**Касимов Ильнур Рафитович**, МАОУ СОШ №1, г.Агидель.

Язык программирования: **Pascal**

**Задание 1**

Разработать программу для проверки истинности высказывания:«Существует треугольник со сторонами a, b, c ?». Значения a, b, c вводятся с клавиатуры в цикле с проверкой истинности высказывания. Выход из цикла выполняется при вводе нулевого или отрицательного значения стороны треугольника.

Пример ввода:

50, 10, 10

Пример вывода:

Не существует (или Нет).

**program** One;

**var** a,b,c,i:integer;

k:**array** [1..100000000] **of** integer;

**begin**

writeln('Введите значения сторон треугольника');

**for** i:=1 **to** 3 **do begin**

**if** (i=1) **then begin** read(k[i]);

**if** (k[i]<=0) **then begin**

writeln('Значение должно быть больше 0');

**break**;

**end**;

**end**; // Значения a, b, c вводятся с клавиатуры в цикле с проверкой истинности

**if** (i=2) **then begin** read(k[i]); // высказывания. Выход из цикла выполняется при вводе нулевого или отрицательного

**if** (k[i]<=0) **then begin** // значения стороны треугольника.

writeln('Значение должно быть больше 0');

**break**;

**end**;

**end**;

**if** (i=3) **then begin** read(k[i]);

**if** (k[i]<=0) **then begin**

writeln('Значение должно быть больше 0');

**break**;

**end**;

a:=k[1]; // Присвоение переменным a,b,c значения элементов массива

b:=k[2];

c:=k[3];

**if** (i=3) **and** (a+b>c) **and** (a+c>b) **and** (b+c>a) **then** // Условия истинности высказывания

writeln ('Существует')

**else** writeln ('Не существует');

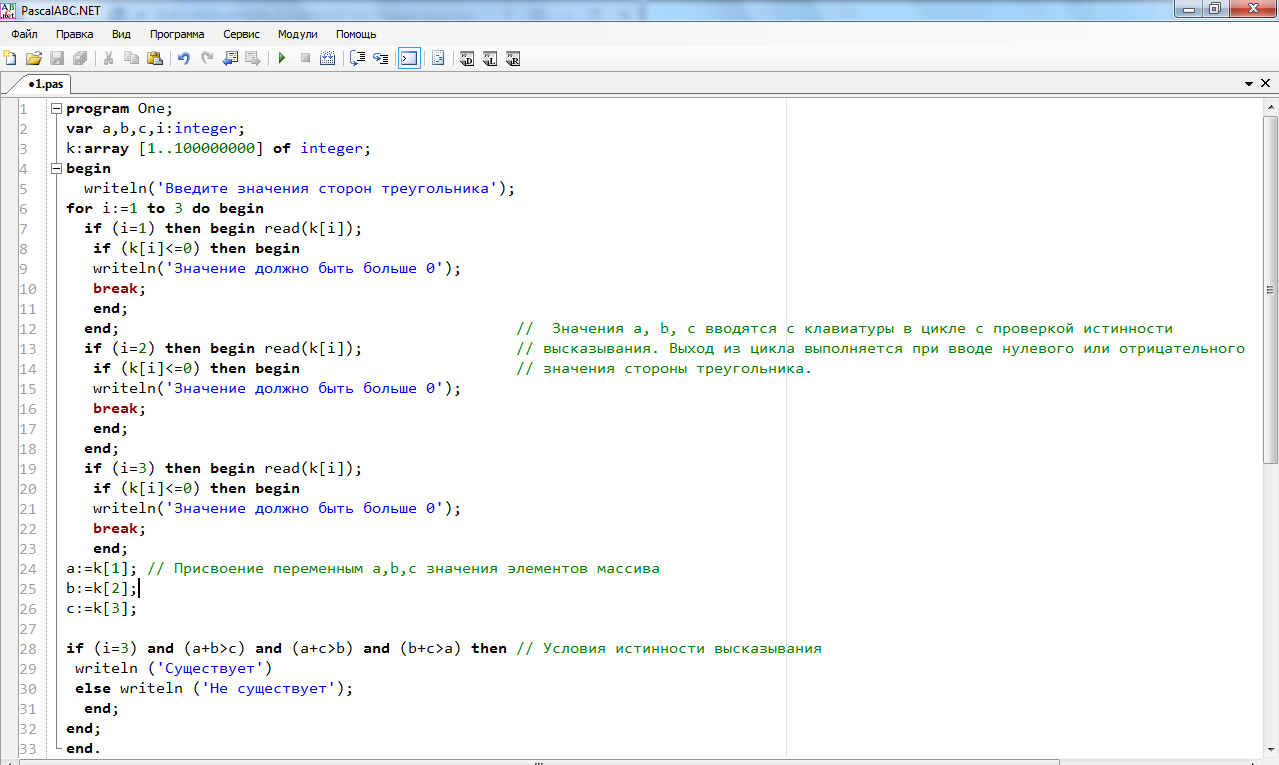
**end**;

