```
        write('Введите длину последовательности n=');
```

```
        readln(n);
```

```
    until n>0;
```

```
    k:=0;
```

```
    for i:=1 to n do
```

```
        begin
```

```
            writeln('Введите пол');
```

```
            readln(pol);
```

```
            if pol='m' then k:=k+1
```

```
                else break
```

```
        end;
```

```
    if k=0 then writeln('Мужчин в начале очереди нет')
```

```
        else writeln('В начале очереди стоит', k, 'мужчин')
```

```
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим длину последовательности, задаем начальное значение счетчика k ;
- в цикле вводим пол человека;
- если это мужчина, то увеличиваем значение счетчика на единицу;
- в зависимости от k выводим результат.

Переменные:

k - счетчик;

i - переменная цикла;

n - количество человека в очереди;

rol - пол человека.

Задания на лабораторную работу 1

1. Дано: натуральное n , действительные a_1, a_2, \dots, a_n . Получить: $a_1 * a_2 * a_3 * \dots * a_k$, где $k = n$, если n -нечетно, и $k = n-1$, если n -четно.

2. Возьмите два натуральных числа a и b . Образуйте последовательность, состоящую из n членов:

$$a = x_1, b = x_2, x_3 = x_1 + x_2, \dots, x_i = x_{i-1} + x_{i-2}, \dots, x_n = x_{n-1} + x_{n-2}.$$

Вычислите величину x_{n-1} / x_n .

(Должно получиться $0.618033 = (\sqrt{5} - 1) / 2$ - "золотое сечение").

3. Дано: натуральное n , действительные a_1, a_2, \dots, a_n . Получить сумму тех членов a_i данной последовательности, которые кратны 5.

4. Напечатать те числа последовательности от 1 до 40, которые не делятся на 5 и на 3.

5. Логической переменной t присвоить значение `true` или `false` в зависимости от того, является натуральное число k степенью 3 или нет.

6. Дано: натуральное n , действительные a_1, a_2, \dots, a_n . Найти среднее арифметическое всех положительных членов последовательности a_i ($i=1,2,\dots,n$).

7. Дано: натуральное n , действительные a_1, a_2, \dots, a_n .

Получить: $a_1 * a_2 + a_2 * a_3 + \dots + a_{n-1} * a_n$ ($n > 1$).

8. Даны натуральные n, k_1, k_2, \dots, k_n . Найти те члены k_i последовательности k_1, k_2, \dots, k_n , которые при делении на 9 дают остаток 0, 5 или 7.

9. Дано: натуральное n , действительные a_1, a_2, \dots, a_n .

Получить: $a_1 + 2 * a_2 + 2 * a_3 + \dots + 2 * a_{n-1} + a_n$.

10. Дано: натуральное n , действительное число a . Вычислить: $a * (a - n) * (a - 2n) * \dots * (a - n^2)$.

11. Вычислить k - количество точек с целочисленными координатами, попадающих в круг радиуса R ($R > 0$) с центром в начале координат.

12. Даны натуральные n, k_1, k_2, \dots, k_n . Найти те члены k_i последовательности k_1, k_2, \dots, k_n , которые являются удвоенными нечетными.

13. Дано: натуральное n , действительные a_1, a_2, \dots, a_n . Получить сумму всех тех членов a_i данной последовательности, которые нечетны и отрицательны.

14. Дано: натуральное n , действительные a_1, a_2, \dots, a_n . Получить: $n + a_n * a_1$.

15. Дано: натуральное n , действительные a_1, a_2, \dots, a_n .

Получить: $a_1,$

$a_1 * a_2,$

$a_1 * a_2 * a_3,$

.....

$a_1 * a_2 * a_3 * \dots * a_n.$

16. Пусть n -натуральное число. Обозначим через $n!!$ произведение $1 * 3 * \dots * n$ для нечетного n и $2 * 4 * 6 * \dots * n$ для четного n . Дано натуральное n . Получить $n!!$.

17. Дано: натуральное n , действительные a_1, a_2, \dots, a_n . Получить сумму тех членов a_i данной последовательности, которые удовлетворяют условию $|a_i| < i^2$.

18. Даны натуральные n , k_1, k_2, \dots, k_n . Найти те члены k_i последовательности k_1, k_2, \dots, k_n , которые при делении на 7 дают остаток 1, 2 или 5.

19. Дана непустая последовательность целых чисел, за которой следует 0 (это признак конца последовательности). Определить, сколько среди них отрицательных.

20. Дано натуральное число n . Вычислить произведение первых n сомножителей

$$1/2 * 3/4 * 5/6 * \dots .$$

21. Дано натуральное число n . Вычислить произведение первых n сомножителей

$$1/1 * 3/2 * 5/3 * \dots .$$

22. Даны действительные числа x, a , натуральное n . Вычислить:

$$((\dots((x+a)^2 + a)^2 + \dots + a)^2 + a$$

└───┘

n скобок

23. Дано действительное число x . Вычислить:

$$\frac{(x-2) * (x-4) * (x-8) * \dots * (x-64)}{(x-1) * (x-3) * (x-7) * \dots * (x-63)}$$

24. Дано: натуральное n , действительные a_1, a_2, \dots, a_n .

Получить: $|a_1^2 - a_n^2|$.

25. Даны действительное число a , натуральное число n . Вычислить:

$$\frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \dots + \frac{1}{a^{2^n}}$$

26. Вычислить произведение:

$$\prod_{i=1}^{10} \left(1 - \frac{1}{i!}\right)^2$$

27. Вычислить произведение:

$$\prod_{k=1}^n \cos(k)$$

28. Вычислить произведение:

$$\prod_{i=1}^n \left(\frac{i}{i+1} * \frac{i+2}{i+3} \right)$$

29. Вычислить произведение:

$$\prod_{k=2}^n (\ln(k))^2$$

30. Вычислить произведение:

$$\prod_{i=1}^{10} \left(2 + \frac{i}{i!} \right)$$

31. Вычислить сумму:

$$\sum_{i=1}^n \frac{x^i}{i!}$$

для данных натурального n и действительного числа x .

32. Вычислить сумму:

$$\sum_{k=1}^n \left(\frac{1}{k} - \frac{1}{k+1} \right)$$

33. Вычислить сумму:

$$\sum_{k=1}^n \left(\frac{(-1)^{k+1}}{k * (k+1)} \right)$$

34. Вычислить сумму:

$$\sum_{i=1}^n \frac{i}{i^2}$$

35. Вычислить сумму:

$$\sum_{i=1}^n \frac{i}{i^5}$$

36. Вычислить сумму:

$$\sum_{i=1}^n \frac{i}{(2*i)^2}$$

37. Вычислить сумму:

$$\sum_{i=1}^n \frac{i^2}{i^2 + 2*i + 3}$$

38. Вычислить сумму:

$$\sum_{k=1}^n \left(\frac{(-1)^k}{(2*k+1)*k} \right)$$

39. Вычислить сумму:

$$\sum_{k=1}^n \left(\frac{(-1)^k}{(2*k+1)^2} \right)$$

40. Даны действительное число x , натуральное число n . Вычислить сумму:

$$\sum_{i=1}^n \left(\frac{1}{i!} + \sqrt{|x|} \right)$$

41. Даны натуральное число n , действительные числа x_1, x_2, \dots, x_n ($n \geq 3$).

Вычислить:

$$(x_1 + x_2 + x_3) * x_2 + (x_2 + x_3 + x_4) * x_3 + \dots + (x_{n-2} + x_{n-1} + x_n) * x_{n-1}.$$

42. Даны натуральное n , целые числа a_1, a_2, \dots, a_n . Заменить каждый из членов остатком от деления его квадрата на n .

43. Даны натуральные числа n, a_1, a_2, \dots, a_n . Определить количество членов a_k последовательности a_1, \dots, a_n кратных 3 и не кратных 5.

44. По длинам двух сторон некоторого треугольника и углу (в градусах) между ними найти длину третьей стороны и площадь этого треугольника.

45. Дано не менее трех различных натуральных чисел, за которыми следует 0. Определить три наибольших числа среди них.

46. Определить число, получаемое выписыванием в обратном порядке цифр заданного натурального числа.

47. Даны натуральные числа n, m . Получить сумму m последних цифр числа n .

48. Даны действительные положительные числа a, b, c, x, y . Выяснить, пройдет ли кирпич с ребрами a, b, c в прямоугольное отверстие со сторонами x, y . Просовывать кирпич в отверстие разрешается так, чтобы каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия.

49. Даны натуральные числа n, a_1, a_2, \dots, a_n . Определить количество членов a_k последовательности a_1, \dots, a_n , являющихся квадратами чисел.
50. Составьте программу, которая в записи натурального числа N меняет местами крайние цифры, т.е. из 451 получает 154 или из 1934 получает 4931.
51. Даны натуральные числа n, a_1, a_2, \dots, a_n . Определить количество членов a_k последовательности a_1, \dots, a_n , являющихся нечетными числами.
52. Найти сумму цифр заданного натурального числа.
53. Даны натуральные числа i, n , действительные числа a_1, \dots, a_n ($i \leq n$).
Найти среднее арифметическое чисел a_1, \dots, a_n , кроме a_i .
54. Дано не менее двух различных натуральных чисел, за которыми следует 0.
Определить два наименьших числа среди них.
55. Даны натуральные числа n, a_1, a_2, \dots, a_n . Определить количество членов a_k последовательности a_1, \dots, a_n , имеющих четные порядковые номера и являющихся нечетными числами.
56. Определить, является ли данное n ($n > 1$) составным числом.
57. Даны натуральное n , действительные положительные числа c_1, c_2, \dots, c_n .
Значения c_1, \dots, c_n являются емкостями n конденсаторов. Определить емкости систем конденсаторов, которые получаются последовательным и параллельным соединением исходных конденсаторов.
58. Составьте программу, которая к записи целого числа $N < 1000$ добавляет 1 справа и слева, т.е. из числа 924 получается 19241.
59. Дано действительное $b > 0$. Последовательность a_1, a_2, \dots образована по следующему закону: $a_1 = b$,
- $$a_i = a_{i-1} - \frac{1}{\sqrt{i}} \quad i = 2, 3, \dots$$
- Найти первый отрицательный член последовательности a_1, a_2, \dots .
60. Дано натуральное число n . Вычислить:
- $$1 * 2 + 2 * 3 * 4 + \dots + n * (n + 1) * \dots * 2n.$$

61. Дано действительное $b < 0$. Последовательность a_1, a_2, \dots образована по следующему закону: $a_1 = b$,

$$a_i = \frac{a_{i-1} + 1}{i - \sin^2(i)} \quad i = 2, 3, \dots$$

Найти первый неотрицательный член последовательности a_1, a_2, \dots

62. Программа. Даны натуральное число n и вещественные числа t, a_0, a_1, \dots, a_n . Вычислить значение многочлена

$$a_0 * x^n + a_1 * x^{n-1} + \dots + a_{n-1} * x + a_n$$

и его производной в точке $t^2 + 0.5$.

63. Даны натуральное число n , действительные числа x_1, x_2, \dots, x_n

($n \geq 3$). Вычислить:

$$(x_1 + 2 * x_2 + x_3) * (x_2 + 2 * x_3 + x_4) * \dots * (x_{n-2} + 2 * x_{n-1} + x_n).$$

64. Получить таблицу температур по Цельсию от 0 до 100 градусов и их эквивалентов по шкале Фаренгейта, используя для перевода формулу

$$t_f = 9/5 * t_c + 32.$$

65. Вычислить y - первое из чисел $\sin(x), \sin(\sin(x)), \sin(\sin(\sin(x))), \dots$, меньшее по модулю 10^4 .

66. Вычислить бесконечную сумму с заданной точностью E ($E > 0$ и $E < 1$).

$$\sum_{i=0}^{\infty} \frac{(-2)^i}{i!}$$

Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем E , - это и все последующие слагаемые можно уже не учитывать.

67. Вычислить бесконечную сумму с заданной точностью E ($E > 0$ и $E < 1$).

$$\sum_{i=0}^{\infty} \frac{(-1)^i}{i!}$$

Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем E , - это и все последующие слагаемые можно уже не учитывать.

68. Вычислить бесконечную сумму с заданной точностью E ($E > 0$ и $E < 1$).

$$\sum_{i=1}^{\infty} \frac{1}{i^2}$$

Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем E , - это и все последующие слагаемые можно уже не учитывать.

69. Дано действительное $b > 0$. Последовательность a_1, a_2, \dots образована по следующему закону: $a_1 = 1$,

$$a_i = a_{i-1}^2 + 1 \quad i = 2, 3, \dots$$

Требуется получить все a_1, a_2, \dots , меньшие или равные b .

70. Вычислить приближенно значение бесконечной суммы

$$\frac{1}{1*2*3} + \frac{1}{2*3*4} + \frac{1}{3*4*5} + \dots$$

Нужное приближение считается полученным, если очередное слагаемое оказалось меньше данного положительного числа E ($0 < E < 1$).

71. Пусть $a_0 = a_1 = 1$;

$$a_i = a_{i-2} + \frac{a_{i-1}}{2^{i-1}} \quad i = 2, 3, \dots$$

Найти произведение $a_0 * a_1 * \dots * a_{14}$.

72. Пусть $v_1 = v_2 = 0$; $v_3 = 1.5$;

$$v_i = \frac{i+1}{i^2+1} * v_{i-1} - v_{i-2} * v_{i-3} \quad i = 4, 5, \dots$$

Дано натуральное $n \geq 4$. Получить v_n .

73. Даны целые числа n, k ($n \geq k \geq 0$). Вычислить:

$$\frac{n * (n-1) * \dots * (n-k+1)}{k!}$$

$k!$

74. Числа Фибонначи: $F_0 = 0$, $F_1 = 1$, а любое следующее число Фибонначи равно сумме двух предыдущих: $F_n = F_{n-1} + F_{n-2}$. Известно, что при достаточно больших n справедливо приближенно равенство

$$F_n = \frac{1}{\sqrt{5}} * \left(\frac{1 + \sqrt{5}}{2} \right)^n$$

Определите наименьший номер n , начиная с которого равенство выполняется с точностью до заданного ϵ ($0 < \epsilon < 1$).

75. Пусть $x_1 = x_2 = x_3 = 1$; $x_i = x_{i-1} + x_{i-3}$; $i = 4, 5, \dots$ Найти:

$$\sum_{i=1}^{100} \frac{x_i}{2^i}$$

76. Дано натуральное n , действительное x . Вычислить:

$$\sin(x) + \sin(\sin(x)) + \dots + \sin(\sin(\dots \sin(x) \dots))$$

$$\underbrace{\hspace{10em}}_{n \text{ раз}}$$

77. Дано натуральное n . Вычислить:

$$\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \sin 2 + \dots + \sin n}$$

78. Пусть $x_1 = y_1 = 1$; $x_i = 0.3 * x_{i-1}$; $y_i = x_{i-1} + y_{i-1}$; $i = 2, 3, \dots$ Дано натуральное n . Найти:

$$\sum_{i=1}^n \frac{x_i}{1 + |y_i|}$$

79. Даны натуральное число n , действительные числа a_1, \dots, a_n . Получить b_1, b_2, \dots, b_n , где

$$b_i = \frac{a_i}{1 + (a_1 + a_2 + \dots + a_i)^2}, \quad i = 1, \dots, n.$$

80. Дано натуральное n , действительное a . Вычислить:

$$\frac{1}{a} + \frac{1}{a * (a+1)} + \dots + \frac{1}{a * (a+1) * \dots * (a+n)}$$

81. Пусть $x_0 = 1$;

$$x_k = \frac{2 - x_{k-1}^3}{5} \quad k = 1, 2, \dots$$

Найти первый член x_n , для которого $|x_n - x_{n-1}| < 10^{-5}$.

82. Вычислить:

$$(1 + \sin(0.1)) * (1 + \sin(0.2)) * \dots * (1 + \sin(10)).$$

83. Пусть $a_1 = b_1 = 1$;

$$a_k = \frac{1}{2} * \left(\sqrt{b_{k-1}} + \frac{1}{2} * \sqrt{a_{k-1}} \right) \quad k = 2, 3, \dots$$

$$b_k = 2 * a_{k-1}^2 + b_{k-1}$$

Дано натуральное n . Найти:

$$\sum_{k=1}^n a_k * b_k$$

84. Написать программу для решения указанной задачи: для заданного числа a найти корень уравнения $f(x)=0$, где

$$f(x) = \begin{cases} 2 * a * x + |a - 1|, \text{ при } a > 0, \\ \frac{e^x}{\sqrt{1+a^2}} - 1 - \text{иначе} \end{cases}$$

85. Пусть

$$a_1 = 1 + 1/2; \quad a_2 = 1 + \frac{1}{1 + \frac{1}{2}}; \quad a_3 = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}; \quad ;$$

$$a_n = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{\dots}}}$$

$$1 + \frac{1}{2}$$

той же прямой, определить концы отрезка, являющегося пересечением всех этих отрезков. Если такого отрезка нет, сообщить об этом.

2. Недетерминированный однозначный алгоритм Чудо-дерево, на котором растут апельсины и бананы, обладающие следующим свойством - если с него срывают два одинаковых плода, то тут вырастает один апельсин, а если два разных, то один банан.

Программа. Сбирать урожай, пока на дереве не останется один плод. Какой это плод? Начальное количество плодов на Чудо-дерево задается случайно. Сбор урожая тоже происходит случайным образом.

3. Определить, является ли заданное натуральное число совершенным, т.е. равным сумме всех своих (положительных) делителей, кроме самого этого числа (например, число 6 совершенно: $6=1+2+3$).
4. Дано n вещественных чисел. Определите сколько из них больше своих "соседей", т.е. предыдущего и последующего чисел.
5. Дано натуральное число n . Выяснить, имеются ли среди $n, n+1, \dots, 2n$ близнецы, т.е. простые числа, разность между которыми равна двум.
6. Дана последовательность из не менее чем двух натуральных чисел, за которой следует 0. Вычислите сумму тех из них, порядковые номера которых - простые числа.
7. Программа. Дано натуральное k . Напечатать k -ую цифру последовательности 149162536..., в которой выписаны подряд квадраты всех натуральных чисел.
8. Программа. Дано натуральное k . Напечатать k -ую цифру последовательности 1123581321... , в которой выписаны подряд все числа Фибонначи.
9. Дана непустая последовательность ненулевых целых чисел, за которой следует 0. Определить, сколько раз в этой последовательности меняется знак (например, в последовательности 1, -34, 8, 14, -5 знак меняется 3 раза).
10. Напечатать в возрастающем порядке трехзначные числа, в десятичной записи которых нет одинаковых цифр (операции деления ($/$, div и mod) не использовать).

11. Даны натуральное n и последовательность a_1, a_2, \dots, a_n .
Определить, со скольких отрицательных чисел она начинается.
12. Даны целое $n > 0$ и целые числа a_1, a_2, \dots, a_n . Определить: сколько чисел последовательности a_1, a_2, \dots, a_n принимает наибольшее значение.
13. Дана последовательность из n целых чисел. Определите количество чисел в наиболее длинной последовательности из подряд идущих нулей.
14. Дана непустая последовательность различных натуральных чисел, за которой следует 0. Определить порядковый номер наименьшего из них.
15. Найти 100 первых простых чисел.
16. Даны целое $n > 0$ и последовательность из n вещественных чисел.
Вычислить разность между максимальным и минимальным из них.
17. Задано натуральное число n , напечатать его цифры в r -ичной системе счисления.
18. Дано целое k от 1 до 100. Определить, какая цифра находится в k -й позиции последовательности 10111213...9899, в которой выписаны подряд все двузначные числа.
19. Дано натуральное k . Напечатать k -ю цифру последовательности 1234567891011121314..., в которой выписаны подряд все натуральные числа.
20. Определить число, получаемое выписыванием в обратном порядке цифр заданного натурального числа.
21. Даны натуральное n и последовательность a_1, a_2, \dots, a_n вещественных чисел. Определить, образуют ли они возрастающую последовательность.
22. Дано натуральное число n . Выбросить из записи числа n цифры 0 и 5, оставив прежними порядок остальных цифр. Например, из числа 59015509 должно получиться 919.
23. Найти сумму цифр заданного натурального числа.
24. Даны целое $n > 0$ и вещественные числа x_1, x_2, \dots, x_n . Определить, сколько из них больше своего предыдущего числа.

25. Дано n вещественных чисел. Найти порядковый номер того из них, которое наиболее близко к квадрату какого-нибудь целого числа.
26. Даны натуральные числа a, b ($a \leq b$). Получить все простые числа p , удовлетворяющие условию $a \leq p \leq b$.
27. Даны натуральное n и вещественные числа $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Рассматривая пары x_i, y_i как координаты точек на плоскости, определить радиус наименьшего круга (с центром в начале координат), внутрь которого попадают все эти точки.
28. Даны целое $n > 0$ и последовательность из n вещественных чисел, среди которых есть хотя бы одно отрицательное число. Найти величину наибольшего среди отрицательных чисел последовательности.
29. Даны натуральные числа n, m . Найти наименьшее общее кратное n и m . (Как здесь может помочь алгоритм Евклида?)
30. Программа. Дано натуральное k . Определить k -ую цифру в последовательности $110100100010000100000\dots$, в которой выписаны подряд степени числа 10.
31. Не используя стандартные функции, вычислить с точностью ϵ ($0 < \epsilon < 1$)

$$y = \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{(-1)^n * x^{2n}}{(2n)!} + \dots$$

Считать, что требуемая точность достигнута, если очередное слагаемое по модулю меньше ϵ , - все последующие слагаемые можно уже не учитывать.

32. Дана непустая последовательность положительных вещественных чисел x_1, x_2, \dots, x_n (n заранее не известно), за которыми следует отрицательное число. Вычислить величину:

$$n * x_1 + (n-1) * x_2 + \dots + 2 * x_{n-1} + x_n$$

33. Найти все целые корни уравнения $ax^3 + bx^2 + cx + d = 0$, где a, b, c, d - заданные целые числа, причем $a \neq 0$ и $d \neq 0$. (Замечание: целыми корнями

могут быть только положительные и отрицательные делители коэффициента d).

34. Числа Фибонначи (f_n) определяются формулами $f_0=f_1=1$; $f_n = f_{n-1} + f_{n-2}$ при $n=2, 3, \dots$. Найти f - первое число Фибонначи, большее m ($m>1$).

35. Для любого целого k обозначим количество цифр в его десятичной записи через $N(k)$. Дано натуральное число m . Вычислить:

$$\frac{N(1)}{1^2} + \frac{N(2)}{2^2} + \dots + \frac{N(m)}{m^2}$$

36. Не используя стандартные функции, вычислить с точностью ϵ ($0<\epsilon<1$)

$$y = \arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + \frac{(-1)^n * x^{(2n+1)}}{2n+1} + \dots$$

Равенство справедливо при $|x|<1$.

Считать, что требуемая точность достигнута, если очередное слагаемое по модулю меньше ϵ , - все последующие слагаемые можно уже не учитывать.

37. Числа Фибонначи (f_n) определяются формулами $f_0 = f_1 = 1$; $f_n = f_{n-1} + f_{n-2}$ при $n=2, 3, \dots$. Вычислить сумму всех чисел Фибонначи, которые не превосходят 1000.

38. Не используя стандартные функции (за исключением abs), вычислить с точностью ϵ ($0<\epsilon<1$), $x<1$

$$y = \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + \frac{(-1)^{(n-1)}}{n} + \dots$$

Считать, что требуемая точность достигнута, если очередное слагаемое по модулю меньше ϵ , - все последующие слагаемые можно уже не учитывать.

39. Программа. Напечатать таблицу значений функций $\sin(x)$ и $\cos(x)$ на отрезке $[0,1]$ с шагом 0.1 в следующем виде (считать, что при печати на каждое вещественное число отводится по 6 позиций строки):

x	sin(x)	cos(x)
.....		
0.0000	0.0000	1.0000
0.1000	0.0998	0.9950
.....		
1.0000	0.8415	0.5403

40. Вычислить

$$\sum_{k=1}^n \frac{k!}{1/2 + 1/3 + \dots + 1/(k+1)}$$

41. Даны целое $n > 1$ и вещественные числа x_1, x_2, \dots, x_n .

Вычислить:

$$M = \frac{\sum x[i]}{n}, \quad D = \sqrt{\frac{\sum (x[i] - m)^2}{n - 1}}$$

42. В четырехзначном числе все цифры разные и отличные от 0. Если его написать в обратном порядке получится число, на 8532 меньше первоначального. Найдите это число.

43. Дано натуральное n . Можно ли n представить в виде суммы трех квадратов натуральных чисел? Если можно, то указать все тройки x, y, z таких натуральных чисел, что $x^2 + y^2 + z^2 = n$.

44. Разгадайте ребус:

	ПРИМЕР
	РИМЕР
+	ИМЕР
	МЕР
	ЕР
	Р

ЗАДАЧА

45. Напечатать все трехзначные десятичные числа, сумма цифр которых равна данному числу.

46. В выражение $((((1 ? 2) ? 3) ? 4) ? 5)$ вместо каждого знака "?" вставить знак одного из четырех арифметических действий: $+$, $-$, $*$, \div так, чтобы результат вычислений равнялся 35.

47. Составьте программу, решающую числовой ребус:

$$\begin{array}{r} \text{МУХА} \\ + \text{МУХА} \\ + \text{МУХА} \\ \hline \end{array}$$

(Сократите число переборов)

СЛОН

48. Даны натуральные n , m . Получить все меньшие n натуральные числа, сумма цифр которых равна m .

49. Дано: натуральное n , действительные a_1, a_2, \dots, a_n . Вычислить: $a_1^n + a_2^n + \dots + a_n^n$.

50. Назовем натуральное число палиндромом, если его запись читается одинаково с начала и с конца (как, например, 4884, 393, 1). Найти все меньшие 100 натуральные числа, которые при возведении в квадрат дают палиндром. (Введите функцию, распознающую является ли данное число палиндромом).

51. Дано натуральное число n . Среди чисел $1, 2, \dots, n$ найти все те, которые можно представить в виде суммы квадратов двух натуральных чисел. (Определить функцию, позволяющую распознавать полные квадраты).

52. Даны координаты вершин двух треугольников. Определить, какой из них имеет большую площадь.

53. Для данного n найти все пифагоровские тройки чисел, меньшие n , т.е. найти такие натуральные a, b, c , что $a^2 + b^2 = c^2$ и $a, b, c < n$.

54. Два натуральных числа называются "дружественными", если каждое из них равно сумме всех делителей другого, за исключением его самого (таковы,

например, числа 220 и 284). Напечатать все пары "дружественных" чисел, не превосходящих заданного натурального числа.

55. Найти наименьшее общее кратное четырех заданных натуральных чисел.

56. Найти все натуральные числа, не превосходящие заданного N и равные сумме кубов своих цифр.

57. Определить функцию $f(n)$, значением которой является натуральное число, получаемое выбрасыванием из записи натурального числа n первой справа цифр 0 или 5. Например, $f(13510) = 1351$, $f(1351) = 131$. Для данного натурального числа m , примените функцию f последовательно необходимое число раз, так чтобы в записи m не осталось цифр 0 и 5.

58. Последовательность целых чисел a_1, a_2, \dots, a_n определяется следующим образом:

1) a_1 - произвольное нечетное число $\langle 1$;

2) для $k > 1$
$$a_k = \begin{cases} a_{k-1} / 2, & \text{если } a_{k-1} - \text{четно} \\ 3 * a_{k-1} + 1, & \text{если } a_{k-1} - \text{нечетно} \end{cases}$$

3) последовательность заканчивается, когда в ней встречается 1. Вот последовательность, которую мы получим, исходя из 7: 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 5, 16, 8, 4, 2, 1.

Рассматривая в качестве начального значения по очереди все нечетные числа от 3 до 99, определить последовательность с максимальным количеством членов.

59. Эйлер высказал гипотезу: "Любое целое число вида $8n+3$ является суммой квадрата и удвоенного простого числа". Проверить гипотезу Эйлера для всех целых чисел вида $8n+3$ меньших 2000.

60. Даны действительные числа a, b . Получить

$$u = \min(a, b), \quad v = \min(a*b, a+b),$$

$$w = \min(u+v^2, 3.14)$$

61. Описать функцию $F(m,n) = n! * m! / (n+m)!$, где n и m - неотрицательные целые числа.

(Определить внутреннюю функцию, вычисляющую факториал).

62. Определите функцию $f(n)$, n - натуральное число, в теле которой вводятся вещественные числа x_1, x_2, \dots, x_n и значением функции является количество членов последовательности, больших предыдущего члена.

63. Даны 4 натуральных числа. Определить их наибольший общий делитель.

64. Дано натуральное $n > 0$. Найти произведение первых n простых чисел.

(Определите функцию, позволяющую распознавать простые числа).

65. Определите функцию f без параметров, в теле которой вводятся ненулевые целые числа, до тех пор пока не введется 0. Значением функции является число перемен знаков у членов введенной последовательности (например, в последовательности 1, -34, 8, 14, -5 знак меняется 3 раза).

66. Даны координаты $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ вершин четырехугольника. Определите его площадь.

67. Определите все совершенные числа < 500 . Число n называется совершенным, если оно равно сумме всех своих (положительных) делителей, кроме самого этого числа (например, число 6 совершенно: $6 = 1+2+3$).

68. Программа. В точках 1, 2, ..., k , где k - заданное целое число от 2 до 70, напечатать (по отдельности) графики следующих функций:

$F(n)$ - количество целых чисел от 1 до n , взаимно простых с числом n ;

$H(n)$ - количество положительных делителей числа n ;

$G(n)$ - количество простых чисел, не превосходящих n .

69. Даны действительные числа s, t . Получить: $g(1, 2, s) + g(t, s) - g(2s-1, st)$, где

$$g(a,b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$$

70. Программа. Дано натуральное k . Напечатать k -ую цифру в последовательности 11235813213455..., в которой выписаны все числа Фибонначи. (Определите функции: $nd(n)$ - вычисляющую количество цифр в числе n ; $d(n, k)$ - вычисляющую k -ую цифру справа в целом числе n).

71. Определите функцию $f(x, \epsilon)$, значением которой является для $|x| < 1$

$$y = \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{(-1)^n * x^{2n}}{(2n)!} + \dots$$

n такое, что $\left[\frac{x^{2n}}{(2n)!} \right] < \epsilon$ (ϵ - точность вычисления, значительно меньше 1)

72. var a,b:real; t:boolean;

Переменной t присвоить значение true, если уравнения $x^2 + 6.2x + a^2 = 0$ и $x^2 + ax + b - 1 = 0$ имеют вещественные корни и при этом оба корня первого уравнения лежат между корнями второго, и присвоить значение false во всех остальных случаях.

73. Даны длины a , b и c сторон некоторого треугольника. Найти медианы треугольника, сторонами которого являются медианы исходного треугольника. (Замечание: длина медианы, проведенной к стороне a , равна

$$0,5 * \sqrt{2 * b^2 + 2 * c^2 - a^2}$$

74. var x, y, z:real;

Вычислить: $z = (\text{sign}(x) + \text{sign}(y)) * \text{sign}(x+y)$, где

$$\text{sign}(a) = \begin{cases} -1, & \text{при } a < 0, \\ 0, & \text{при } a = 0, \\ 1, & \text{при } a > 0. \end{cases}$$

Определить и использовать функцию sign.

75. Даны отрезки a, b, c, d . Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь треугольника.

(Определите: функцию $\text{triangle}(x, y, z : \text{real}) : \text{boolean}$, определяющую можно ли построить треугольник из отрезков x, y, z ; функцию $s(x, y, z : \text{real}) : \text{real}$, вычисляющую площадь треугольника со сторонами x, y, z).

76. Даны действительные числа s, t . Получить:

$f(t, -2s, 1.17) + f(2.2, t, s-t)$, где

$$f(a, b, c) = \frac{2 * a - b - \sin(c)}{5 + |c|} \quad |$$

77. Ферма высказал гипотезу: "Любое целое число n является суммой трех треугольных чисел", т.е.

$$n = \frac{x * (x-1)}{2} + \frac{y * (y-1)}{2} + \frac{z * (z-1)}{2}$$

Проверить гипотезу Ферма для всех $n < 100$. (Определите булевскую функцию, проверяющую является ли данное число треугольным).

78. Определите функцию $f(x, \epsilon)$, $|x| < 1$, значением которой является

$$y = \text{arctg}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + \frac{(-1)^n * x^{(2n+1)}}{2n+1} + \dots$$

n такое, что $\left[\frac{x^{(2n+1)}}{2n+1} \right] < \epsilon$ (ϵ - погрешность вычисления, значительно меньше 1)

79. Даны действительные числа a, b, c . Получить:

$$\frac{\max(a, a+b) + \max(a, b+c)}{1 + \max(a+b*c, 1.15*a)}$$

80. Определите функцию $f(x, \epsilon)$, $|x| < 1$, значением которой для x является

$$y = \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + \frac{(-1)^{(n-1)}}{n} + \dots$$

n такое, что $\left[\frac{x^n}{n}\right] < \epsilon$ (ϵ - точность вычисления, значительно меньше 1)

81. Дано натуральное n. Получить:

$$\left[1 + \frac{1}{1^1}\right] * \left[1 + \frac{1}{2^2}\right] * \dots * \left[1 + \frac{1}{n^n}\right]$$

82. Дано натуральное n. Получить:

$$\left[1 + \frac{1}{1}\right]^n * \left[1 + \frac{1}{2}\right]^{n-1} * \dots * \left[1 + \frac{1}{n}\right]^1$$

83. Дано натуральное n. Получить:

$$\left[1 + \frac{1}{1^n}\right] * \left[1 + \frac{1}{2^n}\right] * \dots * \left[1 + \frac{1}{n^n}\right]$$

84. Дано натуральное n. Вычислить сумму:

$$\left[\frac{1}{1}\right]^1 + \left[\frac{1}{2}\right]^2 + \dots + \left[\frac{1}{n}\right]^n$$

85. Дано натуральное n. Получить:

$$\left[1 + \frac{1}{1}\right]^n * \left[1 + \frac{1}{2}\right]^n * \dots * \left[1 + \frac{1}{n}\right]^n$$

86. Дано натуральное n. Вычислить сумму:

$$\left[\frac{1}{1}\right]^n + \left[\frac{1}{2}\right]^{n-1} + \dots + \left[\frac{1}{n}\right]^1$$

87. Дано натуральное n. Получить:

$$\left[1 + \frac{1}{1}\right]^1 * \left[1 + \frac{1}{2}\right]^2 * \dots * \left[1 + \frac{1}{n}\right]^n$$

88. Программа. Для данного натурального n напечатать следующие значения:

$$\left[\frac{1}{1}\right]^1, \left[\frac{1}{2}\right]^2, \dots, \left[\frac{1}{n}\right]^n$$

89. Дано: натуральное n, действительные a_1, a_2, \dots, a_n . Вычислить:

$$a_1 + a_2 * (a_2 - 1) + a_3 * (a_3 - 1) * (a_3 - 2) + a_4 * (a_4 - 1) * (a_4 - 2) * (a_4 - 3) + \dots + a_n * (a_n - 1) * (a_n - 2) * \dots * (a_n - n + 1)$$

90. Рассмотрим последовательность e_1, e_2, e_3, \dots , образованную по следующему закону

$$e_k = \left[1 + \frac{1}{k} \right]^k, \quad k = 1, 2, \dots$$

Дано действительное ϵ ($0 < \epsilon < 1$). Найти первый член e_n этой последовательности, для которого $|e_n - e_{n-1}| < \epsilon$.

(Существование e_n гарантируется одной из теорем математического анализа: чем меньше ϵ , тем ближе e_n к числу $e = 2,718281828\dots$).

91. Дано натуральное n , действительные $a_1, a_2, a_3, \dots, a_n$.

Вычислить: $a_1^1 - a_2^2 + \dots + (-1)^{(n+1)} * a_n^n$

92. Эйлер высказал гипотезу "Простое число вида $4n+1$ является суммой двух квадратов". Проверить гипотезу Эйлера для всех простых чисел вида $4n+1$ меньших 2000.

93. Определите функции $f(n)$, вычисляющую n -ое число Фибонначи. Найдите сумму первых m ($m > 1$) чисел Фибонначи.

94. Определите функцию $f(n)$, n - натуральное число, в теле которой вводятся целые числа

a_1, a_2, \dots, a_n , а значением которой является количество отрицательных чисел, с которых начинается введенная последовательность.

95. Определите функцию $f(n)$, n - натуральное число, равную сумме цифр числа n . Исследуйте, есть ли среди чисел $< m$ такие n , что $(f(n))^2 = n$.

96. Даны действительные числа $x_1, y_1, x_2, y_2, \dots, x_5, y_5$. Найти периметр пятиугольника, вершины которого имеют координаты $(x_1, y_1), (x_2, y_2), \dots, (x_5, y_5)$. (Определить функцию вычисления расстояния между двумя точками, заданными своими координатами).

97. Рассмотрим некоторое натуральное число n . Если это - не палиндром, то изменим порядок его цифр на обратный и сложим исходное число с получившимся. Если сумма - не палиндром, то над ней повторяется то же действие и т.д., пока не получится палиндром. До настоящего времени неизвестно, завершится ли этот процесс для любого натурального n . Даны

натуральные числа k, l, m ($k \leq l$). Проверить, верно ли, что для любого натурального числа из диапазона от k до l процесс завершается не позднее, чем после m таких действий.

98. Два простых числа называются "близнецами", если они отличаются друг от друга на 2 (таковы, например, числа 41 и 43). Напечатать все пары "близнецов" из отрезка $[2, n]$, где n - заданное целое число, большее 2.

99. Дано натуральное $n > 0$. Найти сумму первых n составных чисел. (Определить функцию, позволяющую распознавать составные числа).

100. Даны действительные числа a_0, a_1, \dots, a_6 . Получить для $x=1,3,4$ значения $p(x+1) - p(x)$,

$$\text{где } p(y) = a_6 * y^6 + a_5 * y^5 + \dots + a_0.$$

101. Назовем натуральное число палиндромом, если его запись читается одинаково с начала и с конца (как, например, 4884, 393, 1). Найти все меньшие 100 числа - палиндромы, которые при возведении в квадрат также дают палиндром. (Определите функцию, которая для данного целого n строит число из тех же цифр, но в обратном порядке).

102. Определите функцию $f(n)$, вычисляющую n -ое число Фибонначи и найдите первое число Фибонначи, большее заданного числа m ($m > 1$).

103. Дано натуральное k . Напечатать k -ую цифру последовательности 149162536..., в которой выписаны подряд квадраты всех натуральных чисел. (Определите функции: $nd(n)$ - вычисляющую количество цифр в числе n ; $d(n,k)$ - вычисляющую k -ую справа цифру в целом числе n).

104. Определите булевскую функцию $f(n)$, n - натуральное число, в теле которой вводится последовательность вещественных чисел a_1, a_2, \dots, a_n . Значение функции равно true, если члены последовательности образуют возрастающую последовательность.

105. Даны действительные числа $x_1, x_2, x_3, y_1, y_2, y_3$. Принадлежит ли начало координат треугольнику с вершинами $(x_1, y_1), (x_2, y_2), (x_3, y_3)$? (Точка O находится внутри треугольника, если площадь всего треугольника равна $S_1+S_2+S_3$). За пояснениями обратитесь к преподавателю.

ЛАБОРАТОРНАЯ РАБОТА 3 (одномерные массивы, матрицы)

Для решения задач из этой лабораторной работы требуется знание основных правил работы с массивами, нужно четко знать, какие операции возможны над массивом в целом, а какие- только над его отдельными элементами. Смотрите примеры решения некоторых задач с массивами.

Примеры решения некоторых задач

ОДНОМЕРНЫЕ МАССИВЫ

1. Дан массив чисел. Найти, сколько в нем пар одинаковых соседних элементов.

