**ЗАДАНИЯ**

**2 тура дистанционной олимпиады по программированию**

**для 10-11 классов**

Галимарданов Альберт Асхатович (MrTrollolo1337@gmail.com)

Адрес: Россия, Республика Башкортостан, Агидель, Цветочный бульвар, 4

МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ "СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №1" ГОРОДСКОГО ОКРУГА ГОРОД АГИДЕЛЬ РЕСПУБЛИКИ БАШКОРТОСТАН

Решение произведено в программе PascalABC.NET версии 3.3.

**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.

**program** qq;

**var** i,n:integer;

a:**array**[1..3]**of** real;

**begin**

**while** n=0 **do begin**//зацикливаем программу, пока не введется 0 или отрицательное число

**for** i:=1 **to** 3 **do begin**//цикл для считывания сторон треугольника в массив

read(a[i]);

**if** a[i]<1 **then begin** inc(n);**break**;//условие задачи"Выход из цикла выполняется при вводе нулевого или отрицательного значения стороны треугольника"

writeln('Введено нулевое или отрицательное значение стороны треугольника');

**end**;**end**;

**if** (a[1]<a[2]+a[3])**and**(a[2]<a[1]+a[3])**and**(a[3]<a[2]+a[1]) **then** //проверка истиности существования треугольника

writeln('Существует') **else** writeln('Не существует');//вывод результата

a[1]:=0;a[2]:=0;a[3]:=0;//обнуление массива для дальнейших треугольников

**end**;

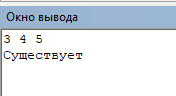
**end**.

Входные данные:

3 4 5

Выходные данные:

Существует



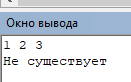
ИЛИ

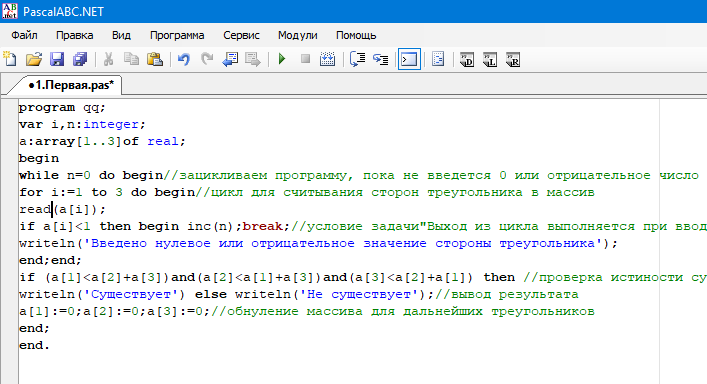
Входные данные:

1 2 3

Выходные данные:

Не существует





**Задание 2**

Разработать программу, которая выполняет классификации треугольников по величине углов и длине сторон, по вводимым с клавиатуры значениям.

//Программа работает для треугольника с углами ABC, BAC, BCA.

//А также со сторанами а против угла bАC, b против угла ABC и c против угла BCA.

**program** qq;

**var** i,k,p,x,pu,ps:integer;

s,dannoe:string;

l,l1,l2,k1,k2,c1,c2,c3,a,b,c,abc,bca,bac:real;

**begin**

writeln('Для работы программы введите стороны или углы треугольника');

writeln('Программа работает для трёх введённых данных');

writeln('Формат: Сторона a=20');

writeln(' Сторона b=10');

writeln(' Угол ABC=90');

**for** i:=1 **to** 3 **do begin**//из курса геометрии известно,что достаточно трёх данных, чтобы опознать вид треугольника

k:=0;

s:='';

readln(dannoe);//считывается в виде строки

k:=pos('торона',dannoe);// ищу фрагмент "торона", так как пользователь может написать с заглавной буквы "с"

**if** k>0 **then begin**

k:=0;

//'работа со сторонами'

k:=pos('a',dannoe);//опознание стороны а

**if** k>0 **then begin** k:=pos('=',dannoe);

**for** x:=k+1 **to** length(dannoe) **do**

s:=s+dannoe[x];

k:=0;

val(s,a,p);

**end**;

k:=pos('b',dannoe);//опознание стороны b

**if** k>0 **then begin** k:=pos('=',dannoe);

**for** x:=k+1 **to** length(dannoe) **do**

s:=s+dannoe[x];

k:=0;

val(s,b,p);

**end**;

k:=pos('c',dannoe);//опознание стороны c

**if** k>0 **then begin** k:=pos('=',dannoe);

**for** x:=k+1 **to** length(dannoe) **do**

s:=s+dannoe[x];

k:=0;

val(s,c,p);