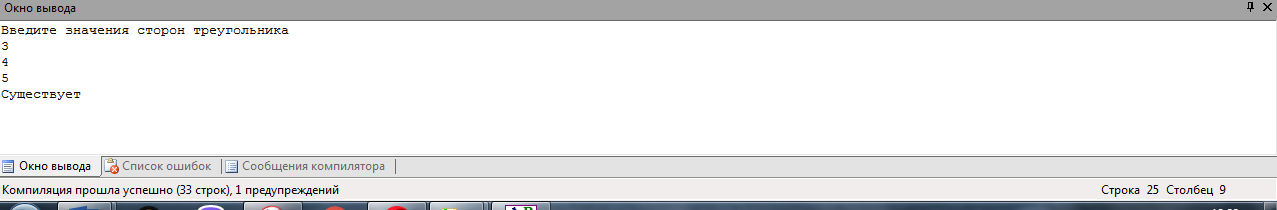
**end**;

**end**.

Входные данные: 3 4 5

Выходные данные: Существует





**Задание 2**

Разработать программу, которая выполняет классификации треугольников по величине углов и длине сторон, по вводимым с клавиатуры значениям (количество входных данных для решения должно быть достаточным).

Пример ввода:

Сторона a=20

Сторона b=10

Угол ABC (градусов)=90

Пример вывода:

Прямоугольный, разносторонний треугольник

**program** Two;

**var** a,b,c,ya,yb,yc:real;

i,j:integer;

k:**array** [1..1000000] **of** real;

**begin**

writeln('Введите значения сторон треугольника (a,b,c):');

**for** i:=1 **to** 3 **do begin**

**if** (i=1) **then begin** write ('Сторона a= '); read(k[i]);

**if** (k[i]<=0) **then begin**

writeln('Значение должно быть больше 0');

**exit**;

**end**; // Значения a, b, c вводятся с клавиатуры в цикле с проверкой истинности условия. Выход из

**end**; // цикла выполняется при вводе нулевого или отрицательного значения стороны треугольника.

**if** (i=2) **then begin** write ('Сторона b= '); read(k[i]);

**if** (k[i]<=0) **then begin**

writeln('Значение должно быть больше 0');

**exit**;

**end**;

**end**;

**if** (i=3) **then begin** write ('Сторона c= '); read(k[i]);

**if** (k[i]<=0) **then begin**

writeln('Значение должно быть больше 0');

**exit**;

**end**;

a:=k[1]; // Присвоение переменным a,b,c значения элементов массива

b:=k[2];

c:=k[3];

**if** (i=3) **and** (a+b>c) **and** (a+c>b) **and** (b+c>a) **then** // Условия существования треугольника

writeln ('Значения сторон введены ВЕРНО')

**else exit**;

**end**;

**end**;

writeln('Введите значения углов треугольника (<CAB,<ABC,<BCA):');

**for** i:=1 **to** 3 **do begin**

**if** (i=1) **then begin** write ('Угол CAB (градусов)= '); read(k[i]);

**if** (k[i]<=0) **then begin**

writeln('Значение должно быть больше 0');

**exit**;

**end**;

**end**;

**if** (i=2) **then begin** write ('Угол ABC (градусов)= '); read(k[i]);

**if** (k[i]<=0) **then begin** // Значения a, b, c вводятся с клавиатуры в цикле с проверкой истинности условия. Выход из

writeln('Значение должно быть больше 0'); // цикла выполняется при вводе нулевого или отрицательного значения стороны треугольника.