```
const m=100;
var mas : array [1..m] of integer;
    i, k, n : integer;
begin
    write('Введите размер массива n=');
    readln(n);
    k:=0;
    for i:=1 to n do
        begin
            write('Введите элемент массива');
            readln(mas[i]);
        end;
    for i:=1 to n-1 do
        if mas[i]=mas[i+1] then k:=k+1;
    wrietln('Одинаковых пар соседних элементов ',k);
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим длину массива n , значение элементов массива и устанавливаем начальное значение k ;
- последовательно просматриваем элементы, и если очередной $mas[i]$ равен следующему $mas[i+1]$, то увеличиваем значение k на единицу;
- выводим результат.

Переменные:

mas - массив чисел;

n - размер массива;

i - переменная цикла;

k - количество одинаковых пар соседних элементов.

2. Программа. Дано 100 целых чисел. Распечатать их в обратном порядке по 6 чисел в строке.

```
const n=100; d=6;
```

```
var x : array [1..n] of integer;
```

```
    i, k : integer;
```

```
begin
```

```
    {ВВОД
```

```
массива:}
```

```
    writeln('Введите массив из ', n, ' чисел');
```

```
    for i:=1 to n do read(x[i]);
```

```
    {ВЫВОД массива по d чисел
```

```
в строке:}
```

```
    k:=0;
```

```
    {номер числа в
```

```
строке}
```

```
    for i:=n downto 1 do
```

```
    begin
```

```
        k:=k+1;
```

```
        write(x[i]:4);
```

```
        if k=d then
```

```

        begin
        k:=0; writeln
        end

    end;

    if k<>0 then writeln
end.

```

3. const k=50; m=20; n=70; {n=k+m}

```

var x : array [1..k] of real;
    y : array [1..m] of real;
    z : array [1..n] of real;
    i,p,j : integer;

```

Элементы каждого из массивов x и y упорядочены по неубыванию.

Объединить элементы этих двух массивов в один массив z так, чтобы они снова оказались упорядоченными по неубыванию.

```

const k=50; m=20; n=70;                                {n=k+m}
var x : array [1..k] of real;
    y : array [1..m] of real;
    z : array [1..n] of real;
    i, p, j : integer;

begin
    writeln('Введите массив x');
    for i:=1 to k do read(x[i]);
    writeln;
    writeln('Введите массив y');
    for i:=1 to m do read(y[i]);
    writeln;
    writeln('Введенные Вами массивы');
    writeln('Массив x');
    for i:=1 to k do write(x[i]:7:3);

```

```
writeln;
writeln('Массив y');
for i:=1 to m do write(y[i]:7:3);
writeln;
p:=1; {индекс очередного элемента из z}
i:=1; {из x}
j:=1; {из y}
{пока есть нерассмотренные
```

элементы и в x, и в y: }

```
repeat if x[i]<y[j] then
begin
z[p]:=x[i]; i:=i+1
end
else
begin
z[p]:=y[j]; j:=j+1
end;
```

p:=p+1

```
until (i>k)or(j>m);
```

{один из массивов x или y исчерпан

--> перепись в z

оставшихся элементов другого

массива:}

```
if i>k {исчерпан массив x:}
```

```
then repeat z[p]:=y[j]; j:=j+1; p:=p+1 until j>m
```

```
else repeat z[p]:=x[i]; i:=i+1; p:=p+1 until i>k;
```

```
writeln('Полученный массив: ');
```

```
for i:=1 to n do write(z[i]:7:3);
```

end.

МНОГОМЕРНЫЕ МАССИВЫ

4. Дана матрица $N \times M$, состоящая из натуральных чисел. Выбрать в строках самые левые наименьшие элементы и поставить их в первый столбец.

Для решения этой задачи нужно сначала найти самый левый минимальный элемент в строке и запомнить его местоположение, а затем поменять его местами с элементом в первом столбце.

```
const t=100; s=100;
var a : array [1..t,1..s] of integer;
    n, m, jm, i, j, min : integer;
begin
    write('Введите количество строк n=');
    readln(n);
    write('Введите количество столбцов m=');
    readln(m);
    for i:=1 to n do
        begin
            write('Вводите через пробел', m,'чисел');
            for j:=1 to m do
                read(a[i,j]);
            end;
        for i:=1 to n do
            begin
                min:= a[i,1];
                jm:=1;
                for j:=1 to m do
                    if min > a[i,j]
                        then
                            begin
                                jm:=j;
```

```

        min:=a[i,j];
    end;
    a[i,jm]:=a[i,1];
    a[i,1]:=min;
end;
for i:=1 to n do
    {вывод матрицы на экран в виде
таблицы}
    begin
        for j:=1 to m do
            write(a[i,j]:4);
            writeln;
        end
    end.
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим размеры массива A и значения его элементов;
- просматриваем строки массива слева направо, ищем минимальное значение и запоминаем значения индексов;
- для каждой строки меняем местами минимальный элемент и элемент в первом столбце;
- выводим матрицу на экран.

Переменные:

a - двумерный массив;

n,m - количество строк и столбцов массива;

i,j - переменные цикла;

jm - столбец минимального элемента для каждой строки;

min - текущий минимум.

5. Дана квадратная матрица $N \times N$, состоящая из натуральных чисел. Зеркально отразить ее элементы относительно побочной диагонали. Вывести результат на экран.

Рассмотри матрицу 3×3 и посмотри, что происходит с элементами при зеркальном отображении:

$A_{11} \ A_{12} \ A_{13} \quad A_{33} \ A_{23} \ A_{13}$

$A_{21} \ A_{22} \ A_{23} \ \leftrightarrow \ A_{32} \ A_{22} \ A_{12}$

$A_{31} \ A_{32} \ A_{33} \quad A_{31} \ A_{21} \ A_{11}$

Если считать, что после преобразования у нас появилась новая матрица B , то соответствие между элементами устанавливается следующим образом:

$B_{11} \leftrightarrow A_{33}$

$B_{12} \leftrightarrow A_{23}$

$B_{21} \leftrightarrow A_{32}$

$B_{22} \leftrightarrow A_{22}$ и т.д., т.е. $B_{[I,J]} \leftrightarrow A_{[L,M]}$

Внимательно изучив соответствие, можно утверждать, что для элементов матрицы $N \times M$ справедлива следующая система уравнений:

$I+M = N+1,$

$J+L = N+1.$

Отсюда правило преобразования элементов выглядит следующим образом:

$B_{[I,J]} = A_{[N+1-J, N+1-I]}.$

Программа, решающая данную задачу, выглядит так:

```
const n=100;
```

```
var a,b : array [1..n] of integer;
```

```
    k, m, i, j : integer;
```

```
begin
```

```
    write('Введите размер матрицы m=');
```

```
    readln(m);
```

```
    writeln('Исходная матрица');
```

```
k:=1;
for i:=1 to m do
  for j:=1 to m do
    begin
      a[i,j]:=k;
      k:=k+1;
      if j<m then write(a[i,j]:4)
        else writeln(a[i,j]:4)
    end;
  writeln('Матрица после преобразования');
  for i:=1 to m do
    for j:=1 to m do
      begin
        b[i,j]:= a[m+1-j, m+1-i];
        if j<m then write(b[i,j]:4)
          else writeln(b[i,j]:4)
      end;
    end;
  end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим размеры массива А и присваиваем значения его элементам;
- присваиваем значения элементам матрицы В и выводим их на экран.