**end**;

**end else begin**

k:=0;

//'работа с углами'

k:=pos('ABC',dannoe);//опознание угла ABC

**if** k>0 **then begin** k:=pos('=',dannoe);

**for** x:=k+1 **to** length(dannoe) **do**

s:=s+dannoe[x];

k:=0;

val(s,abc,p);

**end**;

k:=pos('BAC',dannoe);//опознание угла BAC

**if** k>0 **then begin** k:=pos('=',dannoe);

**for** x:=k+1 **to** length(dannoe) **do**

s:=s+dannoe[x];

k:=0;

val(s,bac,p);

**end**;

k:=pos('BCA',dannoe);//опознание угла BCA

**if** k>0 **then begin** k:=pos('=',dannoe);

**for** x:=k+1 **to** length(dannoe) **do**

s:=s+dannoe[x];

k:=0;

val(s,bca,p);

**end**;

**end**;

**end**;

//решение задачи

**if** (abc>0)**and**(bca>0)**and**(bac=0) **then** bac:=180-abc-bca;//нахождение третьего угла, по данным двум

**if** (abc>0)**and**(bac>0)**and**(bca=0) **then** bca:=180-abc-bac;//нахождение третьего угла, по данным двум

**if** (bac>0)**and**(bca>0)**and**(abc=0) **then** abc:=180-bca-bac;//нахождение третьего угла, по данным двум

**if**(a>0)**and**(bac>0)**and**(abc>0)**and**(b=0)**then** b:=(a\*sin(abc))/sin(bac);//нахождение стороны по теореме синусов

**if**(a>0)**and**(bac>0)**and**(bca>0)**and**(c=0)**then** c:=(a\*sin(bca))/sin(bac);//нахождение стороны по теореме синусов

**if**(b>0)**and**(abc>0)**and**(bca>0)**and**(c=0)**then** c:=(b\*sin(bca))/sin(abc);//нахождение стороны по теореме синусов

**if**(b>0)**and**(abc>0)**and**(bac>0)**and**(a=0)**then** a:=(b\*sin(bac))/sin(abc);//нахождение стороны по теореме синусов

**if**(c>0)**and**(bca>0)**and**(bac>0)**and**(a=0)**then** a:=(c\*sin(bac))/sin(bca);//нахождение стороны по теореме синусов

**if**(c>0)**and**(bca>0)**and**(abc>0)**and**(b=0)**then** b:=(c\*sin(abc))/sin(bca);//нахождение стороны по теореме синусов

**if**(a>0)**and**(b>0)**and**(bac>0)**and**(abc=0)**then** abc:=arcsin((sin(bac)\*b)/a);//нахождение угла по теореме синусов

**if**(a>0)**and**(c>0)**and**(bac>0)**and**(bca=0)**then** bca:=arcsin((sin(bac)\*c)/a);//нахождение угла по теореме синусов

**if**(b>0)**and**(c>0)**and**(abc>0)**and**(bca=0)**then** bca:=arcsin((sin(abc)\*c)/b);//нахождение угла по теореме синусов

**if**(b>0)**and**(a>0)**and**(abc>0)**and**(bac=0)**then** bac:=arcsin((sin(abc)\*a)/b);//нахождение угла по теореме синусов

**if**(c>0)**and**(a>0)**and**(bca>0)**and**(bac=0)**then** bac:=arcsin((sin(bca)\*a)/c);//нахождение угла по теореме синусов

**if**(c>0)**and**(b>0)**and**(bca>0)**and**(abc=0)**then** abc:=arcsin((sin(bca)\*b)/c);//нахождение угла по теореме синусов

**if**(abc>0)**and**(bca>0)**and**(bac>0)**and**((abc+bca+bac)<>180) **then begin** writeln('Треугольник не существует. Повторите попытку.');**exit**;**end**;

**if** (a>0)**and**(b>0)**and**(c>0)**and**(a<b+c)**and**(b<a+c)**and**(c<b+a)**then begin** writeln('Треугольник не существует. Повторите попытку.');**exit**;**end**;

**if** (pu<1) **then**

**if** (a=b)**and**(a=c)**and**(b=c)**and**(a>0)**and**(b>0)**and**(c>0)**then begin** inc(pu);write('Остроугольный, ')**end**;//известны 3 стороны