**exit**;

**end**;

**end**;

**if** (i=3) **then begin** write ('Угол BCA (градусов)= '); read(k[i]);

**if** (k[i]<=0) **then begin**

writeln('Значение должно быть больше 0');

**exit**;

**end**;

ya:=k[1];

yb:=k[2]; // Присвоение переменным ya,yb,yc значения элементов массива

yc:=k[3];

**if** (i=3) **and** (ya+yb+yc=180) **then** // Условия существования треугольника

writeln ('Значения углов введены ВЕРНО')

**else exit**;

**end**;

**end**;

**if** (ya<90) **and** (yb<90) **and** (yc<90) **and** (a<>b) **and** (a<>c) **and** (b<>c) **then** writeln ('ОСТРОУГОЛЬНЫЙ, РАЗНОСТОРОННИЙ ТРЕУГОЛЬНИК');

**if** (ya<90) **and** (yb<90) **and** (yc<90) **and** (ya=yb) **and** (b=c) **and** (ya<>yc) **then** writeln ('ОСТРОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

**if** (ya<90) **and** (yb<90) **and** (yc<90) **and** (yb=yc) **and** (a=c) **and** (ya<>yc) **then** writeln ('ОСТРОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

**if** (ya<90) **and** (yb<90) **and** (yc<90) **and** (ya=yc) **and** (a=b) **and** (ya<>yb) **then** writeln ('ОСТРОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

**if** (ya=60) **and** (yb=60) **and** (yc=60) **and** (a=b) **and** (a=c) **and** (b=c) **then** writeln ('ОСТРОУГОЛЬНЫЙ, РАВНОСТОРОННИЙ ТРЕУГОЛЬНИК');

**if** (ya=90) **and** (yb<90) **and** (yc<90) **and** (a<>b) **and** (a<>c) **and** (b<>c) **then** writeln ('ПРЯМОУГОЛЬНЫЙ, РАЗНОСТОРОННИЙ ТРЕУГОЛЬНИК');

**if** (ya<90) **and** (yb=90) **and** (yc<90) **and** (a<>b) **and** (a<>c) **and** (b<>c) **then** writeln ('ПРЯМОУГОЛЬНЫЙ, РАЗНОСТОРОННИЙ ТРЕУГОЛЬНИК'); // выполнение классификации треугольников

**if** (ya<90) **and** (yb<90) **and** (yc=90) **and** (a<>b) **and** (a<>c) **and** (b<>c) **then** writeln ('ПРЯМОУГОЛЬНЫЙ, РАЗНОСТОРОННИЙ ТРЕУГОЛЬНИК');

**if** (ya<90) **and** (yb<90) **and** (yc=90) **and** (ya=yb) **and** (a<>b) **and** (a<>c) **and** (b=c) **and** (b\*b+c\*c=a\*a) **then** writeln ('ПРЯМОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

**if** (ya=90) **and** (yb<90) **and** (yc<90) **and** (yb=yc) **and** (a<>b) **and** (a=c) **and** (b<>c) **and** (a\*a+c\*c=b\*b) **then** writeln ('ПРЯМОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

**if** (ya<90) **and** (yb=90) **and** (yc<90) **and** (ya<>yc) **and** (a=b) **and** (a<>c) **and** (b<>c) **and** (a\*a+b\*b=c\*c) **then** writeln ('ПРЯМОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

**if** (ya>90) **and** (yb<90) **and** (yc<90) **and** (a<>b) **and** (a<>c) **and** (b<>c) **then** writeln ('ТУПОУГОЛЬНЫЙ, РАЗНОСТОРОННИЙ ТРЕУГОЛЬНИК');

**if** (ya<90) **and** (yb>90) **and** (yc<90) **and** (a<>b) **and** (a<>c) **and** (b<>c) **then** writeln ('ТУПОУГОЛЬНЫЙ, РАЗНОСТОРОННИЙ ТРЕУГОЛЬНИК');

**if** (ya<90) **and** (yb<90) **and** (yc>90) **and** (a<>b) **and** (a<>c) **and** (b<>c) **then** writeln ('ТУПОУГОЛЬНЫЙ, РАЗНОСТОРОННИЙ ТРЕУГОЛЬНИК');

**if** (ya>90) **and** (yb<90) **and** (yc<90) **and** (yb=yc) **and** (a<>b) **and** (a=c) **and** (b<>c) **then** writeln ('ТУПОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