Переменные:

a,b - двумерные массивы;

m - количество строк и столбцов массива;

i,j - переменные цикла;

k- вспомогательная переменная.

6. Вычислить: а) $C = A + B$

```
type mas=array [1..40,1..40] of real;
    vec=array[1..40] of real;
var a,b,c : mas;
    x,y : vec;
    i, j, n, k : integer;
    r, s : real;
procedure tab_in(var a1 : mas);
var i, j : integer;
begin
    for i:=1 to n do
        for j:=1 to n do
            read(a1[i,j]);
        writeln;
    end;

procedure tab_out(var b1 : mas);
var i,j:integer;
begin
    for i:=1 to n do
        begin
            for j:=1 to n do
                write(b1[i,j]:7:3);
            writeln;
        end;
    end;

begin
    writeln('Введите размер матрицы');
    readln(n);
    writeln('Введите матрицу A');
```

```
tab_in(a);
writeln('Введите матрицу B');
tab_in(b);
writeln('Матрица A');
tab_out(a);
writeln('Матрица B');
tab_out(b);
for i:=1 to n do
  for j:=1 to n do
    c[i,j]:=a[i,j]+b[i,j];
writeln('Полученная матрица C=A+B');
tab_out(c);
end.
```

б) $y = A * x$

основная программа

```
begin
  writeln('Введите размер матрицы');
  readln(n);
  writeln('Введите матрицу A');
  tab_in(a);
  writeln('Введите вектор x');
  for i:=1 to n do read(x[i]);
  writeln;
  writeln('Матрица A');
  tab_out(a);
  writeln('Введенный вектор x');
  for i:=1 to n do write(x[i]:7:2);
  writeln;
```

```
for i:=1 to n do
begin
  s:=0;
  for j:=1 to n do
    s:=s+A[i,j]*x[j];
  y[i]:=s
end;
writeln('Полученный вектор y=A*x');
for i:=1 to n do
write (y[i]:7:3);
writeln;
end.
```

в) $C = A * B$

основная программа

```
begin
  writeln('Введите размер матрицы');
  readln(n);
  writeln('Введите матрицу A');
  tab_in(a);
  writeln('Введите матрицу B');
  tab_in(b);
  writeln('Матрица A');
  tab_out(a);
  writeln('Матрица B');
  tab_out(b);
  for i:=1 to n do
    for j:=1 to n do
      begin
```

```
        s:=0;
        for k:=1 to n do
            s:=s+a[i,k]*b[k,j];
        c[i,j]:=s;
    end;
    writeln('Полученная матрица C=A+B');
    tab_out(c);
end.
```

г) транспонированную матрицу В
основная программа

```
begin
    writeln('Введите размер матрицы');
    readln(n);
    writeln('Введите матрицу В');
    tab_in(b);
    writeln('Матрица В');
    tab_out(b);
    for i:=1 to n-1 do
        for j:=i+1 to n do
            begin
                r:=b[i,j]; b[i,j]:=b[j,i]; b[j,i]:=r;
            end;
        end;
    writeln;
    writeln('Полученная транспонированная матрица В');
    tab_out(b);
end.
```

Задания на лабораторную работу 3

1. Дано n целых чисел. Распечатать их в обратном порядке по 4 числа в строке.
2. Даны целые числа a_1, a_2, \dots, a_n . Все члены последовательности с четными номерами, предшествующие первому по порядку члену со значением $\max(a_1, a_2, \dots, a_n)$, домножить на $\max(a_1, a_2, \dots, a_n)$.
3. Даны действительные числа a_1, a_2, \dots, a_n . Все числа попарно различны. Поменять в этой последовательности местами:
 - 1) наибольший и наименьший члены;
 - 2) наибольший и последний члены.
4. Даны целые числа a_1, a_2, \dots, a_n . Наименьший член последовательности a_1, a_2, \dots, a_n заменить целой частью среднего арифметического всех членов, остальные члены оставить без изменения. Если в последовательности несколько членов со значением $\min(a_1, a_2, \dots, a_n)$, то заменить последний по порядку.
5. Даны целые числа $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n$. Преобразовать последовательность b_1, b_2, \dots, b_n по правилу:
если $a_i \leq 0$, то b_i увеличить в 10 раз,
иначе b_i заменить нулем ($i = 1, 2, \dots, n$).
6. Даны две последовательности по n целых чисел в каждой. Найти наименьшее среди тех чисел первой последовательности, которые не входят во вторую последовательность (считая, что хотя бы одно число есть).
7. В заданном целочисленном массиве найти элементы, сумма которых равна данному числу, в предположении, что такие существуют.
8. Даны натуральное m , действительные a_1, a_2, \dots, a_n (числа a_1, a_2, \dots, a_n попарно различны, $m \leq n$). В последовательности a_1, a_2, \dots, a_n поменять местами наибольший член и член с номером m .

9. Даны действительные числа a_1, a_2, \dots, a_n . Требуется домножить все члены последовательности a_1, a_2, \dots, a_n на квадрат ее наименьшего члена, если $a_1 \geq 0$, и на квадрат ее наибольшего члена, если $a_1 < 0$.
10. Даны действительные числа $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n$.
Вычислить $(a_1 + b_n) * (a_2 + b_{n-1}) * \dots * (a_n + b_1)$.
11. Дана последовательность из n различных целых чисел. Найти сумму чисел этой последовательности, расположенных между максимальным и минимальным числами (в сумму включить и оба этих числа).
12. Даны целые числа a_1, a_2, \dots, a_n . Получить новую последовательность, выбросив из исходной все члены со значением $\max(a_1, a_2, \dots, a_n)$.
13. Даны целые числа a_1, a_2, \dots, a_n . Получить новую последовательность из n целых чисел, заменяя a_i нулями, если $|a_i|$ не равно $\max(a_1, a_2, \dots, a_n)$, и заменяя a_i единицами в противном случае ($i = 1, 2, \dots, n$).
14. Программа. Даны координаты n точек на плоскости: $x_1, y_1, \dots, x_n, y_n$. Найти номера двух точек, расстояние между которыми наибольшее (считать, что такая пара точек единственная).
15. Даны действительные числа a_1, a_2, \dots, a_n . Если в результате замены отрицательных членов последовательности a_1, a_2, \dots, a_n их квадратами члены будут образовывать неубывающую последовательность, то получить сумму членов исходной последовательности; в противном случае получить их произведение.
16. `const n = ...;`
`var x, y : array [1..n] of real;`
Преобразовать массив x по следующему правилу (воспользоваться массивом y как вспомогательным): все отрицательные элементы массива x перенести в его начало, а все остальные - в конец, сохраняя исходное

взаимное расположение как среди отрицательных, так и среди остальных элементов.

17. `const n = ...;`

`var s : array [1..n] of integer;`

Напечатать те элементы массива s , индексы которых являются полными квадратами

(1, 4, 9, 16, 25,...).

18. Дана последовательность из n целых чисел. Определить количество инверсий в этой последовательности (т.е. таких пар элементов, в которых большее число находится слева от меньшего: $x_i > x_j$ при $i < j$).

19. Дано натуральное число n . Сколько различных цифр встречается в его десятичной записи.

20. Даны целые числа a_1, a_2, \dots, a_n . Если в данной последовательности ни одно четное число не расположено после нечетного, то получить все отрицательные члены последовательности, иначе - все положительные. Порядок следования чисел в обоих случаях заменять на обратный.

21. Определить булевскую функцию, проверяющую на равенство элементов двух массивов (порядок не учитывать).

22. Даны целые числа a_1, a_2, \dots, a_n , каждое из которых отлично от нуля. Если в последовательности отрицательные и положительные члены чередуются (+, -, +, -,.... или -, +, -, +,...), то ответом должна служить исходная последовательность. Иначе получить все отрицательные члены последовательности, сохранив порядок их следования.

23. По заданным коэффициентам многочлена 5 степени и многочлена 6 степени определить коэффициенты произведения этих многочленов.

24. Даны натуральное число n , действительные числа a_1, a_2, \dots, a_n . Получить b_1, b_2, \dots, b_{10} , где b_i равно сумме тех членов последовательности a_1, \dots, a_n , которые принадлежат полуинтервалу $[i-1, i$

$] (i = 1, 2, \dots, 10)$. Если полуинтервал не содержит членов последовательности, то соответствующие b_i положить равным нулю.

25. Составьте программу, которая реализует для массивов операции, аналогичные действиям над множествами.

26. По заданным коэффициентам многочлена $P(x)$ 5 степени и многочлена $Q(x)$ 6 степени определить коэффициенты многочлена $P(Q(x))$.

27. Даны вещественные числа a_0, a_1, \dots, a_n . Найти коэффициенты многочлена:

$$(x - a_0) * (x - a_1) * \dots * (x - a_n).$$

28. Даны действительные числа $a_1, a_2, \dots, a_n, \dots, a_{2n}$. Преобразовать эту последовательность по правилу: большее из a_i и a_{n+i} ($i = 1, 2, \dots, n$) принять в качестве нового значения a_i , а меньшее - в качестве нового значения a_{n+i} .

29. `const n = ...;`

`var x, y : array [1..n] of real;`

Дано k ($1 \leq k \leq n-1$). Преобразовать массив x по следующему правилу (воспользоваться массивом y как вспомогательным): элементы массива x циклически сдвинуть на k позиций влево.

30. `const n = ...;`

`type digit = 0..9; number = array [1..n] of digit;`

`var a, b, c : number; t : boolean;`

Рассматривая a и b как последовательности цифр десятичной записи некоторых неотрицательных целых чисел, получить c – аналогичное представление для суммы этих двух чисел. Если в сумме окажется больше n цифр, то ее левую цифру отбросить, а переменной t присвоить `true`, иначе переменной t присвоить значение `false`.

31. `type мантисса = array [1..9] of '0'..'9';`

`порядок = array [1..2] of '0'..'9';`

`var m : мантисса; p : порядок; x : real;`

Переменной x присвоить вещественное число

$$0.m[1]m[2]...m[9]*10^{(p1p2)}$$

32. const $n = \dots$;

Вычислить величину

$$\frac{x_1 * y_n^2 + x_2 * y_{n-1}^2 + \dots + x_n * y_1^2}{y_1 * x_n^2 + y_2 * x_{n-1}^2 + \dots + y_n * x_1^2}$$

33. Программа. Напечатать величины a_0, a_1, \dots, a_n , где a_0 - заданное целое число,

$$a_i = a_{i/2} + a_{i-1} \text{ при } i = 1, 2, \dots, n. [i/2] - \text{означает целую часть числа } i/2.$$

34. Даны действительные числа a_1, a_2, \dots, a_n , среди которых заведомо есть как отрицательные, так и неотрицательные. Получить

$$x_1 * y_1 + \dots + x_s * y_s,$$

где x_1, \dots, x_p - отрицательные члены последовательности a_1, a_2, \dots, a_n ,
взятые в порядке их следования,

y_1, y_2, \dots, y_q - неотрицательные члены, взятые в обратном порядке,

$$s = \min(p, q).$$

35. const $n = \dots$; $n1 = \dots$; $\{n1 = n+1\}$

var P, Q : array $[0..n]$ of real;

a : real;

По P - массиву коэффициентов многочлена

$$P(x) = p_0 * x^n + p_1 * x^{n-1} + \dots + p_{n-1} * x + p_n$$

получить Q - массив коэффициентов многочлена $P(x + a)$.

36. Дано целое k от 2 до 20. Найти коэффициенты k -го многочлена

Чебышева. (Замечание: многочлены Чебышева $T_n(x)$ определяются

формулами $T_0(x) = 1$; $T_1(x) = x$;

$$T_n(x) = 2 * T_{n-1}(x) - T_{n-2}(x) ; n = 2, 3, \dots).$$

37. Даны действительные числа $a_1, a_2, \dots, a_n, \dots, a_{2n}$.

Получить $\max(a_1 + a_{2n}, a_2 + a_{2n-1}, \dots, a_n + a_{n+1})$;

$$\min(a_1 * a_n, a_2 * a_{n+1}, \dots, a_{n+1} * a_{2n}).$$

38. Даны натуральные числа n_1, n_2, \dots, n_7 , действительные числа x_1, x_2, \dots, x_7 . Вычислить

$$\frac{n_1 * x_1 + n_2 * x_2 + \dots + n_7 * x_7}{n_1 + n_2 + \dots + n_7}$$

39. const n =...;

var x : array [1..n] of real; y:real;

Напишите программу для вычисления

$$y = x_1 * x_n - x_2 * x_{n-1} + \dots + (-1)^{i+1} * x_i * x_{n+1-i} + \dots + (-1)^{n+1} * x_n * x_1$$

40. const n =...; {n- нечетно}

var x : array [1..n] of real; y : real;

Вычислить

$$y = x_1^2 - x_2^2 + x_3^2 + \dots + x_{n-1}^2 + x_n^2$$

41. const n= ...; m=...;

var x : array [1..n] of real;

y : array [1..m] of 1..100;

s : real;

Вычислить s - сумму тех элементов массива x, индексы которых совпадают со значениями элементов массива y ($y_i \diamond y_j$, при $i \diamond j$).

42. Последовательность целых чисел a_1, a_2, \dots, a_n определяется следующим образом:

1) Первый член последовательности есть произвольное нечетное число большее 1;

2) для $k > 1$ $a_k = a_{k-1} / 2$, если a_{k-1} - четно

$3 * a_{k-1} + 1$, если a_{k-1} - нечетно;

3) последовательность заканчивается, когда в ней встречается 1. Вот последовательность, которую мы получим, исходя из 7 =>

7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Напишите программу, которая печатает последовательности, если в

качестве начального значения по очереди берутся все нечетные числа от 3 до 99.

43. Элемент матрицы назовем седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для заданной целой матрицы размером $N \times M$ напечатать индексы всех ее седловых точек.

44. Дана вещественная матрица размером $N \times N$, все элементы которой различны. Найти скалярное произведение строки, в которой находится наибольший элемент матрицы, на столбец с наименьшим элементом.

45. `var k : integer;`

`с : array [1..n, 1..m] of char;`

Определить k - количество различных элементов массива c (т.е. повторяющиеся элементы считать один раз).

46. Дана (построчно) вещественная матрица размером $M \times N$.

Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (один из них) оказался в верхнем левом углу.

47. `var k : integer;`

`с : array [1..n, 1..m] of integer;`

Определить k - количество "особых" элементов массива c , считая элемент "особым", если в его строке слева от него находятся элементы, меньшие его, а справа - большие.

48. Определить, является ли заданная целая квадратная матрица N -го порядка симметричной (относительно главной диагонали).

49. Дана матрица целых чисел размером $M \times N$. Найти номера строки и столбца наибольшего элемента матрицы.

50. `var A : array [1..10,1..10] of integer;`

Заполнить массив A следующим образом:

91 92 100

81 82 90

71 71 80

.....

11 12 20

1 2 10

51. var A : array [1..10, 1..10] of integer;

Заполнить массив A следующим образом:

1 2 3 10

0 1 2 9

0 0 1 8

.....

0 0 0 1

52. var A : array [1..4,1..5] of integer;

b : array [1..4] of boolean;

По массиву A получить массив b, присвоив его k-му элементу значение true, если элементы

k-й строки массива A упорядочены по убыванию, и значение false иначе.

53. Начиная с центра обойти по спирали все элементы матрицы размером 9x9 (распечатывая их в порядке обхода).

54. var A : array [1..4,1..5] of integer;

b : array [1..4] of boolean;

По массиву A получить массив b, присвоив его k-му элементу значение true, если k-ая строка массива A симметрична, и значение false иначе.

55. Определить, является заданная целая квадратная матрица n-го порядка магическим квадратом, т.е. такой, в которой суммы элементов во всех строках и столбцах одинаковы.

Пример:

0 1 2 3 4

1 2 3 4 0

2 3 4 0 1

3 4 0 1 2

4 0 1 2 3

56. Определить, является ли заданная целая квадратная матрица n -го порядка ортонормированной, т.е. такой, в которой скалярное произведение каждой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно 1.
57. Программа. Элемент матрицы назовем седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для заданной целой матрицы размером 10×10 напечатать индексы всех ее седловых точек.
58. `var k : integer;`
`C : array [1..n,1..m] of integer;`
 Определить k - количество "особых" элементов массива C , считая элемент особым, если он больше суммы остальных элементов своего столбца.
59. `type scr=array [1..n,1..n] of 0..1;`
`var s : scr;`
 Преобразовать массив s , осуществив поворот элементов вокруг его центра на 90 градусов против часовой стрелки.
60. `type вектор = array [1..n] of integer; (n-четно)`
`матрица = array [1..n] of вектор;`
`var A : матрица; x : вектор;`
 В матрице A поменять местами 1-ю и 2-ю строки, 3-ю и 4-ю строки, ..., $(n-1)$ -ю и n -ю строки (воспользоваться x как вспомогательным массивом).
61. `var A : array [1..6,1..6] of boolean;`
`B : array [1..5,1..5] of boolean;`
`n,k : 1..6;`
 Получить массив B из массива A удалением n -ой строки и k -го столбца.
62. Соседями элемента $a[i, j]$ в матрице назовем элементы $a[k, l]$

64

$c(i-1) \leq k \leq (i+1), (j-1) \leq l \leq (j+1), (k, l) \neq (i, j)$. Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером $N \times M$.

63. const n =...;

var A : array [1..n,1..n] of real;

x,y : array [1..n] of real;

Вычислить $y = A^2 * x$.

64. const n =..;

var A,B,C : array [1..n,1..n] of real;

Вычислить $C = (A+B)^2$.

65. Расстояние между k -ой и l -ой строками матрицы $A = \| a[i, j] \|$ определяется как

$$\sum_{j=1}^n |a[k, j]| * |a[l, j]|$$

Указать номер строки, максимально удаленной от первой строки заданной матрицы.

66. var A : array [1..n,1..m] of real;

x : array [1..m] of real;

Заполнить массив A по следующему правилу: $A[i, j] = (x[j])^i$.

67. const n =..;

var A : array [1..n,1..n] of real; s:real;

Найти сумму элементов из области массива A, отмеченной символом '*':

```
*****
* 0 0 0 0 *
* 0 0 0 0 *
* 0 0 0 0 *
* 0 0 0 0 *
* 0 0 0 0 *
*****
```

68. По заданной квадратной матрице размером 10x10 построить вектор длиной 19, элементы которого - максимумы элементов диагоналей, параллельных главной диагонали.

69. var A : array [1..n,1..n] of real;

Найти сумму элементов из области матрицы A, отмеченной символом '*'

(диагонали входят в выделенную область):

```

* * * * *
0 * * * * 0
0 0 * * * 0 0
0 0 0 * 0 0 0
0 0 * * * 0 0
0 * * * * * 0
* * * * *

```

70. Даны натуральное n и (построчно) элементы квадратной вещественной матрицы A четвертого порядка. Вычислить n -ю степень этой матрицы ($A^1 = A$; $A^2 = A * A$; $A^3 = A^2 * A$ и т.д.)

71. Имеется таблица T результатов некоторого шахматного турнира, в котором участвовало n

шахматистов ($n > 2$):

T : array [1..n,1..n] of (w,e,f,s),

где $T[i, j] = w$, если i -ый участник выиграл у j -го (при этом $T[j, i] = f$);

$T[i, j] = e$, если i -ый и j -ый участники сыграли вничью;

$T[i, i] = s$.

Возможный вид таблицы:

s w f

f s e

w e s

За выигрыш дается 1 очко, за ничью - 0.5 очка, за проигрыш - 0 очков.

Выдать на печать номера участников в порядке невозрастания набранных ими очков.

72. const n = ...;

var A : array[1..n,1..n] of real; s:real;

Найти s - сумму элементов из области массива A, отмеченной символом '*'

!:

```
* 0 0 0 0 0 *
0 * 0 0 0 * 0
0 0 * 0 * 0 0
0 0 0 * 0 0 0
0 0 * 0 * 0 0
0 * 0 0 0 * 0
* 0 0 0 0 0 *
```

73. var A : array [1..n,1..n] of real; (n-нечетно)

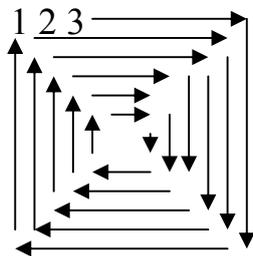
Найти сумму элементов из области матрицы A, отмеченной символом '*':

```
1          n
0 0 0 * 0 0 0
0 0 * * * 0 0
0 * * * * * 0
* * * * * * *
0 * * * * * 0
0 0 * * * 0 0
n 0 0 0 * 0 0 0
```

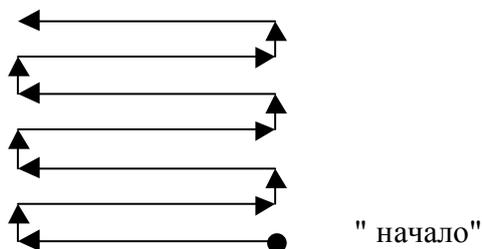
74. По матрице $A = \| a[i, j] \|$ размером 10×10 построить матрицу $B = \| b[i, j] \|$ того же размера, где $b[i, j]$ определяется следующим образом: через $a[i, j]$ проведем в A диагонали, параллельные главной и побочной диагоналям; $b[i, j]$ определяется как максимум в области массива A, отмеченной символом '*':

```
          j
0 0 0 0 1 0 0 0 1 0
0 0 0 0 0 1 0 1 0 0
i 0 0 0 0 0 0 0 * 0 0 0
```


77. Получить целочисленную квадратную матрицу порядка $2n+1$, элементами которой являются числа $1, 2, 3, \dots, (2*n + 1)$, расположенные в ней по спирали.



78. Напечатать элементы заданной матрицы размером $N \times N$ (N -четно) в следующем порядке:



ЛАБОРАТОРНАЯ РАБОТА 4 (символьный и строковый типы)

Примеры решения некоторых задач

1. Вывести в одну строку $A B B C C C \dots Z Z \dots Z$.

```
var i : char;
    k,j : integer;
begin
    k:=1;
    for i:= 'A' to 'Z' do
    begin
        for j:=1 to k do
```

```

    write(i);
    k:=k+1;
end;
writeln;
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- цикл for i:='A' to 'z' do определяет, какую букву выводим на печать;
- цикл for j:=1 to k do определяет, сколько раз будет печататься буква;
- после вывода всех букв закрываем строку оператором writeln.

Переменные:

- i - переменная цикла; определяет, какая буква выводится;
- k - количество повторений буквы;
- j - переменная цикла.

2. Дана строка символов. Удалить из нее все знаки препинания.