**if** (pu<1) **then**

**if** (abc=90)**or**(bca=90)**or**(bac=90) **then begin** inc(pu);write('Прямоугольный, ');**end**;//известен угол

**if** (pu<1) **then**

**if** (abc>90)**or**(bca>90)**or**(bac>90) **then begin** inc(pu);write('Тупоугольный, ');**end**;//известен угол

**if** (pu<1) **then**

**if** (abc<90)**and**(bca<90)**and**(bac<90)**then begin** inc(pu);write('Остроугольный, ');**end**;//известен угол

**if** (pu<1) **then**

**if** (a>0)**and**(b>0)**and**(c>0) **then begin**

c1:=(sqr(a)-sqr(b)-sqr(c))/((-2)\*b\*c);

c2:=(sqr(c)-sqr(a)-sqr(b))/((-2)\*a\*b);

c3:=(sqr(b)-sqr(c)-sqr(a))/((-2)\*c\*a);

**if** (c1=1)**or**(c2=1)**or**(c3=1)**then begin** inc(pu);write('Прямоугольный, ')**end**;//проверка на прямоугольность треугольника по теореме косинусов

**end**;

**if** (pu<1) **then**

**if**(a>0)**and**(b>0)**and**(c>0) **and**((a\*a=b\*b+c\*c) **or** (b\*b=a\*a+c\*c) **or** (c\*c=a\*a+b\*b)) **then begin** inc(pu);write('Прямоугольный, ');**end**;//проверка прямоугольности по теореме Пифагора

**if** (pu<1) **then**

**if**(abc>0)**and**(b>0)**then begin** l:=b/sin(abc);**if** (l=a)**or**(l=c)**then begin** inc(pu);write('Прямоугольный, ') ;**end**;**end**;//известен угол и противолежащая сторона

**if** (pu<1) **then**

**if**(bac>0)**and**(a>0)**then begin** l:=a/sin(bac);**if** (l=c)**or**(l=b)**then begin** inc(pu);write('Прямоугольный, ') ;**end**;**end**;//известен угол и противолежащая сторона

**if** (pu<1) **then**

**if**(bca>0)**and**(c>0)**then begin** l:=c/sin(bca);**if** (l=a)**or**(l=b)**then begin** inc(pu);write('Прямоугольный, ') ;**end**;**end**;//известен угол и противолежащая сторона

**if** (pu<1) **then**

**if** (a>0)**and**(b>0)**and**(c>0)**and**(b\*b+c\*c-a\*a>0)**and**(a\*a+b\*b-c\*c>0)**and**(a\*a+c\*c-b\*b>0) **then begin** inc(pu);write('Остроугольный, ')**end**;//проверка остроугольности по теореме Пифагора

**if** (pu<1) **then**

**if** (a>0)**and**(b>0)**and**(c>0)**and**((b\*b+c\*c-a\*a<0)**and**(a\*a+b\*b-c\*c>0)**and**(a\*a+c\*c-b\*b>0))**or**

((b\*b+c\*c-a\*a>0)**and**(a\*a+b\*b-c\*c<0)**and**(a\*a+c\*c-b\*b>0))**or**((b\*b+c\*c-a\*a>0)**and**(a\*a+b\*b-c\*c>0)**and**(a\*a+c\*c-b\*b<0)) **then**

**begin** inc(pu);write('Тупоугольный, ');**end**;//проверка тупоугольности по теореме Пифагора

**if** (pu<1) **then**

**if**(abc>0)**and**(c>0)**and**(a>0)**then begin** l1:=c/cos(abc);l2:=a/cos(abc);**if** (l1=a)**or**(l2=c)**then begin** inc(pu);write('Прямоугольный, ') ;**end**;**end**;//известен угол и 2 прилежащие стороны

**if** (pu<1) **then**

**if**(bca>0)**and**(a>0)**and**(b>0)**then begin** l1:=a/cos(bca);l2:=b/cos(bca);**if** (l1=b)**or**(l2=a)**then begin** inc(pu);write('Прямоугольный, ') ;**end**;**end**;//известен угол и 2 прилежащие стороны

**if** (pu<1) **then**

**if**(bac>0)**and**(c>0)**and**(b>0)**then begin** l1:=c/cos(bac);l2:=b/cos(bac);**if** (l1=b)**or**(l2=c)**then begin** inc(pu);write('Прямоугольный, ') ;**end**;**end**;//известен угол и 2 прилежащие стороны

**if** ps<1 **then**

**if**(c>0)**and**(a=c)**and**(abc<>60)**then begin** inc(ps);write('равнобедренный');**end**;//известны 2 стороны,с исключением угла 60 градусов

**if** ps<1 **then**

**if**(b>0)**and**(b=c)**and**(bac<>60)**then begin** inc(ps);write('равнобедренный');**end**;//известны 2 стороны,с исключением угла 60 градусов