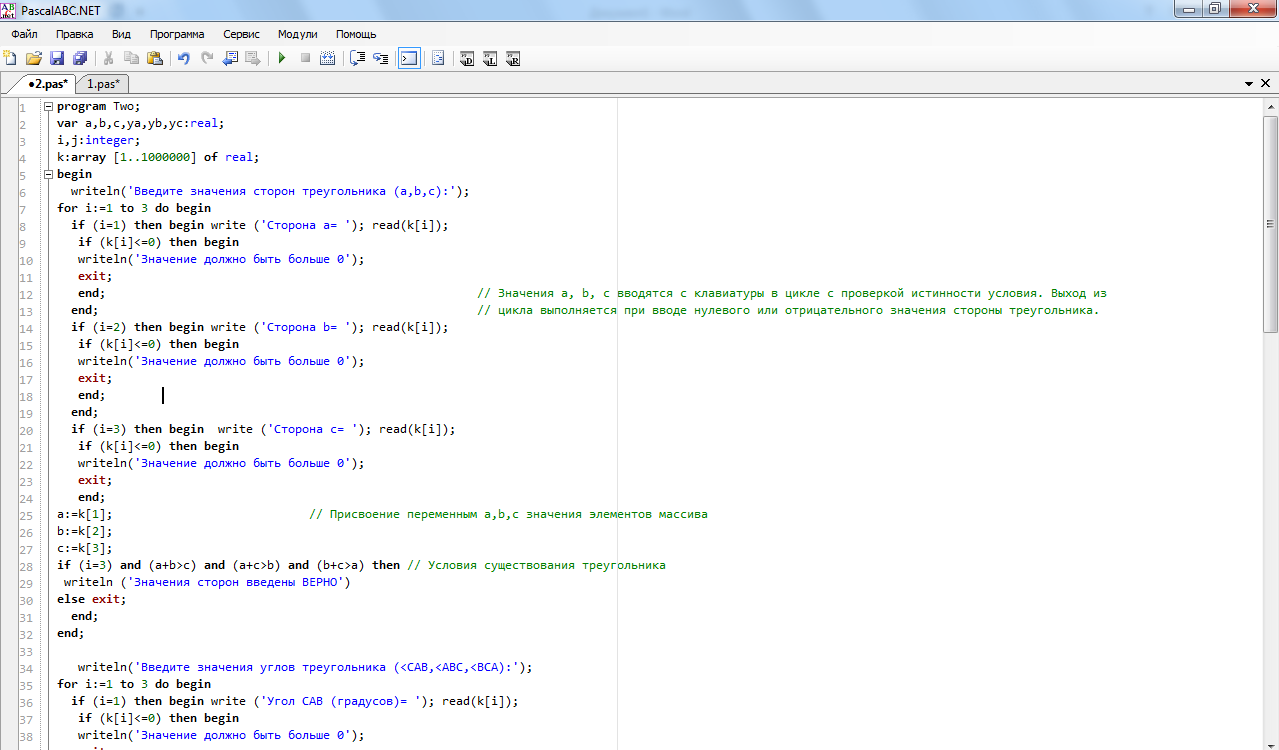
**if** (ya<90) **and** (yb>90) **and** (yc<90) **and** (ya=yc) **and** (a=b) **and** (a<>c) **and** (b<>c) **then** writeln ('ТУПОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

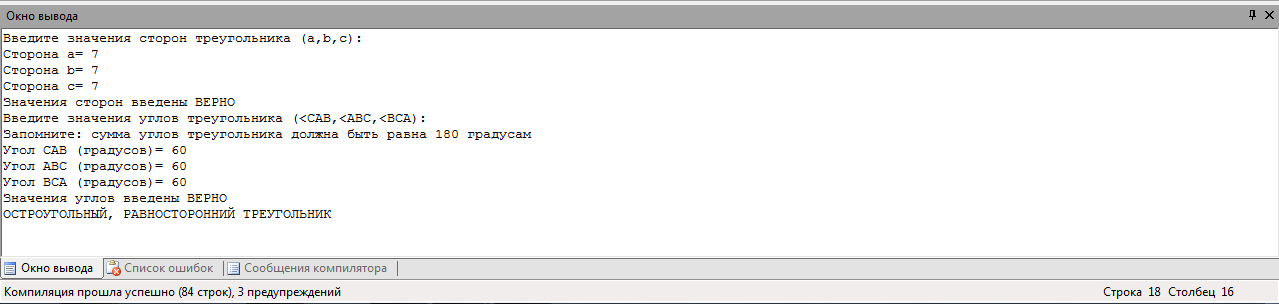
**if** (ya<90) **and** (yb<90) **and** (yc>90) **and** (ya=yb) **and** (a<>b) **and** (a<>c) **and** (b=c) **then** writeln ('ТУПОУГОЛЬНЫЙ, РАВНОБЕДРЕННЫЙ ТРЕУГОЛЬНИК');

**end**.

Входные данные: 7 7 7 60 60 60

Выходные данные: ОСТРОУГОЛЬНЫЙ, РАВНОСТОРОННИЙ ТРЕУГОЛЬНИК





**Задание 3**

Разработать программу, которая выводит на экран все простые числа в заданном диапазоне от N1 до N2 и подсчитывает их количество.