```

var  str : string;
     l, i : integer;
     m : set of char;
begin
    m:=['!', ',', ';', ':', '.', '?', '-', ''];
    writeln('Введите текст');
    readln(str);
    l:=length(str);
    writeln('Преобразованный текст');
    for i:=1 to l do
        if not(str[i] in m) then write(str[i]);
    writeln;
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- задаем значение множества m - множества знаков препинания;
- цикл `for i:=1 to l do` осуществляет вывод строки без знаков препинания;
- после вывода закрываем строку оператором `writeln`.

Переменные:

i - переменная цикла;

l - длина строки;

`str` - строка текста;

m - множество знаков препинания.

3. Присвоить литерным переменным c_2 , c_1 и c_0 соответственно левую, среднюю и правую цифры трехзначного числа k .

```
var c0, c1, c2 : char;
```

```
    n0, k, d : integer;
```

```
begin
```

```
    writeln('Введите трехзначное число');
```

```
    readln(k);
```

```
    n0:=ord('0');
```

```
    d:=k div 100;      c2:=chr(n0+d);
```

```
    d:=k mod 100 div 10; c1:=chr(n0+d);
```

```
    d:=k mod 10;      c0:=chr(n0+d);
```

```
    writeln('c2= ', c2);
```

```
    writeln('c1= ',c1);
```

```
    writeln('c0= ',c0);
```

```
end.
```

4. Используя только литерный ввод, т.е. процедуру `readln(c)`, где c - литерная переменная, ввести непустую последовательность цифр, перед которой

может находиться знак "+" или "-" и за которой следует пробел, и, получив соответствующее число, присвоить его целой переменной k.

```

var c : char;
    sign, n0, k : integer;
begin
                                                    {определение знака
числа:}
    sign:=1;
    readln(c);
    if c='- ' then
        begin sign:=-1;
            readln(c)
        end
    else if c='+ ' then readln(c);
                                                    {чтение цифр (первая - в c) и вычисление
абсолютной вели-
                                                    чины числа k по схеме Горнера:}

    n0:=ord('0'); k:=0;
    repeat
        k:=10*k+ord(c)-n0;
        readln(c)
    until c=' ';
                                                    {учет знака:}

    k:=sign*k;
    writeln(k);
end.

```

5. Дана строка символов. Выделить подстроку между первой и последней точкой.

```
var str, s1 : string;
```

```
    i, m, j : integer;
begin
    writeln('Введите текст');
    readln(str);
    s1:=str;
    i:=pos('.',s1);
    j:=i; m:=0;
    if (i<>0) then
        begin
            while i<>0 do
                begin
                    delete(s1,1,i);
                    m:=m+1;
                    i:=pos('.',s1);
                end;
            if m<>j then
                begin
                    s1:=copy(str,j+1,m-j-1);
                    writeln('Часть текста между 1-й и последней точкой');
                    writeln(s1);
                end
            else writeln('В тексте только одна точка');
        end
    else writeln('В тексте нет ни одной точки');
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим строку str и присваиваем значение вспомогательной переменной s1:=str;

- определяем местоположение первой точки в тексте; если точка есть, то вырезаем из `s1` текст до нее;
- ищем последнюю точку в цикле `while i <> 0 do`; если она есть, то значение переменной `l` указывает на ее местоположение;
- в зависимости от присутствия точек выделяем подстроку `s1:=copy(str,j+1,m-j-1)` и выводим результат на экран.

Переменные:

`i` - номер позиции, которая соответствует точке;

`J` - местоположение первой точки;

`m` - местоположение последней точки;

`str` - строка текста;

`s1` - вспомогательная переменная.

6. Дана строка символов. Определить, является ли она записью десятичного числа, кратного трем.

Необходимо удостовериться, что введенная строка состоит только из цифр, т.е. может быть преобразована в десятичное число. Само преобразование цифры в число сделать очень просто. Поскольку коды символьных переменных - цифр - следуют один за другим, то функция `ord` в выражении `ord(s1[i])-ord('0')` поможет нам сделать это.

```
var s1 : string;
    k, i, m : integer;
begin
    writeln('Введите текст до точки');
    readln(s1);
    m:=length(s1);
    k:=0; i:=1;
    repeat
        case s1[i] of
            '0'..'9' : k:=k+(ord(s1[i])-ord('0'));
```

```

    ' ' : ;
    else k:=-1;
    end;
    i:=i+1;
until (i>m) or (k<0);
if k mod 3 = 0
    then writeln('Это десятичное число, кратное 3')
    else writeln('Это не десятичное число, кратное 3')
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим строку s1;
- организуем в строке поиск цифр до тех пор, пока не встретим конец строки или не обнаружим наличие ошибки в арифметическом выражении;
- в зависимости от значения суммы цифр введенного числа выводим результат на экран.

Переменные:

- i - переменная цикла;
- s1 - строка цифр;
- m – длина строки;
- k - счетчик цифр.

7. Программа. Напечатать заданную непустую строку:

- а) удалив из нее все цифры и удвоив знаки "+" и "-";

```

var s : string;
    i : integer;
begin
    writeln('Введите строку:');
    readln(s);

```

```

for i:=1 to length(s) do
    if (s[i]='+')or(s[i]='-') then write(s[i],s[i])
        else if (s[i]<'0')or(s[i]>'9') then write(s[i]);
    writeln
end.

```

б) удалив из нее все знаки "+", непосредственно за которыми идет цифра;

```

var a, b : char;           {a - очередная литера строки,
                           b - следующая}
s : string;
i : integer;
begin
    writeln('Введите строку:');
    readln(s);
    for i:=1 to length(s) do
        begin
            a:=s[i];
            if a<>'+' then write(a)
                else if i<length(s) then begin
                    b:=s[i+1];
                    if (b<'0') or (b>'9') then write(a)
                        end
                    else write(a);
            end;
        writeln
    end.

```

8. Дана строка символов. Группы символов в ней между группами пробелов считаются словами.

Посчитать, сколько слов содержит данная строка.

76

```
var s1 : string;
    j, k, i : integer;
begin
    writeln('Введите текст до точки');
    readln(s1);
    j:=length(s1);
    if s1[j]=' ' then k:=-1
        else k:=0;
    s1:=copy(s1,1,j-1);
    repeat
        i:=pos(' ',s1);
        delete(s1,1,i);
        if i<>1 then k:=k+1;
    until i=0;
    writeln('количество слов k=',k)
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим строку s1;
- организуем поиск в строке пробелов до тех пор, пока они там встречаются;
- найдя пробел, отрезаем очередное слово и следующий за ним пробел;
- если в найденном слове есть хотя бы одна буква, то увеличиваем счетчик слов;
- выводим результат на экран.

Переменные:

- i - вспомогательная переменная;
- J - местоположение первой точки;
- s1 - строка текста;
- k - счетчик слов.

9. Дана строка символов. Группы символов в ней между группами пробелов считаются словами.

Определить длину самого короткого и самого длинного слова.

```
var s1 : string;
    max, min, i ,j : integer;
begin
    writeln('Введите текст до точки');
    readln(s1);
    s1:=s1 + ' ';
    max:=0; min:=256;
    repeat
        i:=pos(",s1);
        delete(s1,1,i);
        if i-1 >max then max:=i-1;
        if (i-1 <min) and (i>1) then min:=i-1;
    until i=0;
    writeln('Самое длинное слово max=',max);
    writeln('Самое короткое слово min=',min);
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- вводим строку s1;
- устанавливаем начальные значения max и min;
- организуем поиск в строке пробелов до тех пор, пока они там встречаются;
- найдя пробел, отрезаем очередное слово вместе со следующим за ним пробелом и сравниваем его длину с max и min;
- в случае необходимости переустанавливаем значения max и min;
- выводим результат на экран.

Переменные:

i - номер позиции, которая соответствует пробелу;

J - местоположение первой точки;

s1 - строка текста;

max - длина наибольшего слова;

min - длина наименьшего слова.

10. Дана строка. Преобразовать строку, заменив в ней каждую группу стоящих рядом точек одной точкой.

```
var s : string;
```

```
    k : byte;
```

```
begin
```

```
    writeln('Введите строчку');
```

```
    readln(s);
```

```
    while pos('..', s) <> 0 do
```

```
        delete (s, pos('..', s), 1);
```

```
    writeln('Полученная строка : ', s);
```

```
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;

- вводим строку *s*;

- найдем пару стоящих рядом точек, удалим одну из них; повторим поиск... - и так до тех пор, пока в строке не окажется ни одной такой пары точек.

- выводим результат на экран.

Переменные:

s - строка текста.

Задания на лабораторную работу 4

1. Для заданных трех строк S1, S2 и S3 определить какая из этих строк удовлетворяет следующему свойству: строка является десятичной записью числа, кратного 9.
2. Для заданных трех строк S1, S2 и S3 определить какая из этих строк является палиндромом.
3. Для заданных трех строк S1, S2 и S3 определить какая из этих строк является десятичной записью числа, кратного 4.
4. Для заданных трех строк S1, S2 и S3 напечатать те латинские буквы, которые входят только в одну из этих строк.
5. Даны две строки S1 и S2. Создать строку, состоящую из латинских букв, не принадлежащих какой-либо строке S1 и S2.
6. Ввести две строки S1 и S2, состоящие только из цифр. Определить в какой строке сумма цифр большая.
7. Даны две строки S1 и S2. Создать строку, состоящую из символов, входящих либо в S1, либо в S2, но не в обе сразу.
8. Для двух заданных строк, состоящих только из цифр и начинающихся с 0, найти строку - сумму чисел, представленных данными строками.
9. Из данной строки удалить все цифры и малые латинские буквы.
10. Составьте программу, в результате выполнения которой в первой заданной строке удваивается каждый символ, принадлежащий также второй строке.
11. Составить программу, в результате выполнения которой из первой заданной строки удаляется каждый символ, не принадлежащий также второй строке.
12. Составить программу, в результате выполнения которой из первой заданной строки удаляется каждый символ, принадлежащий второй строке.
13. В данной строке удвоить все цифры и малые латинские буквы.
14. Составить программу, в результате выполнения которой в первой заданной строке удваивается каждый символ, не принадлежащий второй строке.
15. Даны две строки. Создать строку, состоящую только из символов, общих для двух данных строк.

16. Для каждой цифры '0', '1', ... , '9' подсчитать количество вхождений в данную строку S.
17. Для каждой малой латинской буквы 'a', 'b' , 'c' , ... , 'z' напечатать количество вхождений в данную строку S.
18. Дана строка, состоящая из слов. Напечатать те слова, которые являются палиндромами.
19. Русский шифр 15-16 веков:
Все согласные буквы русской азбуки записывались в два ряда; одна половина букв вверху, другая половина внизу, причем в обратном порядке (одна буква под другой),
Б В Г Д Ж З К Л М Н
Щ Ш Ч Ц Х Ф Т С Р П
При зашифровке слов согласные взаимно заменялись, а остальные буквы вписывались без изменений. Написать программу шифровки и дешифровки текста.
20. Исключить из данной строки группы символов, расположенные между скобками (,). Сами скобки тоже должны быть исключены. Предполагается, что внутри каждой пары скобок нет других скобок.
21. Программа. Проверить, правильно ли в данной строке расставлены круглые скобки (т.е. находится ли справа от каждой открывающей скобки соответствующая закрывающая скобка, а слева от каждой закрывающей - соответствующая открывающая). Ответ - "да" или "нет".
22. В заданном предложении найти пару слов, из которых одно является обращением другого.
23. Программа. Определить, является ли заданная строка правильной записью вещественного числа (возможно со знаком, а также, с пробелами спереди и/или сзади).
24. Ввести как строку целое число. Определить сумму его цифр.
25. Программа. В заданную непустую строку входят только цифры и буквы.

Определить, удовлетворяет ли строка следующему свойству: строка содержит (помимо букв) только одну цифру, причем ее числовое значение равно длине строки.

26. Дана строка S . Создать новые строки из строки S :

- а) заменить все восклицательные знаки точками;
- б) заменить каждую точку многоточием (т.е. тремя точками);
- в) заменить каждую из групп стоящих рядом точек одной точкой;
- г) заменить каждую из групп стоящих рядом точек многоточием (т.е. тремя точками).

27. Дана строка S . Каждую подстроку длиной $n > 1$, состоящую из одинаковых символов, заменить на подстроку из тех же символов в количестве $2*n$ штук.

28. Пусть дана строка S , состоящая только из цифр. Найти максимальную по длине

неубывающую подпоследовательность цифр, входящую в S .

29. Для каждого из слов заданного предложения указать, сколько раз оно встречается в предложении.

30. Дана строка S . Любую подстроку в S назовем словом, если она удовлетворяет

следующим трем условиям:

- а) не содержит пробелов;
- б) следует после пробела или стоит в начале строки S ;
- в) после нее следует пробел или подстрока стоит в конце строки.

Из строки S сделать строку S_1 , причем в каждом слове поменять местами первый и последний символы, а все остальные оставить без изменения.

31. Программа. Определить, является ли заданная строка правильной записью целого числа (возможно со знаком, а также, с пробелами спереди и/или сзади).

32. Для заданной строки определить длину содержащейся в ней максимальной подстроки, не имеющей латинских букв.

33. Сколько букв останется на своих местах в данной строке, если ее перевернуть.

34. Программа. В заданную непустую строку входят только цифры и буквы. Определить,

удовлетворяет ли она следующему свойству: строка начинается с k букв ($1 \leq k \leq 9$), за которыми следует только одна литера - цифра с числовым значением k .

35. Дана строка. Удалить из нее все группы подряд стоящих одинаковых символов.

36. Дана строка. Подсчитать:

а) сколько раз среди данных символов встречается символ '+' и сколько раз символ '*';

б) общее число вхождений символов '+', '-', '*'.

37. Заданы две строки длиной m и n , причем $m > n$. Проверить, входит ли меньшая строка в большую (не используйте функцию pos).

38. Дана строка S и $S1$. Каждое вхождение $S1$ в строку S в качестве подстроки заменить на обращенную подстроку $S1$.

39. Перечислить все слова заданного предложения, которые состоят из тех же букв, что и первое слово предложения.

40. Дана строка S и дано натуральное n . Удалить из строки S все группы длиной n подряд стоящих одинаковых символов.

41. Дана строка S . Будем называть подстроку, входящую в S , словом, если она следует после пробела, не содержит пробела и после нее также следует пробел. Не забудем о словах, с которых начинается и заканчивается строка S . Напечатать все слова, входящие в данную строку.

42. Программа. Из данной строки сделать новую строку, заменив в ней все цифры на соответствующие слова: "один", "два", "три" и т.д.

43. Дана строка S . Любую подстроку t в S назовем словом, если она удовлетворяет следующим трем условиям:

а) не содержит пробелов;

б) следует после пробела или стоит в начале строки S;

в) после t следует пробел или t стоит в конце строки.

Напечатать все слова, входящие в S и являющиеся идентификаторами в смысле Паскаля.

44. Найти максимум длины таких начальных отрезков заданной строки, которые имеют вид vv , где v - симметричная подстрока.

45. Программа. В заданную непустую строку входят только буквы и цифры. Определить, удовлетворяет ли она следующему свойству: строка является десятичной записью числа, кратного 9.

46. Дана строка. Преобразовать ее, удалив каждый символ "*" и повторив каждый символ, отличный от "*".

47. В данной строке найти самую длинную подстроку, состоящую из одинаковых символов.

48. Найти самое длинное симметричное слово заданного предложения.

49. Программа. Определить, является ли заданная строка правильной записью идентификатора (возможно с пробелами спереди и/или сзади).

50. Заданы две строки. Найти самую длинную их общую часть.

51. Программа. Удалить из заданной строки лишние пробелы, т.е. из нескольких подряд идущих пробелов оставить только один.

52. Определить, является ли данная строка палиндромом, не учитывая пробелов (т.е., если удалить пробелы, то строка читается одинаково с начала и с конца: "А роза упала на лапу Азора"). Использовать только исходную строку, не разрешая получить новую, удаляя пробелы из старой. ("Аргентина манит негра", "Нажал кабан на баклажан").

53. Дана строка S. Любую подстроку t в S назовем словом, если она удовлетворяет следующим трем условиям:

а) не содержит пробелов;

б) следует после пробела или стоит в начале строки S;

в) после t следует пробел или t стоит в конце строки.

Напечатать в обратном порядке все слова (повернуть слова, начиная с конца), входящие в данную строку.

54. Дана строка. Группы символов, разделенных пробелами (одним или несколькими) и не содержащими пробелов внутри себя, будем называть словами. Преобразовать строку так, что порядок символов в каждом слове изменился на противоположный.

Пример: исходная строка 'abc 5* -3y zzk'
 выходная строка 'cba *5 y3- kzz'

55. Напечатать строки следующего вида

abc xyz

bc yza

c zab

..... Всего 26 строк

yzab x

zab xy

ЛАБОРАТОРНАЯ РАБОТА 5 (процедуры и функции)

Для решения задач пятой лабораторной работы нужно обязательно использовать подпрограммы, разработанные вами для решения какой либо части поставленной задачи.

Отдельные части модульной программы, предназначенные для решения частных задач, организованы в подпрограммы. Преимущества в том, что один и тот же фрагмент можно использовать многократно, путем вызова соответствующей подпрограммы.

Программы можно писать небольшими частями, эти части проще отлаживать, выявлять ошибки при тестировании.

Различают подпрограммы-процедуры и подпрограммы-функции.

Описание подпрограммы-функции располагается в разделе описаний программы и начинается с зарезервированного слова `function`:

`Function name_of_function (arguments_list): type_of_result;`

Имя функции **name_of_function** задается в соответствии с правилами Паскаля.

Список параметров **arguments_list** содержит имена параметров функции и их тип через двоеточие, если используются параметры различных типов, указывается тип для каждой группы параметров.

Идентификатор типа результата **type_of_result** определяет тип значения функции.

Внутренняя структура функции аналогична структуре программы – сначала идут описания, а затем исполняемые операторы. Завершается служебным словом **end**, после которого ставится точка с запятой.

В теле функции обязательно присутствует оператор присваивания в левой части которого **имя** функции, а в правой – **результат** выполнения функции.

Обращение к подпрограмме-функции производится путем указания ее имени в составе какого-либо выражения.

Особенности функции в том, что результат ее выполнения – единственное значение.

Допустимыми типами результата являются следующие:

- все разновидности вещественного типа;
- строковый тип;
- скалярный тип;
- указатель.

Иногда результаты работы подпрограммы – это целый набор значений, а иногда этот результат не сводится к вычислениям. В этом случае используются подпрограммы – процедуры.

Заголовок процедуры имеет следующий вид:

`procedure name_of_procedure (list_of_parameters);`

Здесь **list_of_parameters** – список параметров процедуры.

Имя процедуры, в отличие от имени функции, не является носителем результата, поэтому тип в заголовке не описывается.

Обращение к процедуре производится путем указания ее имени в нужном месте программы (но не в составе какого-либо выражения!) со списком всех параметров процедуры, если они есть.

Примеры использования процедур и функций приведены ниже.

При решении задач этой лабораторной работы особое внимание следует уделить параметрам процедур и функций.

Параметры подразделяются на **формальные** (те, что указаны в заголовке подпрограммы при ее описании) и **фактические** (те, которые используются при запуске подпрограммы в программе).

Параметры – значения – они подаются на вход подпрограммы своим значением, то есть перед запуском подпрограммы вычисляется значение переменной, записывается в буфер, это значение и используется в подпрограмме.

Параметры – переменные – они подаются на вход подпрограммы своим адресом, то есть в процессе выполнения подпрограммы значение этой переменной может быть изменено и новое значение будет доступно для внешней программы.

Примеры решения некоторых задач

1. Программа. Даны отрезки a, b, c и d . Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь данного треугольника. (Определить процедуру $\text{print}(x, y, z)$, печатающую площадь треугольника со сторонами x, y и z , если такой треугольник существует).

```
var a,b,c,d : real;
procedure print(x,y,z:real);
  var p:real;
  begin
    if (x+y>z)and(y+z>x)and(z+x>y) then
```

```

begin p:=(x+y+z)/2;
writeln(sqrt(p*(p-x)*(p-y)*(p-z)))
end
else writeln('Такой треугольник построить

```

нельзя ');

```
end;
```

```
begin
```

```
  writeln('Введите 4 отрезка a,b,c,d>0 ');
```

```
  read(a,b,c,d);
```

```
  write('Значение площади треугольника a,b,c - ');
```

```
  print(a,b,c);
```

```
  write('Значение площади треугольника a,b,d - ');
```

```
  print(a,b,d);
```

```
  write('Значение площади треугольника a,c,d - ');
```

```
  print(a,c,d);
```

```
  write('Значение площади треугольника b,c,d - ');
```

```
  print(b,c,d)
```

```
end.
```

2. Пусть процедура `maxmin(x,y)` присваивает параметру `x` большее из вещественных чисел `x` и `y`, а параметру `y` - меньшее. Описать данную процедуру и использовать ее для перераспределения значений вещественных переменных `a`, `b` и `c` так, чтобы стало $a \geq b \geq c$.

```
var a,b,c : real;
```

```
procedure maxmin( var x,y:real);
```

```
  var r:real;
```

```
  begin if x<y then
```

```
    begin r:=x;
```

```
      x:=y;
```

```

        y:=r
    end
end;

```

```
begin
```

```

    writeln('Введите три числа a,b,c -');
    readln(a,b,c);
    maxmin(a,b);
    maxmin(a,c);                    {a=max}
    maxmin(b,c);                    {c=min}
    writeln(a,b,c);
end.

```

```
3. type shift=1..99;
```

```
    scale=array [1..100] of real;
```

Описать процедуру `move(s,k)`, которая преобразует шкалу `s`, циклически сдвигая ее элементы на `k` позиций влево, где `k`- параметр типа `shift`.

```
type shift=1..99;
```

```
    scale=array [1..100] of real;
```

```
var a:scale; m:shift;n,i:integer;
```

```
procedure move( var s:scale; k:shift);
```

```
    var i:integer; t:scale;                {вспомогательный массив}
```

```
    begin
```

```
        {t[n-k+1..n]:=s[1..k]:}
```

```
        for i:=1 to k do t[n-k+i]:=s[i];
```

```
        {t[1..n-k]:=s[k+1..n]:}
```

```
        for i:=k+1 to n do t[i-k]:=s[i];
```

```
        s:=t
```

```
    end;
```

```

begin
  writeln('Введите размер массива');
  readln(n);
  writeln('Введите массив');
  for i:=1 to n do read(a[i]);
  writeln('Введите цикл сдвига <=', n-1);
  readln(m);
  writeln('Исходный массив');
  for i:=1 to n do write(a[i]:7:2);
  writeln;
  move(a,m);
  writeln('Полученный массив');
  for i:=1 to n do write(a[i]:7:2);
  writeln;
end.

```

4. Дан прямоугольный бильярдный стол со сторонами A и B , где A, B - натуральные числа (бильярд Льюиса Кэролла). Из угловой лузы вылетает шар под углом 45 градусов к боковым стенкам, ударяется о борт, отскакивает, ударяется еще раз и т.д., пока не вылетит через одну из угловых луз. Рассчитать количество отрезков в ломанной траектории шара. Считать угол падения равным углу отражения.

Данная задача решается с помощью стандартных функций выделения целой части от деления y на x $y \div x$ и выделения остатка $y \bmod x$. При прохождении шаром прямоугольного стола и отражении его от боковых сторон происходит увеличение числа отрезков траектории на два, а обратный путь вычисляется как $y := a - x + y \bmod x$, где y - обратный путь для шара, a - длинная сторона стола, x - короткая сторона стола.