**if** ps<1 **then**

**if**(a>0)**and**(a=b)**and**(bca<>60)**then begin** inc(ps);write('равнобедренный');**end**;//известны 2 стороны,с исключением угла 60 градусов

**if** ps<1 **then**

**if**(c>0)**and**(a=c)**and**(abc=60)**then begin** inc(ps);write('равносторонний');**end**;//известны 2 стороны и угол 60 градусов

**if** ps<1 **then**

**if**(b>0)**and**(b=c)**and**(bac=60)**then begin** inc(ps);write('равносторонний');**end**;//известны 2 стороны и угол 60 градусов

**if** ps<1 **then**

**if**(a>0)**and**(a=b)**and**(bca=60)**then begin** inc(ps);write('равносторонний');**end**;//известны 2 стороны и угол 60 градусов

**if** ps<1 **then**

**if**(a=b)**and**(b=c)**and**(a=c)**and**(a>0)**and**(b>0)**and**(c>0)**then begin** inc(ps);write('равносторонний');**end**;//известны 3 стороны

**if** ps<1 **then**

**if**(abc<>bca)**and**(bca<>bac)**and**(abc<>bac)**and**(abc>0)**and**(bca>0)**and**(bac>0)**then begin** inc(ps); write('разносторонний');**end**;//известны 3 угла

**if** ps<1 **then**

**if** ((abc=90)**and**((bca>0)**or**(bac>0)))**or**((bac=90)**and**((bca>0)**or**(abc>0)))**or**((bca=90)**and**((abc>0)**or**(bac>0))) **then begin** inc(ps);write('разносторонний');**end**;// известны 3 угла

**if** ps<1 **then**

**if** (a>0)**and**(b>0)**and**(c>0)**and**(a<>b)**and**(a<>c)**and**(b<>c)**then begin** inc(ps);write('разносторонний');**end**;//изветсны 3 стороны

**if** ps<1 **then**

**if** ((abc=90)**and**(b>0))**then if**(a>0)**or**(c>0)**then begin** k1:=sqrt((b\*b)-(a\*a));k2:=sqrt((b\*b)-(c\*c));**if** (k1=c)**or**(k2=a)**then begin** inc(ps); write('равнобедренный') **end else begin** inc(ps);write('разносторонний');**end**;**end**;//теорема Пифагора

**if** ps<1 **then**

**if** (abc=60) **and** (bca=60) **and** (bac=60) **then begin** inc(ps);write('равносторонний');**end**;//известны 3 угла

**if** (a=0)**and**(b=0)**and**(c=0)**and**(abc=0)**and**(bca=0)**and**(bac=0)**then begin** writeln('Неправильный формат ввода. Повторите попытку.');**exit**;**end**;//проверка на верность формата ввода

**if** ps=0 **then** write('разносторонний треугольник.');//добавление слова треугольник

**if** (pu>0)**and**(ps>0) **then** write(' треугольник.');//добавление слова треугольник

**end**.

Входные данные:

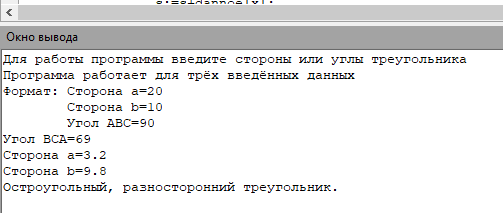
Угол BCA=69

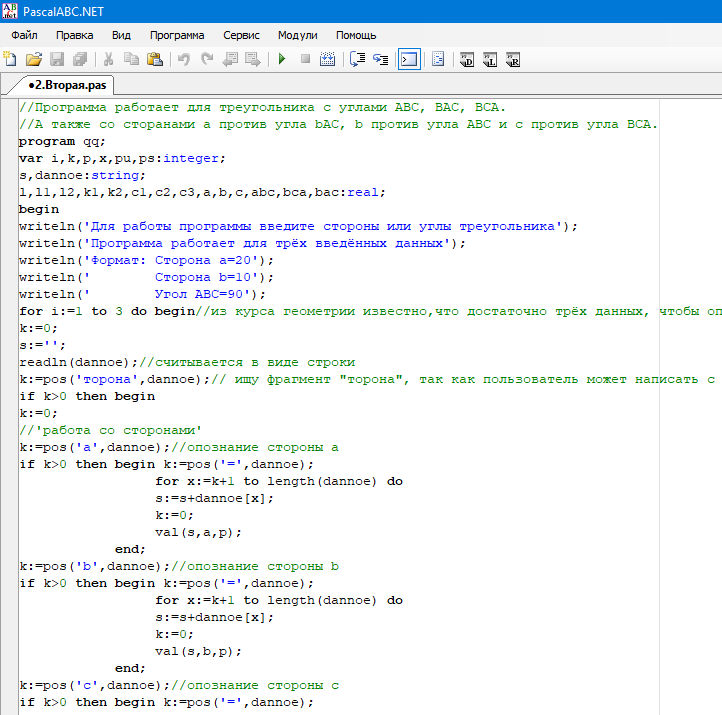
Сторона a=3.2

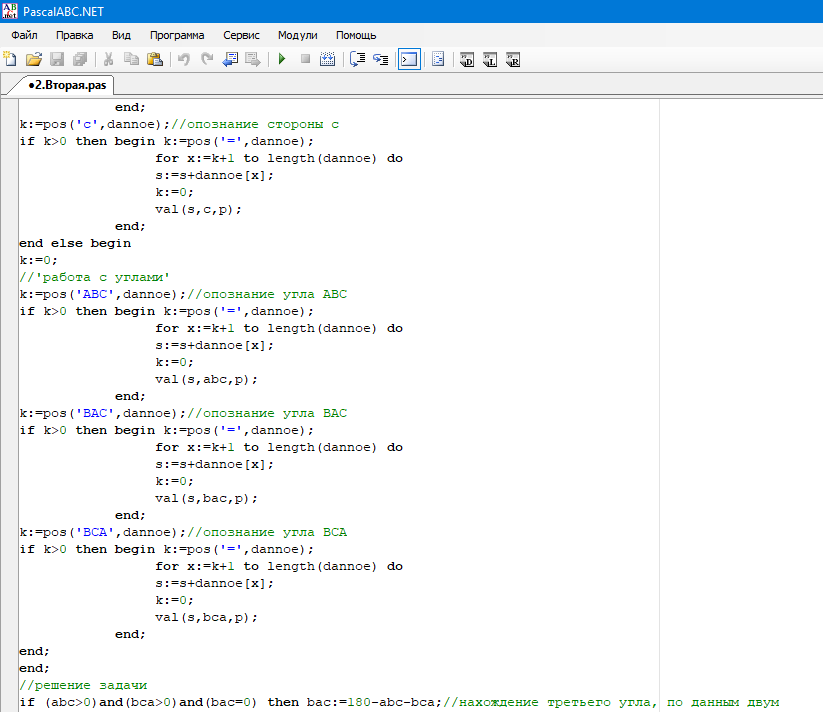
Сторона b=9.8

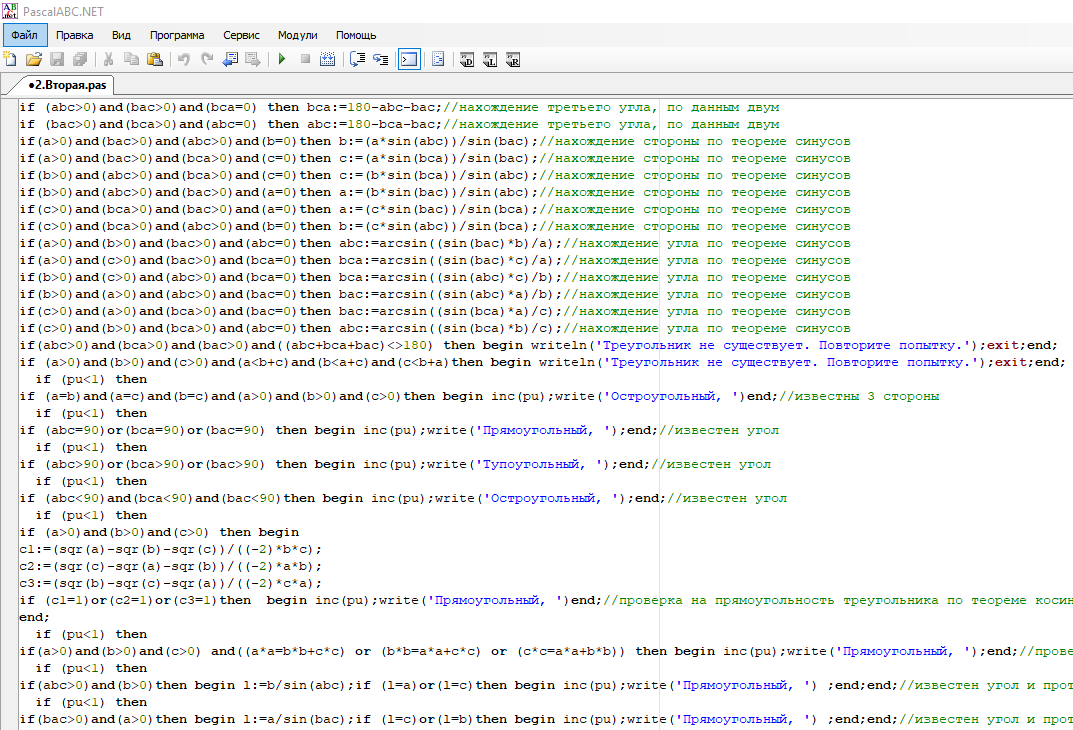
Выходные данные:

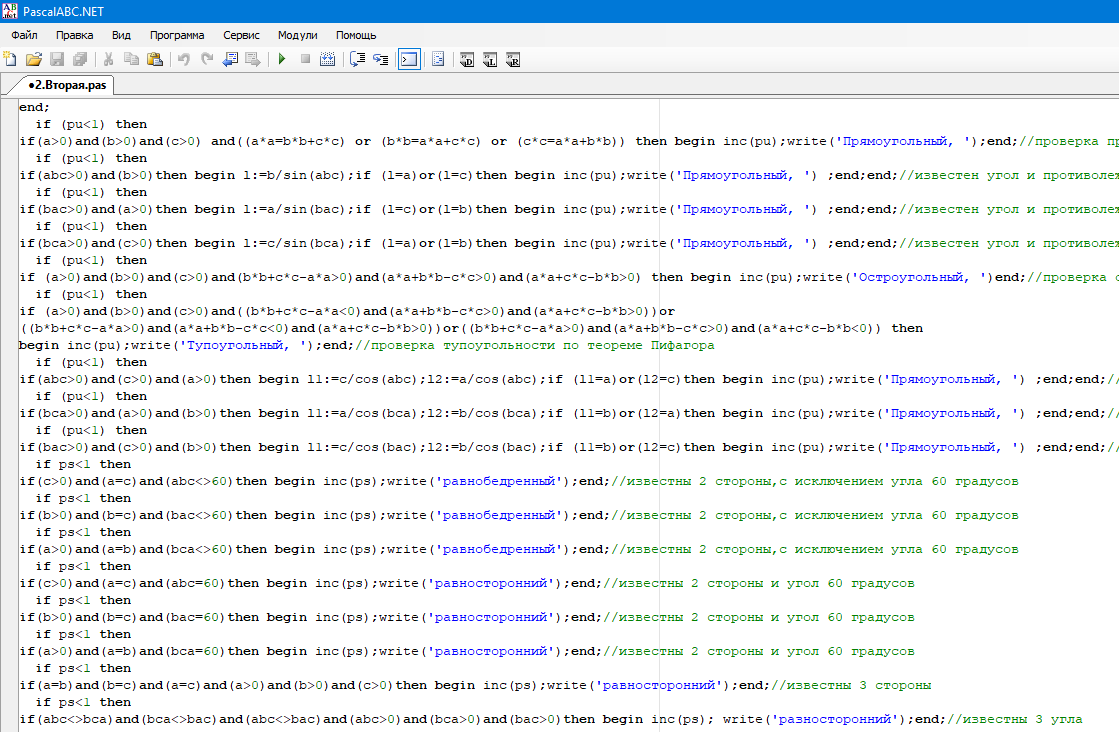
Остроугольный, разносторонний треугольник.