Пример ввода:

N1= 10

N2= 15

Пример вывода:

11 13

Количество простых чисел: 2

**program** three ;

**var** k,j,N1,N2: integer;

**function** Prost(a:longint):boolean;

**var** i:longint;

**begin**

**if** a<2 **then** Prost:=false

**else**

**begin** // Функция проверяющая принадлежность числа к простым числам

Prost:=true;

**for** i:=2 **to** round(sqrt(a)) **do**

**if** a **mod** i=0 **then**

**begin**

Prost:=false;

**break**;

**end**;

**end**;

**end**;

**var** i:integer;

**begin**

writeln ('Введите диапазон:');

write ('N1= '); read (N1); // Ввод данных с клавиатуры о диапазоне

write ('N2= '); read (N2);

**for** i:=N1 **to** N2 **do begin**

**if** Prost(i)**then begin** // Выполнение функции с выводом значений простых чисел

write(i,' ');

k:=k+1; // Подсчет кол-ва простых чисел в заданном диапазоне

**end**;

**end**;

writeln; // Вывод кол-ва простых чисел в заданном диапазоне

writeln('Кол-во простых чисел: ',k);

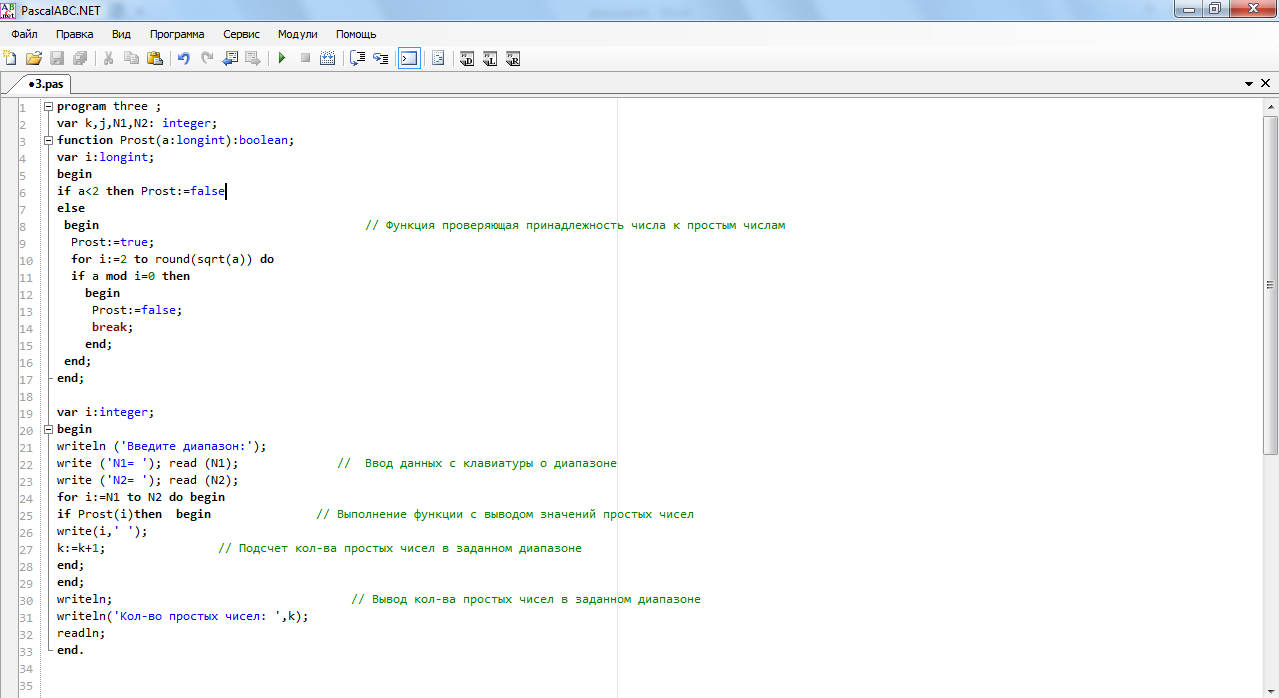
readln;