```

var a, b : integer;
function bill(y,x:integer):integer;

```

```
var k:integer;
begin
  k:=0;
  while y mod x <>0 do
  begin
    k:=k+y div x+2;
    y:=a-x+y mod x;
  end;
  bill:=k;
end;
```

```
begin
  repeat
    writeln('Введите два натуральных числа A>B');
    readln(a,b);
  until a>=b;
  writeln('Количество отрезков в траектории :', bill(a,b));
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- создаем описание функции bill;
- вводим два натуральных числа a и b (не кратные друг другу);
- вызываем функцию bill для определения количества отрезков;
- завершаем работу программы.

Переменные:

в функции bill:

- x,y - два натуральных числа (формальные параметры);
- k - вспомогательная переменная (локальная переменная);
- a - длинная сторона стола (глобальная переменная);

в основной программе:

a, b - два натуральных числа (глобальные переменные).

Задания на лабораторную работу 5

1. Определить процедуру $p(x, y)$, которая для данного массива x , создает массив y такой, что y_i равно наименьшему (отличному от 1) делителю числа x_i (для всех i).
2. Определить процедуру $p(n, m)$, n и m - натуральные числа, в теле которой вводятся целые числа a_1, a_2, \dots, a_n и значением выходного параметра m является количество отрицательных чисел, с которых начинается введенная последовательность.
3. Определите процедуру $p(x, y)$, которая для данного массива чисел x создает массив y , состоящий из элементов массива x , но переставленных так, что сначала идут все положительные числа, а потом все остальные.
4. Определить процедуру $p(n, m)$, n - натуральное число, в теле которой вводятся вещественные числа x_1, x_2, \dots, x_n и значением выходного параметра m является количество членов последовательности, больших предыдущего числа.
5. Определить процедуру $p(x, y)$, которая для данного массива чисел x создает массив y по следующему правилу: пусть m - наибольший элемент среди x_i .
Тогда $y_i = (x_i)^2$, если $x_i < m/2$ и $y_i = 0$ иначе.
6. Определить процедуру $p(n, \max, \min)$, n - натуральное число, в теле которой вводится последовательность вещественных чисел x_1, x_2, \dots, x_n и выходные параметры \max и \min равны соответственно наибольшему и наименьшему членам этой последовательности.
7. Определить процедуру $p(x, y)$, которая для данного массива x , состоящего из натуральных чисел, создает массив y такой, что y_i равно сумме делителей x_i (для всех i).

8. Определить процедуру $p(x, y)$, которая для данного массива x создает массив y по следующему правилу: $y_i = x_i - m$, где m - среднее арифметическое элементов массива x .

9. `const n = ...;`

`type vector = array [1..n] of real;`

`var a, b, c, d : vector;`

Пусть процедура $sum(x, y, z)$ присваивает вектору z сумму векторов x и y .

Описать данную процедуру и использовать ее для вычисления $d = a + b + c$.

10. Даны отрезки a, b, c и d . Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь треугольника.

(Определить процедуру $print(x, y, z)$, печатающую площадь

треугольника со сторонами x, y и z , если такой треугольник существует).

11. Даны действительные числа $U_1, U_2, V_1, V_2, W_1, W_2$.

Получить $U^2 + V - 10W$, где

U, V, W - комплексные числа $U_1+iU_2, V_1+iV_2, W_1+iW_2$. (Определить процедуры выполнения арифметических операций над комплексными числами).

12. Программа. Даны 3-элементные вещественные векторы x, y и квадратные матрицы A и B 3-го порядка. Вычислить величину

$$(A * x, B * y) + (B * x, y) / (x, A * y)$$

13. Дана матрица $M \times N$ порядка. Вычислить максимальные элементы в каждой строке и минимальный элемент из этих максимальных элементов.

14. Программа. Даны две квадратные матрицы 4-го порядка. Вычислить квадрат той матрицы, в которой максимум элементов наибольший.

15. Программа. Даны 5-ти элементные вещественные векторы x, y, z . Вычислить величину $(a, a) - (b, c)$, где a обозначает тот из этих векторов, в котором самый большой минимальный элемент (считать, что такой вектор единственный), b и c обозначает два других вектора, $a(p, q)$ – скалярное произведение p и q .

16. Программа. Даны 4-х элементные вещественные векторы x, y, z . Вычислить величину $(a, a) + (b, c)$, где a обозначает тот из этих векторов, в котором самый меньший максимальный элемент (считать, что такой вектор единственный), b и c обозначает два других вектора, $a(p, q)$ – скалярное произведение p и q .
17. Программа. Даны три вещественные квадратные матрицы 3-го порядка. Напечатать ту из них, норма которой наименьшая (считать, что такая матрица одна). В качестве нормы матрицы взять максимум абсолютных величин ее элементов.
18. Программа. Даны координаты вершин треугольника и координаты некоторой точки внутри него. Найти расстояние от данной точки до ближайшей стороны треугольника. (При определении расстояний учесть, что площадь треугольника вычисляется и через три его стороны, и через основание и высоту).
19. Программа. Даны две квадратные вещественные матрицы 4-го порядка. Напечатать квадрат той из них, в которой наименьший след (след – сумма элементов на главной диагонали), считая, что такая матрица одна.
20. Программа. Даны три квадратные вещественные матрицы 3-го порядка. Напечатать ту из них, в которой наибольший след (след - сумма элементов на главной диагонали), считая, что такая матрица одна.
21. Программа. Даны две квадратные вещественные матрицы 4 порядка. Вычислить куб той матрицы, в которой сумма элементов наибольшая.
22. Даны две квадратные матрицы A и B .
Вычислить:
- 1) $A*B - B*A$
 - 2) $A^2 + B^2 + 2*A*B$
23. Даны 3-х элементные вещественные векторы x и y и квадратные матрицы A и B . Вычислить:
- $$(A^2 * x, y) + (B^2 * y, x) + (A * B * x, y)$$

24. Программа. Даны 3 квадратные целые матрицы A, B и C четвертого порядка.

Вычислить:

1) A^2, B^2, C^2

2) $A * B, B * C, C * A$

3) $A * B * C$

25. Дана матрица квадратная A 3-го порядка и целое число $n > 0$. Вычислить A^n .

26. Даны 3 квадратные матрицы 4-го порядка A, B и C. Вычислить $A^2 + B^2 + A*B*C$

27. Программа. Даны 2 квадратные матрицы с целыми элементами 3-го порядка A и B. Вычислить:

1) транспонированную A, транспонированную B

2) транспонированную $(A + B)$

3) транспонированную $(2 * A)$ - транспонированную $(3 * B)$

28. Пусть e - единичный вектор n-го порядка и A - квадратная матрица n-го порядка.

Пусть t - вещественное число. Вычислить: $(t * A * e, t * e) * A$

29. Даны две квадратные целые матрицы A и B. Пусть E - единичная матрица.

Вычислить:

1) $E * A - A * E$

2) $A^2 - 2*A*B + B^2$

30. Даны 3-ех элементные вещественные векторы x и y и квадратная матрица A 3-го порядка. Вычислить матрицу:

$$(x, y) * A + (A * x, y) * A + (A * y, x) * A$$

31. Даны три вектора x, y, z. Вычислить

$$x^2 + y^2 + z^2 + (x, y) + (z * (x, 2), y * (z, 5)),$$

где 2 и 5 - векторы, состоящие только из чисел 2 или 5 соответственно.

32. Пусть A - квадратная матрица с целыми элементами 4-го порядка.

Вычислить величину

$$B = \sum_{i=1}^n \sum_{j=1}^n (i * j + a[i, j])$$

33. Программа. По заданным 5-ти элементным целым массивам a , b и c вычислить

$$t = \begin{cases} \frac{\min(b[i, j])}{\max(a[i, j])} + \frac{\max(c[i, j])}{\min(b[i, j] + c[i, j])} & \text{если } \min(a[i, j]) < \max(b[i, j]) \\ \max(b[i, j] + c[i, j]) + \min(c[i, j]) & \text{иначе} \end{cases}$$

34. Программа. По заданным 4-х элементным целым массивам a , b и c вычислить

$$t = \begin{cases} \frac{\min(b[i, j] + c[i, j])}{\max(a[i, j]) + \max(c[i, j])} & \text{если } \max(a[i, j] + c[i, j]) > \max(b[i, j]) \\ 1 / (\max(b[i, j]) + \min(b[i, j])) & \text{иначе} \end{cases}$$

35. Программа. По заданным 4-ем элементным целым массивам a , b , c вычислить:

- 1) $\max(a[i, j])$, $\max(b[i, j])$, $\max(c[i, j])$
- 2) $\min(a[i, j])$, $\min(b[i, j])$, $\min(c[i, j])$
- 3) $\max(a[i, j] + b[i, j])$, $\max(a[i, j] + c[i, j])$, $\max(b[i, j] + c[i, j])$
- 4) $\min(a[i, j] - b[i, j])$, $\min(a[i, j] - c[i, j])$, $\min(b[i, j] - c[i, j])$
- 5) $\max(a[i, j] * b[i, j])$, $\max(a[i, j] * c[i, j])$, $\max(b[i, j] * c[i, j])$

36. Программа. По заданным 4-элементным вещественным векторам x , y , z вычислить

$$W = \begin{cases} \prod_i (\sin(x[i, j]) + 2) & \text{если } \prod_i (1 - y[i, j]^2) > 0.5 \\ \prod_i (1 - z[i, j]^2) & \text{иначе} \end{cases}$$

37. Программа. По заданным 4-элементным вещественным векторам x , y , z вычислить

$$W = \begin{cases} \prod_i (1 - y[i]^2) & \text{если } \prod_i (\cos(z[i])) > 0 \\ \prod_i (1 - x[i]^2) & \text{иначе} \end{cases}$$

38. Программа. По заданным 4-элементным вещественным векторам x, y, z
вычислить

$$W = \begin{cases} \prod_i (\cos(x[i]^2)) & \text{если } \prod_i (1 - x[i]^2) > 0.5 \\ \prod_i (1 - z[i]^2 * y[i]^2) & \text{иначе} \end{cases}$$

39. Программа. По заданным 4-элементным целым массивам x, y вычислить

$$V = \begin{cases} \sum_{i=1}^4 x[i]^2 & \text{если } \sum_{i=1}^4 x[i]^2 > \sum_{i=1}^4 y[i]^2 \\ \sum_{i=1}^4 x[i] * y[i] & \text{иначе} \end{cases}$$

40. Программа. По заданным 4-элементным целым массивам a, b и c вычислить

$$T = \begin{cases} \frac{\max(c[i]) - a[i]}{\min(b[i]) - \min(a[i])} & \text{если } \min(a[i] - b[i]) > \min(c[i]) \\ \min(c[i]) - \max(b[i]) & \text{иначе} \end{cases}$$

41. Программа. По заданным 5-элементным целым массивам x, y вычислить

$$V = \begin{cases} \sum_{i=1}^5 x[i]^2 & \text{если } \sum_{i=1}^5 x[i] * y[i] > 0 \\ \sum_{i=1}^5 y[i]^2 & \text{иначе} \end{cases}$$

42. Программа. По вещественному $a > 0$ вычислить величину

$$\frac{a^{1/3} - (a^2 + 1)^{1/6}}{1 + (3 + a)^{1/7}}$$

Корни $y = x^{1/k}$ вычислять с точностью $E=0.0001$ по следующей
итерационной

формуле: $y[0] = 1$;

$$y[n+1] = y[n] + (x / y[n]^{k-1} - y[n]) / k \quad (n=0,1,2,\dots),$$

приняв за ответ приближение $y[n+1]$, для которого $|y[n+1] - y[n]| < E$.

43. Напишите процедуру, которая удаляет из одномерного вещественного массива наибольшее значение.
44. Напишите процедуру, которая преобразовывает одномерный массив $Ar1D$, состоящий из n^2 элементов, в двумерный массив $Ar2D$ размера $n \times n$.
45. Напишите подпрограмму, которая в заданной строке удаляет все пробелы и знаки препинания { . , ; : ! ? }. Примените её для трех заданных строк (строки вводятся с клавиатуры).
46. Даны три строки, в каждой из этих строк заменить все символы * на символы !!! (напишите процедуру).
47. Напишите функцию, которая преобразует символьный массив в строковое значение.
48. Напишите процедуру, которая для любого целого аргумента возвращает массив, содержащий цифры в записи этого аргумента.
49. Напишите функцию, которая для любого целого аргумента возвращает количество цифр в его записи.
50. Дано: n : integer;
 C :char;
 S :string;
- Вставит в заданную строку символ C начиная с позиции n , напишите процедуру.
51. Вычислите выражение $z(x)=(\text{sign}(x)+\text{sign}(y))*\text{sign}(x+y)$). При решении задачи определите и используйте функцию sign :

$$\text{sign}(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0. \end{cases}$$

52. Заданный массив целых чисел делится на три части двумя элементами: максимальным и минимальным. Определите сумму элементов в каждой части массива. Используйте функции для нахождения индексов максимального и минимального элементов и подсчета суммы элементов в указанной части массива.

53. Пусть дана матрица A размером $m \times n$, состоящая из вещественных чисел. Необходимо получить матрицу B размером $m \times n$, каждый элемент b_{ij} которой равен true, если сумма соседних с a_{ij} элементов меньше указанной величины S , и false – в противном случае. Воспользуйтесь функцией для нахождения суммы соседних элементов.

54. Пусть дана матрица A размером $m \times n$, состоящая из целых чисел. Определите для каждого элемента позиции элементов матрицы с такой же суммой цифр.

55. В квадратной матрице определите количество строк, отвечающих следующему условию:

- а) упорядоченных по возрастанию;
- б) упорядоченных по убыванию.

56. В квадратной матрице определите количество строк, отвечающих следующему условию:

- а) состоящих из равных элементов;
- б) неупорядоченных.

57. Найдите все слова в предложении, которые содержат только заданные буквы. Для определения буквенного состава слова используйте процедуру.

58. Пусть даны N шестизначных номеров троллейбусных билетов – XXXXXX. Определите те билеты, номера которых являются «счастливыми», то есть сумма первых трех цифр совпадает с суммой последних трех цифр. Используйте функцию для определения, является ли номер «счастливым».

59. Составьте процедуру выравнивания строки, которая преобразует строку к строке указанной длины путем равномерного добавления пробелов между всеми словами.

60. Пусть дана матрица A размером $n \times n$, найдите максимум всех минимальных элементов матрицы по столбцам.

ЛАБОРАТОРНАЯ РАБОТА 6 (множества)

При решении задач данной лабораторной работы нужно обязательно использовать такой тип данных, как Множества.

Для описания этого типа данных используется служебное слово **set**, например:

```
type M=set of Char;
var b : m;
```

Базовым типом множества может быть любой конечный скалярный тип, состоящий не более чем из 256 элементов. Если переменная типа «множество» описана как

Set of 1..3, она может принимать значения из следующего набора:

{(1,2,3), (1,2), (1,3), (2,3), (1), (2), (3), ()}.

Константы множественного типа записываются с помощью квадратных скобок и списка элементов. Примеры:

Const

```
Alphabet=['A'..'Z', 'a'..'z'];
```

```
Empty=[ ];
```

```
Digits=[0..9];
```

Отсутствие списка элементов в квадратных скобках обозначает пустое множество. Резервированное слово **in** используется для определения принадлежности элемента множеству:

If ch **in** alphabet **then** ...

Чтобы добавить элемент в множество, можно добавить множество, состоящее из единого элемента, а чтобы убрать элемент из множества, можно отнять множество, состоящее из единого элемента:

S:=S+[ch];

S:=S-[ch];.

Для этих же целей можно воспользоваться стандартными процедурами для работы с множествами:

Include(S, ch) – для добавления элемента ch в множество S.

Exclude(S, ch) для исключения элемента ch из множества S.

Примеры решения некоторых задач

1. type str=string[100];

Описать функцию count(s), подсчитывающую общее количество цифр и знаков '+', '-' и '*', входящих в строку s.

```
type str=string[100];
```

```
var s1 : str;
```

```
    k1 : integer;
```

```
function count(var s:str) : integer;
```

```
var i, k : integer;
```

```
begin
```

```
    k:=0;
```

```
    for i:=1 to length(s) do
```

```
        if s[i] in ['0'..'9', '+', '-', '*'] then k:=k+1;
```

```
    count:=k
```

```
end;
```

```
begin
```

```
    writeln('Введите строку');
```

```
    readln(s1);
```

```
    k1:=count(s1);
```

```
writeln('Количество цифр и знаков - ', k1:5);
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- создаем функцию count, подсчитывающую общее количество цифр и знаков '+', '*', '-' в заданной строке;
- вводим строку s1;
- вызываем функцию count и выводим значение счетчика k1 на экран;
- завершаем работу программы.

Переменные:

в функции count:

s – строка (формальный параметр);

i – счетчик цикла (локальная переменная);

k – общее количество цифр и знаков '+', '*', '-' в заданной строке (локальная переменная);

в основной программе:

s1 – введенная строка (локальная переменная);

k1 - общее количество цифр и знаков '+', '*', '-' в заданной строке

s1(фактический параметр).

2. type M=set of 0..99;

Описать функцию card(A), подсчитывающую количество элементов в множестве A типа M (ноль используйте как прекращение ввода). (Например, card([5,8,23])=3.)

```
type M=set of 0..99;
```

```
var b : m;
```

```
    k1, x : integer;
```

```
function card(A:M) : integer;
```

```
var p,k : integer;
```

```

begin
    k:=0;
    for p:=0 to 99 do
        if p in A then k:=k+1;
    card:=k
end;

```

```

begin
    b:=[];
    repeat
    write('Введите число >0<=99 - ');
    readln(x);
    b:=b+[x];
    until x=0;
    k1:=card(b);
    writeln('Количество различных элементов множества', k1);
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- создаем функцию card, подсчитывающую количество элементов в множестве;
- организуем пустое множество;
- в цикле repeat ... until вводим числа до тех пор, пока не ввели 0, и дописываем их в множество;
- вызываем функцию card и выводим значение счетчика k1 на экран;
- завершаем работу программы.

Переменные:

в функции card:

A – множество (формальный параметр);

p – счетчик цикла от 0 до 99 (локальная переменная);

k – количество элементов множества A (локальная переменная);

в основной программе:

x – введенное число (локальная переменная);

b – созданное множество (локальная переменная);

k1 - количество элементов в созданном множестве (фактический параметр).

3. Программа. Дан текст из строчных латинских букв, за которым следует точка. Напечатать первые вхождения букв в текст, сохраняя их исходный взаимный порядок;

var

let : set of 'a'..'z';

s : string;

c : char;

i : integer;

begin

let:=[]; {множество букв в рассмотренной

части текста}

readln(s);

i:=1;

while s[i]<>'.' do

begin if not(s[i] in let) then {1-е вхождение}

begin write(s[i]);

c:=s[i];

let:=let+[c]

end;

i:=i+1;

end;

writeln

end.

Для решения задачи:

- формируем тело программы и описываем переменные (let – множество, состоящее из малых латинских букв);
- организуем пустое множество let;
- вводим строчку s;
- организуем цикл, в котором просматриваем символы строки s до тех пор, пока не встретим '.', и проверяем, входит ли этот символ в множество let (т.е. символ – малая латинская буква);
- если очередной символ строки не входит во множество let, то этот символ встретился впервые и выводим ее на экран и дописываем в множество;
- завершаем работу программы.

Переменные:

let – множество малых латинских букв;

s – очередной символ строки;

s – введенная строка;

i – переменная цикла.

Задания на лабораторную работу 6

1. const n=3; type номер=1..n;

матрица = array [номер, номер] of real;

НОМ = set of номер;

Описать функцию sum (A, S1, S2), вычисляющую сумму тех элементов матрицы, номера строк и столбцов которых принадлежат соответственно непустым множествам S1 и S2 типа НОМ.

Программа: Введите матрицу A, множества S1 и S2, каждое из 3-х элементов, и вызовите функцию sum (A, S1, S2).

2. Спортлото: 6 из 49.

Составьте программу, в которой загадываются, иначе говоря, создаются 6 разных чисел, значения которых никак не связаны друг с другом, а величина лежит в интервале от 1 до 49.

(Выражение $\text{random}(49)+1$ дает случайное целое число в интервале от 1 до 49).

3. В возрастающем порядке напечатать все целые числа из диапазона 1..255, представимые в виде

$$n^2 + m^2, \text{ где } m, n \geq 0.$$

4. type s = set of 0..255;

Описать процедуру print (A), печатающую в убывающем порядке все элементы множества A, имеющего тип S.

Программа: введите множество A (ноль используйте как прекращение ввода) и напечатайте его в убывающем порядке.

5. Дана строка. В алфавитном порядке напечатайте (по разу) все малые латинские буквы, входящие в эту строку ровно один раз.

6. Введите множество, состоящее из 20 целых чисел от 1 до 50. Определите, сколько чисел, у которых первая значащая цифра в десятичной записи есть 1 или 2.

7. Дана строка. Подсчитайте сумму цифр, не входящих в данную строку.

8. Введите множество, состоящее из 15 целых чисел от 0 до 255. Определите, сколько среди них точных квадратов (одинаковые числа считать по разу).

9. type M = set of 0..99;

Описать функцию card (A), подсчитывающую количество элементов в множестве A типа M.

(Например, $\text{card}([5,8,23])=3$).

Программа: введите множество A (ноль используйте как прекращение ввода). Выдайте, вызвав функцию card, количество элементов A.

10. type letters = set of 'a' .. 'z';

Описать процедуру `print (A)`, печатающую в алфавитном порядке все элементы множества A , имеющего тип `letters`.

Программа: введите множество A , состоящее из 20 букв, и напечатайте его в алфавитном порядке.

11. Дана строка. В алфавитном порядке напечатайте все малые латинские буквы, не входящие в эту строку.

12. Введите множество, состоящее из 20 целых чисел от 1 до 50. Определите, сколько среди них чисел Фибоначчи.

13. Дана строка из строчных латинских букв. Напечатать все буквы, входящие в строку по одному разу.

14. Дана строка из строчных латинских букв. Напечатать первые вхождения букв в текст, сохраняя их взаимный порядок.

Указание: формируйте множество, в которое очередной элемент добавляется после первой печати.

15. `type S = set of 0..100;`

Описать функцию `sum (A)`, подсчитывающую сумму нечетных элементов множества A типа S . (Например, `sum ([3,4,99,80]) = 102`).

Программа: введите множество A (ноль используйте как прекращение ввода). Выдайте, вызвав функцию `sum`, сумму нечетных элементов множества A .

16. `type S = set of 0..50;`

Описать функцию `p (A)`, подсчитывающую произведение элементов множества A типа S . (Например, `sum ([5,8,23]) = 920`).

Программа: введите множество A (ноль используйте как прекращение ввода). Выдайте, вызвав функцию `p`, произведение элементов множества A .

17. Дана строка. В алфавитном порядке напечатайте (по разу) все малые латинские буквы, входящие в эту строку более одного раза.