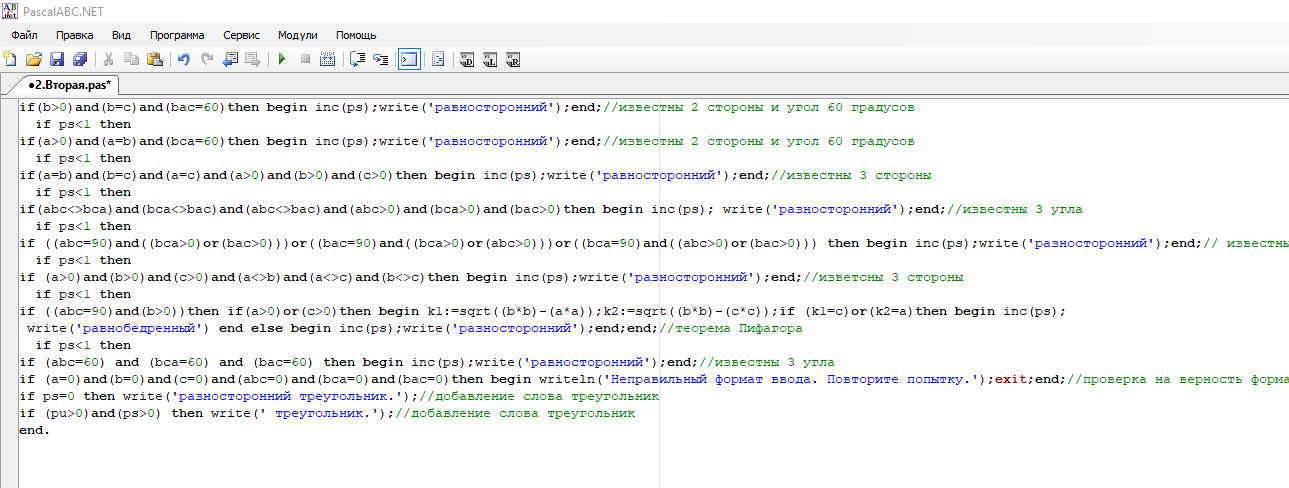












Задание 3

**program** qq;

**var** c,n1,n2,i,k,l,j,ploxo,o:integer;

**begin**

write('Введите начало диапазона N1=');//ввод начала диапазона

readln(n1);

write('Введите конец диапазона N2=');//ввод конца диапазона

readln(n2);

**if** (n2<=n1)**then begin** writeln('Введен неправильный диапазон');**exit**;**end**;//исключение некореектности диапазона

**if** (n1<3)**and**(n2>=2)**then begin** writeln('2');o:=o+1;**end**;//ввод первых двух простых чисел(число 1 нек является простым)

**if** (n1<4)**and**(n2>=3)**then begin** writeln('3');o:=o+1;**end**;

**for** i:=1 **to** n2 **do begin**

k:=6\*i-1;//формула простого числа(любое простое число больше трёх можно представить в виде 6k+-1)

l:=6\*i+1;

**if** (k>=n1)**and**(k<=n2)**then begin**//проверка на истиность нахождения К в заданном диапазоне

ploxo:=0;

**for** j:=1 **to** k **do begin**

**if** k **mod** j =0 **then** inc(ploxo);//проверка на кол-во делителей

**end**;

**if** ploxo=2 **then begin** writeln(k);c:=c+1;**end**;//вывод простых чисел и их подсчет

**end**;

**if** (l>=n1)**and**(l<=n2)**then begin**//проверка на истиность нахождения L в заданном диапазоне

ploxo:=0;

**for** j:=1 **to** l **do begin**

**if** l **mod** j =0 **then** inc(ploxo);//проверка на кол-во делителей

**end**;

**if** ploxo=2 **then begin** writeln(l);c:=c+1;**end**;//вывод простых чисел и их подсчет

**end**;

**end**;writeln;writeln;

writeln('Количество простых чисел: ',c+o);//вывод кол-ва простых чисел

**end**.

**Задание 3**

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

Входные данные:

Введите начало диапазона N1=1

Введите конец диапазона N2=5

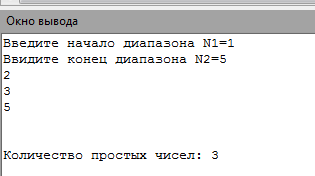
Выходные данные:

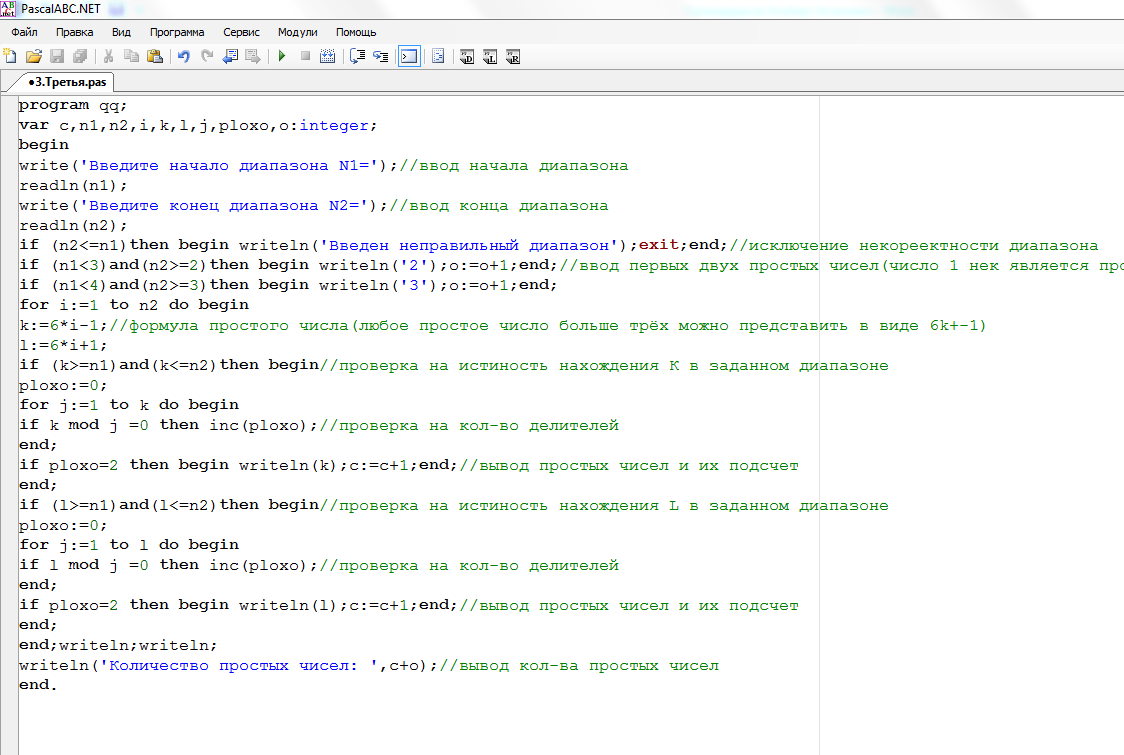
2

3

5

Количество простых чисел: 3





**Задание 4**

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

//В условии задачи сказано составить программу для перевода МагаБайт в Байты, но в примере ввода 128 МегаБайт, а в выводе 131072, якобы Байт, но это КилоБайт.

//Поэтому я составил программу для перевода Мегабайт и в Байты, и в КилоБайты.

**program** qq;

**var** n:real;

**begin**

writeln('Введите размер файла в МегаБайтах');

readln(n);//ввод размера фала

**if** n=0 **then begin** writeln('Файл не имеет размера.');**exit**;**end**;//проверка на сущность числа

writeln('Файл весит: ',n\*1024\*1024,' Байт');//одновременный подсчет и вывод эквивалента МегаБайт в Байт

writeln('Файл весит: ',n\*1024,' КилоБайт');//одновременный подсчет и вывод эквивалента МегаБайт в КилоБайт

**end**.

Входные данные:

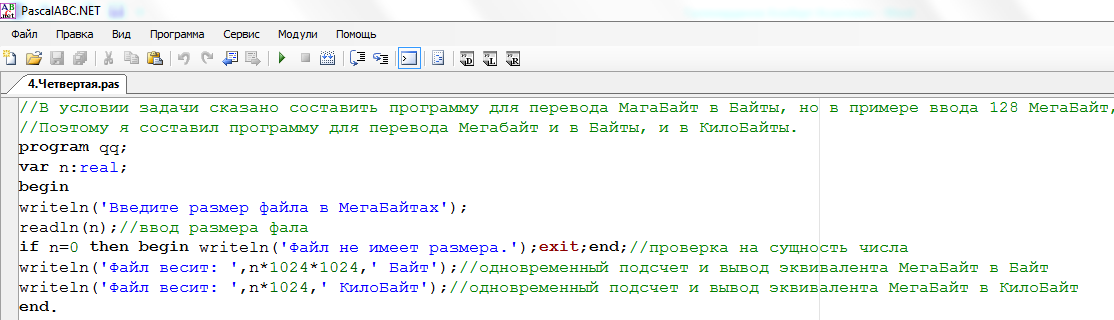
Введите размер файла в МегаБайтах

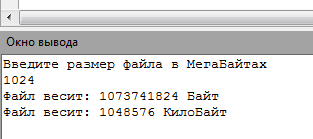
1024

Выходные данные:

Файл весит: 1073741824 Байт

Файл весит: 1048576 КилоБайт

****

****

**Задание 5**

Разработать программу для вычисления расстояния между двумя точками с заданными координатами (x1, y1) и (x2, y2) на плоскости.

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

**begin**

writeln('Координаты точки A');//ввод координат точки А

write('x1=');

readln(x1);//ввод координаты х1

write('y1=');

readln(y1);//ввод координаты у1

writeln('Координаты точки B');//ввод координат точки В

write('x2=');

readln(x2);//ввод координаты х2

write('y2=');

readln(y2);//ввод координаты у2

D:=sqrt(sqr(x2 - x1)+ sqr(y2 - y1));//решение по данной формуле

writeln('Расстояние S=',d);//вывод результата

**end**.

Входные данные:

Координаты точки A

x1=0

y1=0

Координаты точки B

x2=3

y2=4

Выходные данные:

Расстояние S=5