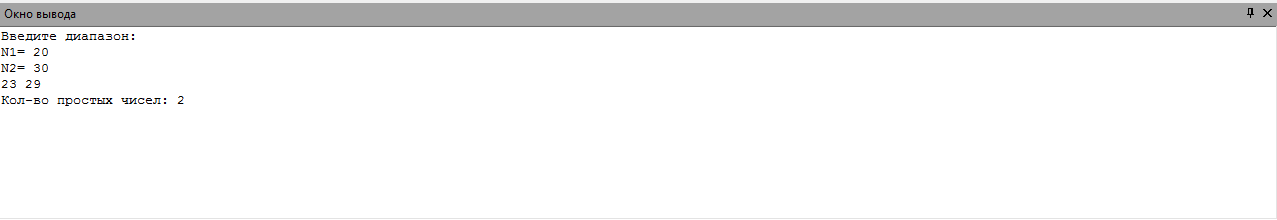
**end**.

Входные данные: 20 30

Выходные данные: 23 29

Кол-во простых чисел: 2





**Задание 4**

Дан размер файла в Мегабайтах. Разработать программу для подсчета количества байт, которые занимает данный файл.

Пример ввода:

128

Пример вывода:

131072

**program** four;

**var** mb,b:integer;

**begin**

write('Введите размер файла (в Мегабайтах): '); // Ввод данных о кол-ве Мегабайт

read(mb);

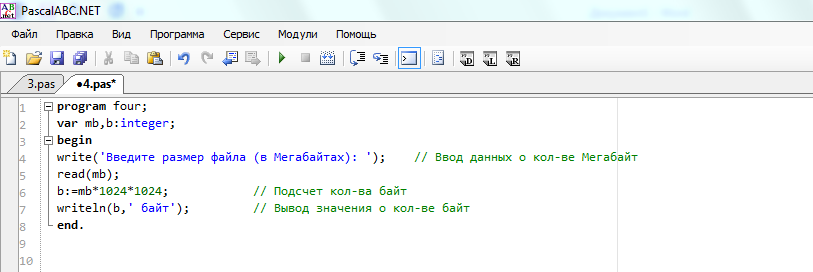
b:=mb\*1024\*1024; // Подсчет кол-ва байт

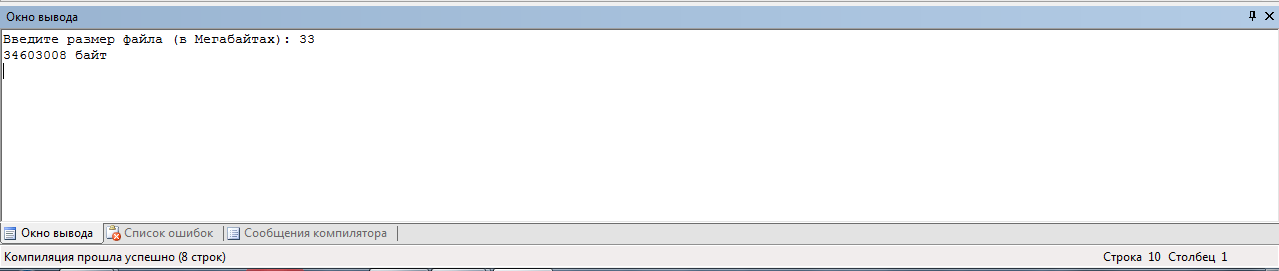
writeln(b,' байт'); // Вывод значения о кол-ве байт

**end**.

Входные данные: 33

Выходные данные: 34603008 байт





**Задание 5**

Разработать программу для вычисления расстояния между двумя точками с заданными координатами (x1, y1) и (x2, y2) на плоскости. Расстояние вычисляется по формуле sqrt((x2 - x1)^2+ (y2 - y1)^2).

Пример ввода:

Координаты точки А x1=9 y1=19

Координаты точки B x2=12 y1=23

Пример вывода:

Расстояние S=5

**program** five;

**var** x1,y1,x2,y2,r:real;

**begin**

writeln('Координаты точки А: ');

write(' x1='); read(x1);

write(' y1='); read(y1);

// Ввод данных о координатах точек

writeln('Координаты точки B: ');

write(' x2='); read(x2);

write(' y2='); read(y2);

r:=sqrt((x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1)); // Подсчет расстояния между точками

writeln('Расстояние S=',r); // Вывод значения расстояния

**end**.

Входные данные: 25 25 30 30

Выходные данные: Расстояние S=7.07106781186548