18. Введите множество, состоящее из 20 целых чисел от 1 до 50. Определите, сколько чисел, у которых последняя цифра 3, 5 или 7.

19. Дана строка символов. Определить, каких букв больше в этом тексте, больших латинских букв или малых.
20. Создайте множество, состоящее из простых чисел < 255 и дающее при делении на 4 остаток 3. Напечатайте элементы этого множества.
21. Дана строка символов. В возрастающем порядке напечатайте все цифры, входящие в эту строку.
22. Дана строка. В алфавитном порядке напечатайте (по разу) все строчные латинские согласные буквы, входящие в эту строку. Указание: гласные буквы - а, е, і, о, u; остальные - согласные.
23. Введите множество, состоящее из 20 целых чисел от 1 до 100. Определите, сколько среди них чисел, которые не являются числами Фибоначчи.
24. Создайте множество, состоящее из чисел Фибоначчи меньших 255. Напечатайте элементы этого множества в порядке убывания.
25. Введите множество, состоящее из 20 целых чисел от 1 до 100. Определите, сколько среди них простых чисел.
26. Дана строка символов. Подсчитайте количество различных латинских малых букв, входящих в данную строку.
27. Дана строка. В алфавитном порядке напечатайте (по разу) все большие латинские гласные буквы, входящие в эту строку.
Указание: гласные буквы - а, е, і, о, u; остальные - согласные.
28. Дана строка из строчных латинских букв. Напечатайте все буквы, входящие в текст не менее двух раз.
Указание. Просматривая в цикле символы текста, формируйте два множества: одно - содержит уже просмотренные символы, другое - наполняется теми элементами, которые входят в первое множество.
29. Дана строка. Определить, каких букв больше в этом тексте, латинских строчных гласных или согласных.
Указание: гласные буквы - а, е, і, о, u; остальные - согласные.
30. Создайте множество, состоящее из составных чисел < 255 . Напечатайте элементы этого множества в порядке убывания.

31. Дана строка. В алфавитном порядке напечатайте (по разу) все малые латинские буквы, входящие в эту строку ровно два раза.
32. Описать функцию "счет", подсчитывающую общее количество цифр и знаков '+', '-' и '*', входящих в данную строку.
Программа: введите строку и с помощью функции "счет" определите число указанных символов.
33. Ввести строку символов, состоящую из латинских букв, цифр и пробелов. Осуществить проверку правильности введенных символов.
34. Спортлото: 5 из 36.
Составьте программу, в которой загадываются, иначе говоря, создаются 5 разных чисел, значения которых никак не связаны друг с другом, а величина лежит в интервале от 1 до 36.
(Выражение $\text{random}(36)+1$ дает случайное целое число в интервале от 1 до 36).
35. Напечатать все натуральные числа от 10 до 32767, в десятичной записи которых нет одинаковых цифр.
36. Создайте множество, состоящее из 30 целых чисел от 1 до 255. Определите, каких чисел больше в множестве – простых или составных.
37. Дана строка. В алфавитном порядке напечатайте (по разу) все большие латинские буквы, входящие в эту строку более одного раза.
38. `type S = set of 0..200;`
Описать функцию `chet (A)`, подсчитывающую сумму четных элементов множества `A` типа `S`. (Например, `sum ([3,4,98,81] = 102)`).
Программа: введите множество `A` (ноль используйте как прекращение ввода). Выдайте, вызвав функцию `chet`, сумму четных элементов множества `A`.
39. Дана строка. Напечатайте те гласные буквы, которые встречаются в каждом слове строки.

40. В порядке убывания напечатайте все целые числа из диапазона $1 \dots 4900$, которые представимы в виде $n^2 + 2k^2$, но не представимы в виде $7ij + j + 3$ ($n, k, i, j \geq 0$).
41. Составьте программу вычисления суммы номеров мест, на которых в строке S стоят гласные буквы.
42. Дана целочисленная квадратная матрица размером $n \times n$. Элементы матрицы находятся в диапазоне $1..100$. Напечатать все цифры из заданного диапазона, которых нет ни в одной из строк матрицы.
43. Дана строка символов. Посчитать общее число вхождений английских букв в строку.
44. Дана строка. Найти наибольшее количество цифр идущих подряд.
45. Дано натуральное число n ($n \geq 2$). Найдите все меньшие n простые числа, используя решето Эратосфена. Решетом Эратосфена называют следующий способ определения простых чисел. Выпишем подряд все целые числа в диапазоне от 2 до n . Первое простое число 2, подчеркнем его, а все большие числа, кратные 2 зачеркнем. Первое из оставшихся чисел – 3. Подчеркнем его, а все большие числа, кратные трем, зачеркнем. Первое число из оставшихся теперь – 5, так как 4 уже зачеркнуто. Подчеркнем его как простое, а все большие числа, кратные пяти, зачеркнем и т.д.
46. С помощью решета Эратосфена (см. задание 45) найдите четверки меньших n простых чисел, принадлежащих одному десятку (например, 11, 13, 17, 19).
47. Напечатайте все целые числа в диапазоне от 1 до 1600, которые представимы в виде $x^2 + y^2$, но которые нельзя представить в виде $xy = c^2$, где c изменяется в диапазоне от 1 до 5.
48. Вводится последовательность слов. Определите, какое количество слов потребуется, чтобы задействовать все буквы английского алфавита. Ввод слов заканчивается, когда задействованы все буквы.

49. Вводится последовательность слов. Определите, какое количество слов потребуется, чтобы задействовать все буквы русского алфавита. Ввод слов заканчивается, когда задействованы все буквы.

50. Напечатайте все целые числа в диапазоне от 1 до 3600, которые представимы в виде $5n+7m$, где n и m – целые числа ($n, m > 0$).

ЛАБОРАТОРНАЯ РАБОТА 7 (сортировка)

Для решения задач седьмой лабораторной работы требуются знания основных методов сортировки, которые были даны в курсе лекций.

Задача. Дана вещественнозначная матрица a с n строками и m столбцами. Определим для каждой строки a_i функцию

$$f(a_i) = a_{i1} - a_{i2} + a_{i3} - \dots + (-1)^{m+1} a_{im}.$$

Переставить строки матрицы a по убыванию значений функции f . (использовать метод простого выбора).

```
const n=3; m=4;
```

```
type
```

```
  index=1..n;
```

```
  element = array[1..m] of real;
```

```
  mas = array[index] of element;
```

```
function f(s:element):real; {функция для нахождения знакопередающей
суммы в строке матрицы}
```

```
var
```

```
  i:integer;
```

```
  y:real;
```

```
begin
```

```
y:=0;
for i:=1 to m do
    if i mod 2 = 0 then y:=y-s[i]
        else y:=y+s[i];
f:=y
end;
procedure sort(var a:mas); {процедура, реализующая алгоритм простого выбора}
var i,j:index;
    x:element;
begin
    for i:=1 to n-1 do
        begin
            k:=i;
            x:=a[i];
            for j:=i+1 to n do
                if f(a[i])< f(x) then begin k:=j; x:=a[j]
            end;
            a[k]:=a[i];
            a[i]:=x
            end;
end;

var
    a:mas;
...
begin
    {ввод матрицы a}
    sort(a);
    {печать матрицы a}
end.
```

Задания на лабораторную работу 7

1. Пусть для целого $n > 1$ функция $f(n)$ равна наименьшему делителю числа n , большему 1.
Дан массив a : array [1..n] of 1.. maxint. Упорядочить элементы массива a по возрастанию в соответствии со следующим критерием: элемент x считается "большим" элемента y , если $f(x) > f(y)$.
2. Дан массив a : array [1..n] of 1.. maxint. Упорядочить элементы массива a по возрастанию в соответствии со следующим критерием: элемент x считается "больше" элемента y , если сумма цифр числа x больше суммы цифр числа y .
3. Дан массив a : array [1..n] of 1.. maxint. Упорядочить элементы массива a по возрастанию в соответствии со следующим критерием: элемент x считается "больше" элемента y , если последняя цифра числа x больше последней цифры числа y .
4. Пусть для целого $n > 0$ функция $d(n)$ равна количеству делителей числа n . Дано $m > 0$. Упорядочить все числа n от 1 до m в порядке неубывания значений $d(n)$. Указание: использовать только один массив длиной m .
5. Пусть для целого $n > 0$ функция $s(n)$ равна сумме делителей числа n . Дано $m > 0$. Упорядочить все числа n от 1 до m в порядке неубывания значений $s(n)$. Указание: использовать только один массив длиной m .
6. Пусть дана строка, в которой между словами находится по одному пробелу. Переставить слова по убыванию в лексикографическом порядке.
7. Пусть дана строка, в которой между словами находится по одному пробелу. Переставить слова по возрастанию в лексикографическом порядке.
8. Пусть для целого $n > 0$ функция $s(n)$ равна сумме цифр числа n . Дано $m > 0$. Упорядочить все числа n от 1 до m в порядке невозрастания значений $s(n)$. Указание: использовать только один массив длиной m .
9. Дан массив a : array [1..n] of integer. Упорядочить элементы массива a по неубыванию в соответствии со следующим критерием: элемент x считается "больше" элемента y , если первая цифра числа x больше первой цифры y .

10. Пусть для целого $n > 0$ функция $d(n)$ равна наименьшему делителю числа n , не равному 1.

Дано $m > 0$. Упорядочить все числа n от 1 до m в порядке неубывания значений $d(n)$.

Указание: использовать только один массив длиной m .

11. Дана действительная матрица размером $M \times N$. Пусть для каждой строки матрицы определена функция $f(a)$, равная сумме нечетных элементов строки a . Упорядочить (переставить) строки матрицы по неубыванию в соответствии со следующим критерием: строка a считается "больше" строки b , если $f(a) > f(b)$. Используйте алгоритм простого выбора.

12. Дана целочисленная матрица размером $M \times N$. Пусть для каждой строки матрицы определена функция $f(a)$, равная количеству простых чисел среди элементов строки a . Упорядочить (переставить) строки матрицы по невозрастанию в соответствии со следующим критерием: строка a считается "больше" строки b , если $f(a) > f(b)$. Используйте алгоритм простого выбора.

13. Дана целочисленная матрица размером $M \times N$. Пусть для каждой строки матрицы определена функция $f(a)$, равная сумме составных чисел, являющихся элементами строки a . Упорядочить (переставить) строки матрицы по неубыванию в соответствии со следующим критерием: строка a считается "больше" строки b , если $f(a) > f(b)$. Используйте алгоритм простого выбора.

14. Дана действительная матрица размером $M \times N$. Пусть для каждой строки матрицы определена функция $f(a)$, равная индексу наименьшего элемента в строке a . Упорядочить (переставить) строки матрицы по возрастанию в соответствии со следующим критерием: строка a считается "больше" строки b , если $f(a) > f(b)$. Используйте алгоритм простого выбора.

15. Дана действительная матрица размером $M \times N$. Пусть для каждой строки матрицы определена функция $s(a)$, равная разности между максимальным и минимальным элементами строки a .

Упорядочить (переставить) строки матрицы по убыванию значений функции $s(a)$. Используйте алгоритм простого выбора.

16. Даны действительные числа $c_1, \dots, c_p, d_1, \dots, d_g$ ($c_1 \leq c_2 \leq \dots \leq c_p$, $d_1 \leq d_2 \leq \dots \leq d_g$). Внести единую упорядоченность в $c_1, \dots, c_p, d_1, \dots, d_g$, получив

f_1, f_2, \dots, f_{p+g} такие, что $f_1 \leq f_2 \leq \dots \leq f_{p+g}$.

Число сравнений не должно превосходить $p+g$.

17. Дана матрица A из целых чисел размером $M \times N$. Переставить строки матрицы так, чтобы строки стали расположены по возрастанию в лексикографическом порядке.

18. Дана матрица A из целых чисел размером $M \times N$. Переставить строки матрицы так, чтобы строки стали расположены по убыванию в лексикографическом порядке.

19. Дана действительная матрица размером $M \times N$. Упорядочить (переставить) строки матрицы по убыванию значений наибольших элементов строк. Используйте алгоритм простого выбора.

20. Дана действительная матрица размером $M \times N$. Упорядочить (переставить) строки матрицы по возрастанию значений первых элементов строк. Используйте алгоритм простого выбора.

21. Дана действительная матрица размером $M \times N$. Упорядочить (переставить) строки матрицы по убыванию сумм элементов строк. Используйте алгоритм простого выбора.

22. Дана действительная матрица размером $M \times N$. Упорядочить (переставить) строки матрицы по возрастанию значений наименьших элементов строк. Используйте алгоритм простого выбора.

23. Дан массив вещественных чисел. Напишите рекурсивную программу для сортировки массива "методом пузырька".

24. Дана матрица из целых чисел A размерности $M \times N$. Упорядочить элементы матрицы так, чтобы

$A[i, j] \geq A[i, k]$, если $k \geq j$ для всех i ,

$A[i_1, j_1] \geq A[i_2, j_2]$, если $i_2 \geq i_1$ для всех j_1, j_2 , т.е. все элементы матрицы должны быть расположены по убыванию.

25. В заданном массиве подсчитать число различных элементов. Дополнительного массива не заводить.
26. Соединить конечное множество точек на плоскости замкнутой линией без самопересечений с вершинами в этих точках. (Полный перебор не делать; ответом будет порядок обхода точек плоскости). Подсказка: перейти к полярным координатам и упорядочить точки по значениям угла, а для точек с одинаковым значением угла - по расстоянию до полюса.
27. Дана действительная матрица размером $M \times N$. Упорядочить элементы матрицы по возрастанию произведений элементов строк.
28. Дана действительная матрица размером $M \times N$. Упорядочить (переставить) строки матрицы по возрастанию значений последних элементов строк. Используйте алгоритм простого выбора.
29. Дана действительная матрица размером $M \times N$. Пусть для каждой строки матрицы определена функция $s(a)$, равная сумме между максимальным и минимальным элементами строки a .
Упорядочить (переставить) строки матрицы по неубыванию значений функции $s(a)$. Используйте алгоритм простого выбора.
30. Дана целочисленная матрица размером $M \times N$. Пусть для каждой строки матрицы определена функция $f(a)$, равная сумме простых чисел среди элементов строки a . Упорядочить (переставить) строки матрицы по невозрастанию в соответствии со следующим критерием: строка a считается "больше" строки b , если $f(a) > f(b)$. Используйте алгоритм простого выбора.
31. Дан массив a : array [1..n] of integer. Упорядочить элементы массива a по неубыванию в соответствии со следующим критерием: элемент x считается "больше" элемента y , если последняя цифра числа x больше последней цифры числа y .

32. Дана целочисленная матрица размером $M \times N$. Пусть для каждой строки матрицы определена функция $f(a)$, равная количеству составных чисел, являющихся элементами строки a . Упорядочить (переставить) строки матрицы по неубыванию в соответствии со следующим критерием: строка a считается "больше" строки b , если $f(a) > f(b)$. Используйте алгоритм простого выбора.

ЛАБОРАТОРНАЯ РАБОТА 8 (рекурсия)

Примеры решения некоторых задач

1. Использовать рекурсию для нахождения цифрового корня целого числа.

Цифровой корень находится суммой через сумму цифр числа до тех пор, пока эта сумма сама не станет цифрой. Например, для числа 9999999 цифровой корень находится так:

$$9+9+9+9+9+9+9 = 63$$

$$6+3 = 9$$

Цифровой корень 9999999 равен девяти.

```
var n:longint;
function num(i:longint):integer;
var s: integer;
begin
  s:=0;
  repeat
    s:=s+n mod 10;
    n:= n div 10;
  until n=0;
  num:=s;
end;
```

```

function root(l:longint):integer;
begin
  if n<10 then root:=n
  else
    begin
      n:=num(n);
      root:=root(n);
    end;
end;

begin
  write('Введите целое число n=');
  readln(n);
  writeln('Его цифровой корень равен : ', root(n));
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- создаем описание функций num и root;
- вводим целое число n;
- вызываем рекурсивную функцию root и определяем цифровой корень числа n;
- завершаем работу программы.

Переменные:

в функции num:

n - целое число (глобальная переменная);

s - вспомогательная переменная (локальная переменная);

в функции root:

n - целое число (глобальная переменная);

в основной программе:

n - целое число (глобальная переменная).

2. Напишите рекурсивную программу вычисления суммы $\sum_{i=2}^n i^2 + i + 5$

```
var n : integer;
function sum(i : integer) : real;
begin
  if i=1 then sum:=0 else sum:=sum(i-1)+i*i+i+5;
end;
```

```
begin
writeln('Введите n');
readln(n);
writeln('Значение суммы равно - ', sum(n));
end.
```

Задания на лабораторную работу 8

1. Описать рекурсивную функцию sign (s), которая подсчитывает количество знаков препинания в строке s (знаки: '!', ',', ';', ':').
2. Напечатать в обратном порядке заданную строку.
3. Напишите программу, в которой для данной геометрической прогрессии используются две функции:
 - а) для вычисления n-го члена;
 - б) для нахождения суммы первых n членов.

4. type

```
reals = file of real;
var f : reals;
```

Опишите рекурсивную функцию sum без параметров для нахождения суммы элементов файла f.

5. Описать рекурсивную логическую функцию $\text{sum}(s, i, j)$, проверяющую, является ли симметричной часть строки s , начинающаяся i -ым и кончающаяся j -ым ее элементами.
6. Описать рекурсивную функцию $\text{letter}(s)$, которая подсчитывает количество букв в строке s .
7. Напишите рекурсивную программу для определения скалярного произведения двух векторов.
8. Напишите рекурсивную программу для сортировки массива методом "пузырька".
9. Напишите рекурсивную программу, которая n раз выводит на экран текст истории о попе и его собаке (рассказ в рассказе).
10. Найти цифровой корень числа n .
11. Рекурсивная программа. Дано n различных натуральных чисел ($n=5$). Напечатать все перестановки этих чисел.
12. Задано натуральное число n , напечатать его цифры в r -ичной системе счисления. Напишите рекурсивную программу.
13. Во входном файле задана непустая последовательность положительных вещественных чисел, за которой следует отрицательное число. Описать рекурсивную функцию sum без параметров для нахождения суммы этих положительных чисел.
14. Рекурсивная программа. Дана последовательность ненулевых целых чисел, за которой следует 0. Напечатать сначала все отрицательные числа этой последовательности, а затем все положительные (в любом порядке).
15. Описать рекурсивную функцию $\text{digits}(s)$, которая подсчитывает количество цифр в строке s .
16. Описать рекурсивную функцию $\text{root}(a, b, \text{eps})$, которая методом деления отрезка пополам находит с точностью eps корень уравнения $f(x) = 0$ на отрезке $[a, b]$. (Считать, что $\text{eps} > 0$, $a < b$, $f(a) \cdot f(b) < 0$ и $f(x)$ - непрерывная и монотонная функция на отрезке $[a, b]$).

17. type

reals = file of real;

var f : reals;

Напишите рекурсивную программу, которая печатает сначала все отрицательные элементы этого файла, а затем положительные (в любом порядке).

18. Напишите программу, в которой для данной арифметической прогрессии используются две рекурсивные функции:

а) для вычисления n -го члена арифметической прогрессии;

б) для нахождения суммы первых n членов.

19. const n=...;

type vector = array [1..n] of real;

Описать функцию $\min(x)$ для определения минимального элемента вектора x , введя вспомогательную рекурсивную функцию $\min1(k)$, находящую минимум среди последних элементов вектора x , начиная с k -го.

20. type

reals = file of real;

var f : reals;

Опишите рекурсивную функцию number без параметров для определения количества элементов файла f .

21. const n=...;

type vector = array [1..n] of real;

Описать функцию $\max(x)$ для определения максимального элемента вектора x , введя вспомогательную рекурсивную функцию $\max1(k)$, находящую максимум среди первых k элементов вектора x .

22. Описать рекурсивную функцию $\text{digits}(s)$, которая подсчитывает сумму цифр в строке s .

23. function f (n : integer) : integer;

begin

if n>100 then f:=n-10 else f:=f(f(n+11))

end;

Вычислить f(100), f(99) и f(85). Какие вообще значения принимает эта функция?

24. Дан массив a : array [1..n] of integer;

Напишите рекурсивную программу для вычисления суммы

$$\sum_{i=1}^n 1/a[i]$$

25. Напишите рекурсивную программу для вычисления суммы

$$\sum_{i=1}^n i^2$$

26. type

reals = file of real;

var f : reals;

Опишите рекурсивную функцию sum (n) для нахождения суммы

$$\sum_{i=1}^n (a[i])^n, \text{ где } a[i] - \text{ элемент файла } f$$

27. Дан массив a : array [1..n] of integer;

Напишите рекурсивную программу для вычисления произведения

$$\prod_{i=1}^n a[i]$$

28. Напишите рекурсивную программу для вычисления суммы

$$\sum_{i=1}^n 1/(i)^2$$

29. Функция F преобразования строк определяется следующим образом:

$$F(t) = \begin{cases} F(h)g, & \text{если } t = g;h, \text{ где строка } g \text{ не содержит символа '}; \\ t, & \text{если в } t \text{ нет вхождений символа '}; \end{cases}$$

Реализовать функцию F с помощью рекурсивной процедуры.

30. Написать рекурсивную функцию для нахождения биномиальных коэффициентов

$$C_n^m = \begin{cases} 1, & \text{если } m = 0, n > 0 \text{ или } n = m \geq 0 \\ 0, & \text{если } m > n \geq 0 \\ C_{n-1}^{m-1} + C_{n-1}^m & \text{в остальных случаях} \end{cases}$$

31. Функция f(n) определена для целых положительных чисел следующим образом:

$$f(n) = \begin{cases} 1, & \text{если } n = 1 \\ \sum_{i=2}^n f(n \operatorname{div} i), & \text{если } n \geq 2 \end{cases}$$

Вычислить f(k) для k=15, 16, ..., 30.

32. Описать рекурсивную функцию pow(x, n) от вещественного x (x > 0) и целого n, которая вычисляет величину xⁿ согласно формуле:

$$x^n = \begin{cases} 1, & \text{если } n = 0 \\ 1/x^{|n|}, & \text{если } n < 0 \\ x * x^{n-1} & \text{при } n > 0 \end{cases}$$

33. Функция F преобразования строк определяется следующим образом:

$$F(a) = \begin{cases} a, & \text{если } |a| \leq 2 \text{ (т.е. длина строки не больше 2)} \\ F(b)F(c), & \text{если } a = bc, |b| = |c|, |a| > 2 \\ F(bk)F(kc), & \text{если } a = bkc, |b| = |c|, |a| > 2, |k| = 1 \end{cases}$$

Отметим, что функция F может удлинять строку. Реализовать функцию F с помощью рекурсивной процедуры.

Примеры: F('123456789') = '1223344556677889'

F('12345678') = '12345678'

34. Дан массив a : array [1..n] of integer;

Напишите рекурсивную программу для вычисления суммы

$$\sum_{i=2}^n \frac{1}{a[i-1] * a[i] * a[i+1]}$$

35. Напишите рекурсивную программу для вычисления суммы

$$\sum_{i=2}^n \frac{1}{(i+1) * i * (i-1)}$$

36. Дан массив a : array [1..n] of integer;

Напишите рекурсивную программу для вычисления суммы

$$\sum_{i=2}^n \frac{1}{(a[i]-1) * a[i] * (a[i]+1)}$$

37. Функция Аккермана $Q(x, y)$ определяется следующим образом:

$$Q(0, y) = y + 1,$$

$$Q(x+1, 0) = Q(x, 1), \quad x, y > 0$$

$$Q(x+1, y+1) = Q(x, Q(x+1, y)).$$

Вычислите значения $Q(x, y)$ при $x, y \leq 3$.

38. Напишите рекурсивную процедуру для вычисления натуральной степени заданного целого числа.

39. Напишите программу, которая выводит для заданного m значение следующей функции:

$$F(n) = \begin{cases} n^2 - n, & \text{если } n < 4 \\ F(n-1 + n \bmod m) & \text{иначе} \end{cases}$$

40. Напишите программу, которая выводит для заданного m значение следующей функции:

$$F(n) = \begin{cases} n^2, & \text{если } n < 10 \\ F(n-1 + m) + F(n) & \text{иначе} \end{cases}$$

41. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \sin^n x dx = \begin{cases} -\frac{\sin^{n-1} x \cos x}{n} + \frac{n-1}{n} \int \sin^{n-2} x dx, n > 2, \\ \frac{x}{2} - \frac{1}{4} \sin 2x, n = 2, \\ -\cos x, n = 1; \end{cases}$$

42. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \cos^n x dx = \begin{cases} \frac{\cos^{n-1} x \sin x}{n} + \frac{n-1}{n} \int \cos^{n-2} x dx, n > 2, \\ \frac{x}{2} + \frac{1}{4} \sin 2x, n = 2, \\ \sin x, n = 1; \end{cases}$$

43. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \frac{dx}{\sin^n x} = \begin{cases} -\frac{1}{n-1} \cdot \frac{\cos x}{\sin^{n-1} x} + \frac{n-2}{n-1} \int \frac{dx}{\sin^{n-2} x}, n \geq 2, \\ \ln \operatorname{tg} \frac{x}{2}, n = 1, \\ x, n = 0; \end{cases}$$

44. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \frac{dx}{\cos^n x} = \begin{cases} \frac{1}{n-1} \cdot \frac{\sin x}{\cos^{n-1} x} + \frac{n-2}{n-1} \int \frac{dx}{\cos^{n-2} x}, n \geq 2, \\ \ln \operatorname{tg} \left(\frac{\pi}{4} + \frac{x}{2} \right), n = 1, \\ x, n = 0; \end{cases}$$

ЛАБОРАТОРНАЯ РАБОТА 9 (работа с файлами)

При решении задач данной лабораторной работы следует использовать такой тип данных, как файлы.

Примеры решения некоторых задач

1. type series=file of real;

Описать функцию neg(s), подсчитывающую сумму отрицательных элементов в файле s типа series.

```
type series=file of real;
```

```
var f : series;
```

```
    n,y : real;
```

```
function neg(var s : series) : real;var sum,x : real;begin    reset(s); sum:=0;
```

```
    while not eof(s) do
```

```
        begin
```

```
            read(s,x);
```

```
            if x<0 then sum:=sum+x
```

```
        end;
```

```
        neg:=sum
```

```
end;
```

```
begin
```

```
    assign (f,'~\proba.txt');
```

```
    rewrite(f);
```

```
    repeat
```

```
        write('Введите число (0 – конец ввода)- ');
```

```
        readln(y);
```

```
        writeln;
```

```

    write(f,y);
until (y=0);
n:=neg(f);
writeln('Сумма отрицательных элементов файла n= ',n:10:3);
close(f);
end.

```

Для решения задачи:

- формируем тело программы и описываем переменные;
- привяжем файл proba.txt с файловой переменной f и откроем его для записи;
- организуем цикл, в котором вводим числа и записываем их в файл до тех пор, пока не введем 0 ;
- вызываем функцию neg, в которой открываем файл для чтения, считываем последовательно элементы файла и определяем сумму отрицательных элементов файла;
- выводим значение суммы на экран.

Переменные:

основная программа:

f – файл вещественных чисел (глобальная переменная);

y – очередное число для записи в файл (глобальная переменная);

n – сумма отрицательных элементов файла (глобальная переменная);

k - счетчик цифр.

функция neg:

s – файловая переменная (локальная, формальная переменная);

x – очередной элемент файла (локальная переменная);

sum – сумма отрицательных элементов файла (локальная переменная).

2. type row=file of 0..999;

Описать логическую функцию sort(r), проверяющую, упорядочены ли по возрастанию элементы непустого файла r типа row.

type row1 = 0..999;

```
    row = file of row1;
var r : row;
    x,y : row1;
    i,k : integer;
function sort(var r : row) : boolean;
var x,y : row1; ok : boolean;
begin
    reset(r);
    read(r,y); ok:=true;
    while not eof(r) and ok do
        begin
            x:=y; read(r,y);
            ok:=x<y
        end;
        sort:=ok
    end;
begin
    write('Введите количество элементов файла');
    readln(k);
    assign(r,'~\12.dat');
    rewrite(r);
    for i:= 1 to k do
        begin
            writeln('Введите ',i,'-ый элемент файла');
            read(x);
            write(r,x);
        end;
        if sort(r) then writeln('Упорядочены по возрастанию')
            else writeln('Не упорядочены по возрастанию');
    end.
end.
```

3. type reals=file of real;

Описать функцию less(f) от непустого файла f типа reals, которая подсчитывает количество элементов файла f, меньших среднего арифметического всех элементов этого файла.

```
type reals = file of real;
```

```
var f1 : reals;
```

```
    a : real;l,i : integer;
```

```
function less(var f : reals) : integer;
```

```
var k : integer; x,s : real;
```

```
begin                                     {подсчет среднего арифметического:}
```

```
    reset(f); k:=0; s:=0;
```

```
    repeat
```

```
        read(f,x);
```

```
        k:=k+1;
```

```
        s:=s+x
```

```
    until eof(f);
```

```
    s:=s/k;
```

```
                                     {новый просмотр f и подсчет элементов <s:}
```

```
    reset(f); k:=0;
```

```
    repeat
```

```
        read(f,x);
```

```
        if x<s then k:=k+1
```

```
    until eof(f);
```

```
    less:=k
```

```
end;
```

```
begin
```

```
    assign(f1,'c:\tp\work\23.dat');
```

```
    rewrite(f1);
```

```
    for i:=1 to 10 do
```

```

begin
  readln(a);
  write(f1,a);
end;
l:=less(f1);
writeln('l=', l);
end.

```

Задания на лабораторную работу 9.

1. type text = file of char;

Описать логическую функцию less(w1,w2), проверяющую, предшествует ли лексикографически текст w1 тексту w2.

Программа: введите файлы w1 и w2 (последние элементы в файлах - '.' (точка)) и вычислите значение функции less(w1,w2).

2. type ряд = file of integer;

Описать процедуру p(f,g) от двух файлов типа ряд, которая в пустой файл f переписывает положительные элементы файла g.

Программа: введите файл g (последний элемент равен 0). Выполните процедуру p(f,g). Распечатайте файл f.

3. type fr = file of real;

Описать процедуру предполс(f), значением которой является предпоследний элемент файла f, имеющего тип fr и содержащего не менее двух элементов.

Программа: введите файл f (последний элемент = 0) и вычислите функцию предполс(f).

4. type rad = file of integer;

Описать процедуру p(f,g) от двух файлов типа rad, которая из файла f переписывает в пустой файл g сначала все положительные числа, а потом все отрицательные.

Программа: введите файл f (последний элемент равен 0). Выполните процедуру $p(f,g)$. Распечатайте файл g .

5. type seria = file of integer;

Описать функцию $prod(s)$ с вещественным значением, подсчитывающую произведение ненулевых элементов файла s .

Программа: введите файл s (последний элемент равен 0) и напечатайте значение функции $prod(s)$.

6. type seria = file of integer;

var s:seria;

Описать функцию $max(s)$, вычисляющую значение максимального элемента файла s .

Программа: введите файл s (последний элемент равен 0) и напечатайте значение функции $max(s)$.

7. Описать логическую функцию $check(s)$ для текстового файла s , проверяющую, содержит ли файл s латинские буквы.

Программа: введите файл s (последний элемент в файле '.') и вычислите значение функции $check(s)$.

8. Для файла s , состоящего из целых чисел, определите процедуру, которая находит значения максимального и минимального элементов файла s .

9. type rad = file of integer;

Описать процедуру $p(g,f,h)$ от трех файлов типа rad , которая из файла g переписывает все положительные числа в пустой файл f , все отрицательные в пустой файл h .

Программа: введите файл g (последний элемент равен 0). Выполните процедуру $p(g,f,h)$. Распечатайте файлы f и h .

10. type rad = file of 1..maxint;

Описать процедуру $prim(f,n)$, записывающую в файл f все простые числа 2,3,5,7,11,13,17..., не превосходящие целого положительного числа n .

Программа: введите n . Выполните процедуру $prim(f,n)$. Распечатайте файл f .

11. type ряд = file of 1..maxint;

Описать процедуру fib(f,n), записывающую в ряд f все числа Фибоначчи (1,1,2,3,5,8,...), не превосходящие целого положительного числа n.

Программа: введите n. Выполните процедуру fib(f,n). Распечатайте файл f.

12. type ряд = file of integer;

Описать логическую функцию check(r), проверяющую, все ли элементы файла r неотрицательны.

Программа: введите файл r (последний элемент = 0) и напечатайте значение функции check(r).

13. type ряд = file of integer;

Описать процедуру append(f,g,h) от трех файлов типа ряд, которая записывает в файл f сначала все элементы файла g, а затем все элементы файла h.

Программа: введите два файла g и h. Выполните процедуру append(f,g,h).

Напечатайте ряд f.

14. type seria = file of integer;

var s:seria;

Описать функцию neg(s), подсчитывающую сумму отрицательных элементов в файле s.

Программа: введите файл s (последний элемент = 0) и напечатайте значение функции neg(s).

15. type fr = file of real;

Описать функцию s3(f), значением которой является сумма последних трех элементов файла f, имеющего тип fr и содержащего не менее 3 элементов.

Программа: введите файл f (последний элемент = 0) и вычислите функцию s3(f).

16. type seria = file of integer;

var s:seria;

Описать функцию poz(s), подсчитывающую сумму положительных элементов в файле s.

Программа: введите файл s (последний элемент = 0) и напечатайте значение функции $\text{roz}(s)$.

17. type $\text{fr} = \text{file of real}$;

Описать логическую функцию $\text{mid}(f,m)$, которая определяет, имеет ли файл f типа fr нечетную длину, и, если имеет, присваивает параметру m средний элемент этого файла.

Программа: введите файл f (последний элемент = 0). Если функция $\text{mid}(f,m)$ - истина, то напечатайте значение m .

18. type $\text{text} = \text{file of char}$;

Описать процедуру $\text{digits}(s,t)$, которая записывает в текст t все цифры из строки s .

Программа: введите строку s . Выполните процедуру $\text{digits}(s,t)$ и распечатайте элементы файла t .

19. type $\text{rad} = \text{file of integer}$;

Описать процедуру $p(f,g)$ от двух файлов типа rad , которая из файла f переписывает в пустой файл g все положительные числа.

Программа: введите файл f (последний элемент равен 0). Выполните процедуру $p(f,g)$. Распечатайте файл g .

20. type $\text{ряд} = \text{file of } 0\dots 999$;

Описать логическую функцию $\text{упор}(r)$, проверяющую, упорядочены ли по возрастанию элементы непустого ряда r .

Программа: введите ряд r (последний элемент = 999) и напечатайте значение функции $\text{упор}(r)$.

21. Type $\text{reals} = \text{file of real}$;

Описать функцию $\text{less}(f)$ от непустого файла f типа reals , которая подсчитывает количество элементов файла f , меньших среднего арифметического всех элементов этого файла.

Программа: введите файл f (последний элемент = 0). Напечатайте значение $\text{less}(f)$.

22. type seria = file of integer;

var s:seria;

Описать функцию $roz(s)$, подсчитывающую количество положительных чисел в файле s .

Программа: введите файл s (последний элемент = 0) и напечатайте значение функции $roz(s)$.

23. Описать процедуру $letter(s,t)$, которая записывает в текстовый файл t все латинские буквы из строки s .

Программа: введите строку s . Выполните процедуру $letter(s,t)$ и распечатайте элементы файла t .

24. Type text = file of char;

Описать логическую функцию $eg(t1,t2)$, проверяющую файлы $t1$ и $t2$ на равенство.

Программа: введите два файла $t1$ и $t2$ (последний элемент в файлах '.') и вычислите значение функции $eg(t1,t2)$.

25. Описать процедуру $red(s,t)$, которая переписывает в файл t все символы из строки s , исключая пробелы и знаки препинания.

Программа: введите строку s , Выполните процедуру $red(s,t)$ и распечатайте элементы текстового файла t .

26. type seria = file of integer;

var s:seria;

Описать функцию $sum(s)$, вычисляющую среднее арифметическое элементов файла s .

Программа: введите файл s (последний элемент = 0) и напечатайте значение функции $sum(s)$.

27. type reals = file of real;

Описать процедуру $p(f,g,h)$ от трех файлов типа reals, которая переписывает из файла f в непустой файл g все элементы, меньшие среднего арифметического всех элементов файла f и в непустой файл h записывает все остальные числа.

Программа: введите файл f (последний элемент = 0). Выполните процедуру $p(f,g,h)$. Распечатайте файлы g и h .

28. Даны два файла целых чисел f_1 и f_2 . Найдите в файле f_2 число, самое близкое по величине к минимальному значению файла f_1 .

29. Дан файл целых чисел. Определите, где больше нулей на четных или нечетных позициях.

30. Дан файл целых чисел. Создайте файл вещественных чисел, компонентами которого будут средние арифметические значения каждой пятерки подряд идущих чисел. Последняя группа может содержать меньше пяти значений.

31. Пусть дан файл целых чисел. Определите длину наибольшего интервала возрастания и среднее арифметическое чисел этого интервала.

32. Дан файл целых чисел. Сформируйте два выходных файла целых чисел, записывая поочередно числа, образующие возрастающую последовательность, то в один то в другой файл. Например:

- исходный файл: $f_1: 2\ 4\ 3\ 8\ 15\ -1\ -3\ 10\ 1$;

- первый файл-результат: $f_2: 2\ 4\ -1\ 1$;

- второй файл-результат: $f_3: 3\ 8\ 15\ -3\ 10$.

33. Дан файл целых чисел. Определите, больше ли минимальное значение первого файла чем максимальное значение второго файла.

ЛАБОРАТОРНАЯ РАБОТА 10(динамические переменные)

Примеры решения некоторых задач

1. type telem='a'..'z'.

```
list=^node;
node= record
    info:telem;
    next:list
end;
```

Пусть E1 и E2 - данные типа telem.

Описать функцию или процедуру, которая :

а) заменяет в списке L все вхождения E1 и E2;

б) проверяет, упорядочены ли элементы списка L по алфавиту

а)

```
type telem='a'..'z';
```

```
list=^node;
node= record
    info : telem;
    next : list
end;
```

```
var s,l : list;
```

```
    x,e,e1 : telem;
```

```
    n,i : integer;
```

```
procedure change (l : list; e,e1: telem);
```

```
var p:list;                                {ссылка на очередное звено}
```

```
begin
```

```
p:=L;
while p<>nil do
begin
    if p^.info=e then p^.info:=e1;
    p:=p^.next           {переход к следующему звену}
end;
end;
procedure out_spisok(l : list);           {ВЫВОДИТ СПИСОК НА ЭКРАН}
begin
    while l<> nil do
    begin
        s:=l^.next;
        write(l^.info, ' ');
        l:=s;
    end;
    writeln;
end;

begin
                                     {формируем список}
s:=nil;
writeln('Введите количество элементов списка');
readln(n);
for i:=1 to n do
begin
    new(l);
    l^.next:=s;
    readln(x);
    l^.info:=x;
    s:=l;
```

```

end;

                                {выводим список на экран}
writeln('Введенный список');
out_spisok(l);

                                {заменяем элемент E на E1}
writeln('Введите элемент, который Вы хотите заменить');
readln(e);
writeln('Введите элемент, на который Вы хотите заменить');
readln(e1);
change (l,e,e1);
writeln('Полученный список');
out_spisok(l);

                                {освобождаем динамическую память}

while l<> nil do
begin
    s:=l^.next;
    dispose(l);
    l:=s;
end;
end.

```

б)

```

type telem='a'..'z';
    list=^node;
    node= record
        info : telem;
        next : list
    end;
var s,l : list;

```

138

```
x : telem;  
n,i : integer;
```

```
function sort(l : list) : boolean;
```

```
var p,q : list;                               {ссылка на пару соседних  
звеньев}
```

```
    ok : boolean;
```

```
begin
```

```
    ok:=true;
```

```
    p:=L;                                       {nil или ссылка на 1-звено}
```

```
    if p<>nil then
```

```
        begin q:=p^.next;                       {nil или ссылка на 2-е звено}
```

```
            while (q<>nil) and ok do
```

```
                begin
```

```
                    ok:=p^.info<=q^.info;
```

```
                    p:=q; q:=q^.next           {переход к след. паре}
```

```
                end
```

```
            end;
```

```
        sort:=ok
```

```
end;
```

```
procedure out_spisok(l:list);
```

```
begin
```

```
    while l<> nil do
```

```
        begin
```

```
            s:=l^.next;
```

```
            write(l^.info, ' ');
```

```
            l:=s;
```

```
        end;
```

```
        writeln;
```

```
end;

begin
    {формируем список}

    s:=nil;
    writeln('Введите количество элементов списка');
    readln(n);
    for i:=1 to n do
    begin
        new(l);
        l^.next:=s;
        readln(x);
        l^.info:=x;
        s:=l;
    end;

    {выводим список на экран}

    writeln('Введенный список');
    out_spisok(l);
    if sort(l) then writeln('Список отсортирован по алфавиту')
        else writeln('Список не отсортирован по алфавиту');

        {освобождаем динамическую память}

    while l<> nil do
    begin
        s:=l^.next;
        dispose(l);
        l:=s;
    end;
end.
```

```
2. type telem=...;
```

```
list=^node;
```

```
node=record
```

```
info : telem;
```

```
next : list
```

```
end;
```

Описать процедуру, которая вставляет в список L новый элемент E1 перед первым вхождением элемента E, если E входит в L;

```
type telem=0..999;
```

```
list=^node;
```

```
node= record
```

```
info : telem;
```

```
next : list
```

```
end;
```

```
var s,l : list;
```

```
x,e,e1 : telem;
```

```
n,i : integer;
```

```
procedure insert( l : list; e,e1 : telem);
```

```
var p,q : list; eq : boolean;
```

```
begin
```

```
{ поиск звена с E:}
```

```
p:=L; eq:=false;
```

```
while (p<>nil) and not eq do
```

```
if p^.info=e then eq:=true
```

```
else p:=p^.next;
```

```
if eq then
```

```
{ вставка E1 перед E}
```

```
begin
```

```
{внимание-трюк:
```

```
запись E1 в звено p
```

```

за звеном p:}
    p^.info:=e1; new(q); q^.info:=e;
    q^.next:=p^.next; p^.next:=q
end;
end;

```

```

procedure out_spisok(l : list);

```

```

begin

```

```

    while l<> nil do

```

```

        begin

```

```

            s:=l^.next;

```

```

            write(l^.info, ' ');

```

```

            l:=s;

```

```

        end;

```

```

        writeln;

```

```

end;

```

```

begin

```

{формируем список}

```

    s:=nil;

```

```

    writeln('Введите количество элементов списка');

```

```

    readln(n);

```

```

    for i:=1 to n do

```

```

        begin

```

```

            new(l);

```

```

            l^.next:=s;

```

```

            readln(x);

```

```

            l^.info:=x;

```

```

            s:=l;

```

```
end;
```

```
{выводим список на экран}
```

```
writeln('Введенный список');
```

```
out_spisok(l);
```

```
writeln('Введите элемент, перед которым Вы хотите вставить число');
```

```
readln(e);
```

```
writeln('Введите элемент, который Вы хотите вставить в список');
```

```
readln(e1);
```

```
insert(l,e,e1);
```

```
writeln('Полученный список');
```

```
out_spisok(l);
```

```
{освобождаем динамическую память}
```

```
while l<> nil do
```

```
begin
```

```
s:=l^.next;
```

```
dispose(l);
```

```
l:=s;
```

```
end;
```

```
end.
```

3. Описать процедуру, которая удаляет из непустого списка l последний элемент.

```
type list=^node;
```

```
node= record
```

```
info : integer;
```

```
next : list
```

```
end;
```

```
var s,l : list;
```

```
x : integer;
n,i : integer;
```

```
procedure del(var l : list);
```

```
var p,q : list;
```

```
begin
```

```
  if l=nil then {удалять нечего}
```

```
    else if l^.next=nil {в списке один элемент} then
```

```
      begin dispose(l);l:=nil end
```

```
    else begin
```

```
      {поиск предпослед.(p) и
```

```
послед.(q) звеньев: }
```

```
      p:=l; q:=p^.next;
```

```
      while q^.next<>nil do
```

```
        begin p:=q; q:=q^.next end;
```

```
      {удаление последнего
```

```
звена: }
```

```
      dispose(q); p^.next:=nil
```

```
    end;
```

```
end;
```

```
procedure out_spisok(l : list);
```

```
begin
```

```
  while l<> nil do
```

```
    begin
```

```
      s:=l^.next;
```

```
      write(l^.info, ' ');
```

```
      l:=s;
```

```
    end;
```

```
  writeln;
```

144

end;

begin

{формируем список}

s:=nil;

writeln('Введите количество элементов списка');

readln(n);

for i:=1 to n do

begin

new(l);

l^.next:=s;

readln(x);

l^.info:=x;

s:=l;

end;

{выводим список на экран}

writeln('Введенный список');

out_spisok(l);

del(l);

writeln('Полученный список');

out_spisok(l);

{освобождаем динамическую память}

while l<> nil do

begin

s:=l^.next;

dispose(l);

l:=s;

end;

end.

4. Описать рекурсивную функцию или процедуру, которая:

а) определяет, входит ли элемент E в список L;

б) удаляет из списка L первое вхождение элемента E, если такое есть;

а)

```
type list=^node;
```

```
  node= record
```

```
    info : integer;
```

```
    next : list
```

```
  end;
```

```
var s,l : list;
```

```
  x,e : integer;
```

```
  n,i : integer;
```

```
function memb(l : list; e : integer) : boolean;
```

```
var l1 : list;
```

```
begin
```

```
  if l=nil then memb:=false
```

```
    else if l1^.info=e then memb:=true
```

```
      else memb:=memb(l1^.next,e)
```

```
end;
```

```
procedure out_spisok(l : list);
```

```
begin
```

```
  while l<> nil do
```

```
    begin
```

```
      s:=l^.next;
```

```
      write(l^.info, ' ');
```

```
      l:=s;
```

146

end;

writeln;

end;

begin

{формируем список}

s:=nil;

writeln('Введите количество элементов списка');

readln(n);

for i:=1 to n do

begin

new(l);

l^.next:=s;

readln(x);

l^.info:=x;

s:=l;

end;

{выводим список на экран}

writeln('Введенный список');

out_spisok(l);

writeln('Введите интересующий Вас элемент');

readln(e);

if not(memb(l,e)) then writeln('Элемент ', e, ' входит в список')

else writeln('Элемент ', e, ' не входит в список');

{освобождаем динамическую

память}

while l<> nil do

begin

s:=l^.next;

dispose(l);

```

l:=s;
end;
end.

```

б)

```

type list=^node;

```

```

    node= record

```

```

        info : integer;

```

```

        next : list

```

```

    end;

```

```

var s,l : list;

```

```

    x,e : integer;

```

```

    n,i : integer;

```

```

procedure delete(var l : list; e :integer);

```

```

var p : list;

```

```

begin

```

```

    if l<>nil then

```

```

        begin

```

```

            if l^.info=e then {удалить первое звено}

```

```

                begin p:=l; l:=L^.next; dispose(p) end

```

```

            else {удалить E из "хвоста" списка и записать
                в 1-е звено ссылку на измененный хвост":}

```

```

                delete(l^.next,e)

```

```

        end;

```

```

end;

```

```

procedure out_spisok(l : list);

```

```

begin

```

```
while l<> nil do
begin
s:=l^.next;
write(l^.info, ' ');
l:=s;
end;
writeln;
end;
```

```
begin
```

{формируем список}

```
s:=nil;
writeln('Введите количество элементов списка');
readln(n);
for i:=1 to n do
begin
new(l);
l^.next:=s;
readln(x);
l^.info:=x;
s:=l;
end;
```

{выводим список на экран}

```
writeln('Введенный список');
out_spisok(l);
writeln('Введите интересующий Вас элемент');
readln(e);
delete(l,e);
writeln('Полученный список');
out_spisok(l);
```

{освобождаем динамическую память}

```
while l<> nil do
begin
s:=l^.next;
dispose(l);
l:=s;
end;
end.
```

Задания на лабораторную работу 10

1. Используйте линейные списки для хранения последовательности чисел.

Опишите процедуру

или функцию, которая:

а) переносит в начало непустого списка его последний элемент;

б) добавляет в конец списка L1 все элементы списка L2;

2. Используйте линейные списки для хранения последовательности чисел.

Опишите процедуру

или функцию, которая:

а) вставляет в список L за первым вхождением элемента E все элементы списка L1, если E входит в L.

б) удаляет из списка L все элементы, которые есть в списке L1.

3. Используйте линейные списки для хранения последовательности чисел.

Опишите процедуру, которая удаляет:

а) из списка второй элемент, если такой есть;

б) из непустого списка последний элемент;

4. Используйте линейные списки для хранения последовательности чисел.

Опишите процедуру, которая удаляет:

а) из списка первый отрицательный элемент, если такой есть;

б) из списка все отрицательные элементы.

5. Используйте линейные списки для хранения последовательности строк.

Опишите функцию, подсчитывающую количество строк - элементов списка, которые:

- а) начинаются и оканчиваются одним и тем же символом;
- б) начинаются с того же символа, что и следующая строка;

6. Используйте линейные списки для хранения последовательности строк.

Опишите функцию, подсчитывающую количество строк - элементов списка, которые:

- а) совпадают с последней строкой.
- б) напишите процедуру, которая удаляет вторую строку из списка.

7. Используйте линейные списки для хранения последовательности чисел.

Опишите процедуру или функцию, которая:

- а) проверяет на равенство списки L1 и L2;
- б) проверяет, есть ли в списке L хотя бы два одинаковых элемента;
- в) переносит в конец непустого списка его первый элемент.

8. Используя представление последовательности чисел в виде линейного списка, напишите программу, решающую задачу: "Объедините две упорядоченные последовательности в одну упорядоченную последовательность (сортировка слиянием)".

9. Используя представление последовательности чисел в виде линейного списка, напишите программу, сортировки этой последовательности при помощи алгоритма простого обмена.

10. Используйте линейные списки для хранения последовательности чисел.

Опишите процедуру или функцию, которая:

- а) для данного списка L создает список L1, содержащий те же элементы, но в обратном порядке;
- б) для данного элемента E, входящего в список, оставляет только его первое вхождение;
- в) то же, что и б), но оставляется последнее вхождение.

11. Используйте линейные списки для хранения последовательности вещественных чисел. Опишите процедуру или функцию, которая:
- находит среднее арифметическое элементов непустого списка;
 - заменяет в списке все вхождения элемента E_1 на элемент E_2 ;
12. Используйте линейные списки для хранения последовательности вещественных чисел. Опишите процедуру или функцию, которая:
- меняет местами первый и последний элементы непустого списка;
 - удаляет последний элемент списка.
13. Используйте линейный список для представления многочлена от переменной x , упорядоченного по степеням x . Напишите программу для сложения двух многочленов от переменной x (удобно воспользоваться сортировкой слиянием).
14. Используйте линейный список для представления многочлена от переменной x , упорядоченного по степеням x . Напишите программу для дифференцирования многочлена.
15. Используйте линейные списки для хранения последовательности чисел. Опишите процедуру, которая вставляет:
- новый элемент E после первого элемента непустого списка;
 - новый элемент E_1 за каждым вхождением элемента E .
16. Используйте линейные списки для хранения последовательности чисел. Опишите процедуру, которая вставляет:
- новый элемент E_1 перед первым вхождением элемента E , если E входит в список;
 - новый элемент E перед последним элементом непустого списка.
17. Используйте представление последовательности строк в виде линейного списка и опишите:
- процедуру ПЕРЕСТАНОВКА(L, i, j), меняющую местами i -ю и j -ю строки списка L ;
 - процедуру ЗАМЕНА(L, i, j), заменяющую i -ю строку списка L на копию j -й строки.

18. Используйте представление последовательности строк в виде линейного списка и опишите:

а) процедуру ДОБАВИТЬ(L, i, j), добавляющую после i -ой строки списка L копию j -й строки;

б) процедуру УДАЛИТЬ(L, i), удаляющую i -ю строку из списка L .

19. Используя представление последовательности чисел в виде линейного списка, напишите программу сортировки этой последовательности при помощи алгоритма простыми включениями.

20. Используя представление последовательности чисел в виде линейного списка, напишите программу сортировки этой последовательности при помощи алгоритма простого выбора.

21. Используйте линейный список для хранения очень длинного целого числа.

Дано натуральное n . Напишите программу для вычисления 2^n .

22. Используйте линейный список для представления стека для решения следующей задачи. Пусть дан файл, состоящий из чисел. За один просмотр файла и без использования дополнительных файлов напечатать элементы файла в следующем порядке: сначала все числа, меньшие a , затем все числа $\geq a$, причем числа в каждой группе должны печататься в обратном порядке относительно исходного (a - заданное число).

Пример: элементы файла - 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

$$a = 6$$

Порядок при печати: 5 4 3 2 1 10 9 8 7 6

23. Используя стек - представленный в виде линейного списка – решить следующую задачу (решение описать в виде процедуры). В текстовом файле записан текст, сбалансированный по круглым скобкам. Требуется для каждой пары соответствующих открывающей и закрывающей скобок напечатать номера их позиций в тексте, упорядочив пары номеров в порядке возрастания номеров позиций закрывающих скобок.

Например, для текста $a + (45 - f(x) * (b - c))$ надо напечатать 8 10; 12 16; 3

17;

24. Используйте линейный двунаправленный список для представления многочлена относительно переменной x . Многочлены Чебышева (относительно x) n -го порядка определяются следующим образом:

$$T_0(x) = 1, T_1(x) = x$$

$$T_n(x) = 2 * x * T_{n-1}(x) - T_{n-2}(x) \quad (n=2,3,\dots)$$

Напишите программу для получения n -го многочлена Чебышева.

25. Дан файл целых чисел, содержащий не менее двух элементов. Выбрав для представления данных подходящую списковую структуру, напечатайте в обратном порядке все числа между наибольшим и наименьшим членами этой последовательности.

26. Реализуйте двусвязные списки для представления очереди. Реализуйте следующие операции над очередью:

а) очистить очередь (создать пустую очередь);

б) проверить, является ли очередь пустой;

в) добавить в конец очереди элемент;

г) удалить из очереди первый элемент.

27. Напечатать в порядке возрастания все неправильные несократимые дроби со знаменателями, меньшими заданного натурального числа p . Массивов не заводить.

28. Дано натуральное n . Программа: напечатать в порядке возрастания все рациональные числа

вида a / b , где $a, b > 0, b \leq n, 1 < a/b \leq 2$, a и b – взаимно просты.

Используйте сортировку с помощью дерева.

29. Дано натуральное n . Программа: напечатать в порядке убывания все рациональные числа вида

a / b , где $a, b > 0, b \leq n, 0 < a/b \leq 1$, a и b - взаимно просты. Используйте сортировку с помощью дерева.

ЛАБОРАТОРНАЯ РАБОТА 11(графика)

Примеры решения некоторых задач

1. Построить семейство одинаковых окружностей, центры которых лежат на вертикально вращающемся отрезке, верхний конец которого закреплен.

```
uses crt,graph;
var drive,mode,x,y,i,t,y0:integer;
begin
  drive:=detect;
  initgraph(drive,mode,'c:\tp\bgi');
  setfillstyle(1,1);
  floodfill(1,4,1);
  t:=-4;
  y0:=10;
  setcolor(16);
  for i:=1 to 150 do
  begin
    t:=t+2;
    y0:=y0+3;
    x:=getmaxx div 2 + trunc(cos(t/10)*i);
    y:=y0 - trunc(sin(t/10)*i);
    setfillstyle(1,14);
    fillellipse(x,y,20,20);
    delay(100);
  end;
repeat
until keypressed;
closegraph;
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- иницилируем модуль graph;
- устанавливаем начальные значения радиуса, координаты центра;
- организуем цикл, в котором закрашиваем круги с все большим радиусом до тех пор, пока не будет нажата любая клавиша.

Переменные:

x, y - координаты центра очередного маленького круга;

y0 - смещение кругов по вертикале;

i - переменная цикла;

t - угол поворота;

drive - тип графического драйвера;

mode - режим работы графического адаптера.

2. Построить движущиеся изображения НЛО на фоне звездного неба.

a)

```
uses crt,graph;
```

```
var drive,mode,x,y,i,r:integer;
```

```
begin
```

```
  r:=40;
```

```
  x:=r*5;
```

```
  y:=r*2;
```

```
  drive:=detect;
```

```
  initgraph(drive,mode,"");
```

```
  setcolor(3);
```

```
  for i:=1 to 600 do
```

```
    putpixel(random(i),random(i),i);
```

```
  repeat
```

```
    ellipse(x,y,0,360,r,(r div 3)+2);
```

```
    ellipse(x,y-4,190,357,r,r div 3);
```

```
line(x-17,y-16,x-25,y-22);
line(x+17,y-16,x+25,y-22);
circle(x+25, y-25,2);
circle(x-25, y-25,2);
setfillstyle(1,3);
floodfill(x+1,y+4,3);
delay(150);
cleardevice;
for i:=1 to 600 do
    putpixel(random(i),random(i),i);
x:=x+random(10);
y:=y+random(10);
until (keypressed);
closegraph;
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- инициализируем модуль graph;
- организуем безусловный цикл по переменной i и рисуем звездное небо;
- организуем цикл до тех пор, пока не будет нажата любая клавиша;
- в этом цикле рисуем НЛО с помощью двух эллипсов, двух линий и двух маленьких кружочков, держим на экране, затем стираем изображение процедурой CLEARDEVICE;
- опять рисуем звездное небо;
- определяем случайным образом координаты следующего изображения НЛО;
- после нажатия любой клавиши закрываем графический режим.

Переменные:

- x, y - случайные координаты;
- r - радиус;

`i` - переменная цикла;
`drive` - тип графического драйвера;
`mode` - режим работы графического адаптера.

б) Та же программа может быть написана с использованием пары процедур `GETIMAGE(lx,ly,rx,ry,saucer^)` и `PUTIMAGE(x,y,saucer^,xorput)`.

Процедура `GETIMAGE(lx,ly,rx,ry,saucer^)` помещает изображение в буфер, а `PUTIMAGE(x,y,saucer^,xorput)` выводит в заданное место изображение.

Параметр `xorput` определяет способ вывода на экран - исключающее ИЛИ.

Например, операторами

```
GETIMAGE(lx,ly,rx,ry,saucer^);
```

```
READLN;
```

```
PUTIMAGE(x,y,saucer^,xorput);
```

мы выводим изображение на экран и после нажатия любой клавиши стираем его.

Можно использовать другие способы вывода изображения на экран,

например:

`NORMALPUT` - стирает часть экрана, и на это место выводит изображение;

`NOTPUT` - делает тоже самое, но изображение инвертируется;

`ORPUY` - дописывает новое изображение.

```
uses crt,graph;
```

```
var drive,mode,x,y,i,r,rx,ry,lx,ly,size:integer;
```

```
    saucer:pointer;
```

```
begin
```

```
    r:=20;
```

```
    x:=r*5;
```

```
    y:=r*2;
```

```
    drive:=detect;
```

```
    initgraph(drive,mode,'c:\tp\bgi');
```

```
setcolor(3);
ellipse(x,y,0,360,r,(r div 3)+2);
ellipse(x,y-4,190,357,r,r div 3);
line(x-17,y-16,x-25,y-22);
line(x+17,y-16,x+25,y-22);
circle(x+25, y-25,2);
circle(x-25, y-25,2);
setfillstyle(1,3);
floodfill(x+1,y+4,3);
lx:=x-r-30;
ly:=y-30;
rx:=x+r+30;
ry:=y+r div 3+30;
size:=imagesize(lx,ly,rx,ry);
getmem(saucer,size);
getimage(lx,ly,rx,ry,saucer^);
readln;
putimage(lx,ly,saucer^,xorput);
for i:=1 to 600 do
  putpixel(random(i),random(i),i);
repeat
  putimage(x,y,saucer^,xorput);
  delay(150);
  putimage(x,y,saucer^,xorput);
  x:=x+random(10);
  y:=y+random(10);
until (keypressed);
readln;
closegraph;
end.
```

Для решения задачи:

- формируем тело программы и описываем переменные;
- иницилируем модуль graph;
- рисуем НЛО с помощью двух эллипсов, двух линий и двух маленьких кружочков, держим на экране, затем стираем изображение процедурой CLEARDEVICE;
- определяем размер буфера и помещаем в него изображение;
- организуем безусловный цикл по переменной i и рисуем звездное небо;
- организуем цикл до тех пор, пока не будет нажата любая клавиша;
- опять рисуем звездное небо;
- в этом цикле помещаем изображение из буфера на экран, держим его на экране, затем стираем изображение;
- определяем случайным образом координаты следующего изображения НЛО;
- после нажатия любой клавиши закрываем графический режим.

Переменные:

x,y - случайные координаты;

r - радиус;

i - переменная цикла;

saucer - указатель буфера хранения изображения;

drive - тип графического драйвера;

mode - режим работы графического адаптера.

3. Построить астроида-кривую, заданную параметрическим уравнением

$$x = b \cos^3(t), \quad y = b \sin^3(t) \quad t \text{ принадлежит интервалу } [0, 2\pi]$$

```
uses crt,graph;
```

```
const B=200;
```

```
var drive,mode,i:integer;
```

```
    x,y,t:real;
```

```
begin
```

```

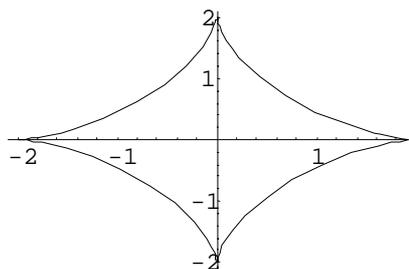
drive:=detect;
initgraph(drive,mode,");      {инициализация графического режима}
setcolor(3);
line(10,240,630,240);        {ось X}
line(320,10,320,470);        {ось Y}
line(630,240,610,235);      {стрелки на оси X}
line(630,240,610,245);
line(320,10,315,30);         {стрелки на оси Y}
line(320,10,325,30);
t:=0;
while t<=2*pi do
begin
    x:=b*sqr(cos(t))*cos(t);   {рассчитываем по формуле координаты
точек}
    y:=b*sqr(sin(t))*sin(t);
    x:=320+x;                  {рисуем в центре экрана}
    y:=240+y;
    putpixel(round(x),round(y),random(15)); {рисуем точку с координатами x,
y}
    t:=t+0.001;
end;
repeat until keypressed;
closegraph;
end.

```

Задания на лабораторную работу 11

1. Построить астроида-кривую, заданную параметрическим уравнением

$$x = b \cos^3(t), \quad y = b \sin^3(t) \quad t \text{ принадлежит интервалу } [0, 2\pi]$$

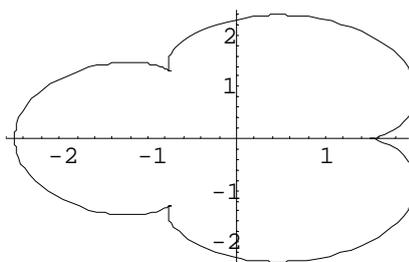


2. Построить эпициклоиду-кривую, заданную параметрическим уравнением

$$x = (a+b) \cos(t) - a \cos((a+b)t/a)$$

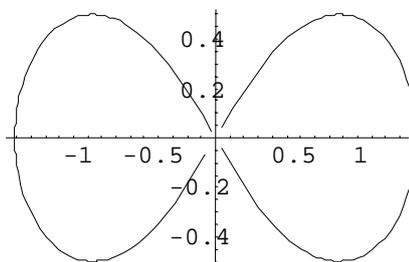
$$y = (a+b) \sin(t) - a \sin((a+b)t/a) \quad a > 0, b > 0$$

b/a - целое положительное число t принадлежит интервалу $[0, 2\pi]$



3. Построить лемнискату-кривую, уравнение которой в полярных координатах

$$\rho = a \sqrt{2 \cos(2\varphi)} \quad a > 0$$

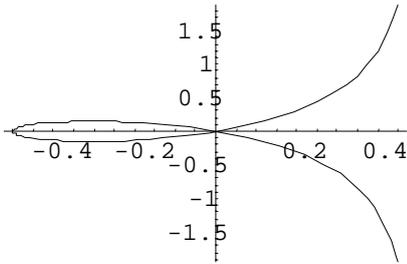


4. Построить строфоиду-кривую, заданную параметрическим уравнением

$$x = a(t^2 - 1)/(t^2 + 1)$$

$$y = at(t^2 - 1)/(t^2 + 1) \quad a > 0$$

t принадлежит интервалу $[-\infty, +\infty]$



5. Построить кривую - "улитку Паскаля" по заданному параметрическому уравнению

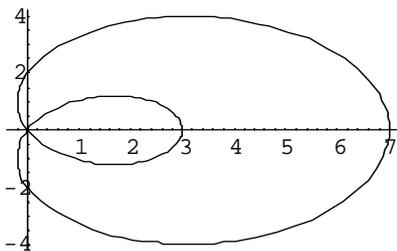
$$x = a \cos^2(t) + b \cos(t)$$

$$y = a \cos(t) \sin(t) + b \sin(t) \quad a > 0, b > 0$$

t принадлежит интервалу $[0, 2\pi]$

Рассмотреть три случая:

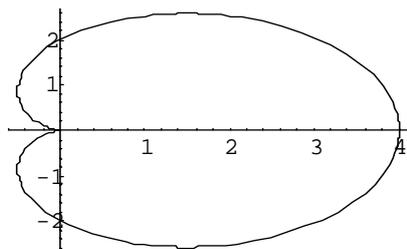
1) $b \geq 2a$ 2) $a < b < 2a$ 3) $a > b$



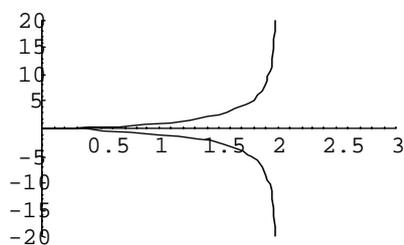
6. Соединить конечное множество точек на плоскости замкнутой ломаной линией без самопересечений с вершинами в этих точках. (Полный перебор не делать; ответом будет порядок обхода точек плоскости).

Подсказка: перейти к полярным координатам и упорядочить точки по значениям угла, а для точек с одинаковым значением угла - по расстоянию до полюса.

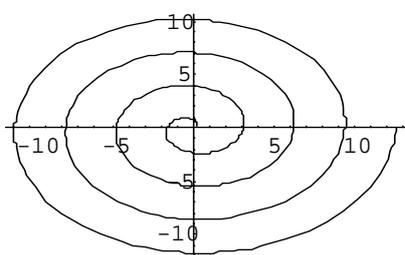
7. Построить кривую кардиоиду по заданному параметрическому уравнению $x = a \cos t (1 + \cos t)$, $y = a \sin t (1 + \cos t)$, $a > 0$, $t \in [0, 2\pi)$.



8. Построить кривую циссоиду по заданному параметрическому уравнению $x = a t^2 / (1 + t^2)$, $y = a t^3 / (1 + t^2)$, $a > 0$, $t \in (-\infty, \infty)$.



9. Построить спираль вокруг начала координат с n витками и внешним радиусом r ; начальное направление спирали образует с осью x угол α . Параметрическое представление спирали: $x = r \cos t$, $y = r \sin t$, $r = t/2$, $\alpha \leq t \leq 2\pi n$.



10. Составить программу для управления размерами окружности и ее положением на экране. Исходная окружность имеет центр в точке (100,100) и радиус $r=20$.

Управление выполняется клавишами:

">" - увеличивает окружность (радиус увеличивается на 5 точек);

"<" - уменьшает радиус окружности на 5 точек.

Клавиши управления курсора вызывают перемещение окружности в соответствующем направлении. ESC- завершает работу.

11. Даны целые числа t_1, t_2, \dots, t_{31} - задающие график температур за март месяц.

Построить график температур. Отрезки прямых, лежащие выше 0 градусов

Цельсия и лежащие ниже 0 градусов Цельсия, должны быть окрашены в разные цвета.

12. Написать программу, которая вычерчивает на экране ломанную линию, состоящую из 200 звеньев, окрашенных в разные цвета, выбираемые случайным образом, причем координаты звеньев тоже выбираются случайно.

13. Написать программу, которая вычерчивает на экране узор из 100 окружностей случайного диаметра и цвета.

14. Написать программу, которая выводит на экран изображение работающего светофора.

15. Написать программу, которая выводит круговую диаграмму, отражающую товарооборот (в процентах) книжного магазина. Исходные данные вводятся с клавиатуры (объем продаж в рублях по категориям: книги, журналы, открытки, канцтовары).

ПРИЛОЖЕНИЕ А (форма титульного листа к лабораторной работе)

Федеральное агентство по образованию
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем
в управлении и проектировании (КСУП)

ОТЧЕТ

К лабораторной работе 1 (простые типы данных) по дисциплине
"Программирование на языках высокого уровня"

Студент гр. 587-1

С. С. Иванов

12.09.2007

Принял:

ст.преподаватель каф.КСУП

Е.А.Потапова

2007

ПРИЛОЖЕНИЕ Б (пример оформления содержания)

СОДЕРЖАНИЕ

1. Введение	3
2. Анализ задачи №21	4
3. Решение задачи	5
3.1. Выбор алгоритма и структур данных	6
3.2. Описание алгоритма	6
3.3. Тестирование программы	7
4. Заключение	8
Приложение 1. Листинг программы	9
Приложение 2. Распечатки тестов	11