**Мухаметов Максим Вячеславович**

**МАОУ СОШ №1 г.Агидель**

[**p.puschckin2017@yandex.ru**](mailto:p.puschckin2017@yandex.ru)

PascalABC.NET Версия 3.3. сборка 1611(16.01.2018)

**Задание 1**

Разработать программу, которая заполняет и выводит на экран таблицу истинности функции **(A \/ B) /\¬C**. Таблица состоит из значений логических переменных A, B, C только для тех строк, в которых значение функции равно введенному с клавиатуры значению **F**.

**Пример ввода:**

Значение функции F = 1

**Пример вывода:**

Значения переменных A, B, C: 0 1 0

Значения переменных A, B, C: 1 0 0

Значения переменных A, B, C: 1 1 0

program qq;

var a,b,c,f,k:byte;

begin

writeln('введите F');

readln(f);

while ((f<>0) and (f<>1)) do begin

writeln('Неверный формат,введите 1 или 0');

readln(f);

end;

for a:=0 to 1 do

for b:=0 to 1 do

for c:=0 to 1 do begin

if (a+b<>0) then // если сумма a+b не ноль,то она равна 1

k:=1

else k:=0;

if (k\*abs(c-1)=f) then begin //abs(c-1) - не с, если это произведение

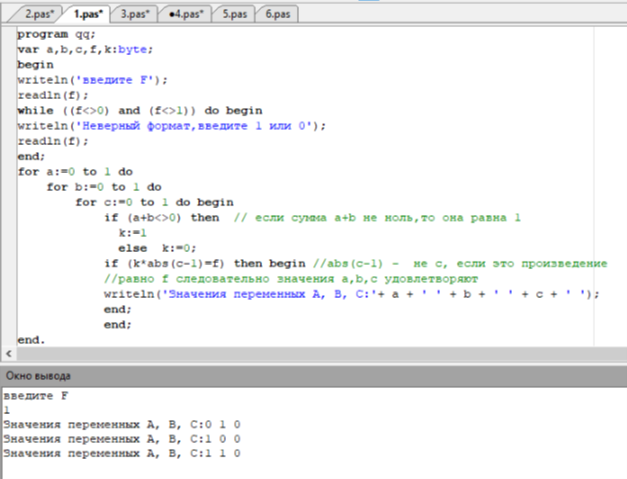
//равно f следовательно значения a,b,c удовлетворяют

writeln('Значения переменных A, B, C:'+ a + ' ' + b + ' ' + c + ' ');

end;

end;

end.



**Задание 2**

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

**Пример ввода:**

Количество известных углов: 4

Количество известных сторон: 0

Угол 1: 90

Угол 2: 90

Угол 3: 90

Угол 4: 90

**Пример вывода:**

Прямоугольник

program qq;

var k,l,g,w,e,j,i:integer;

a: array [1..4] of integer;

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

a1: array [1..4] of integer;

b1: array [1..4] of integer;

begin

writeln('Введите количество известных углов');

readln(k);

while ((k>4) or (k<0)) do begin

writeln('Введите количество известных углов (от 0 до 4)');

readln(k);

end;

writeln('Введите количество известных сторон');

readln(l);

while ((l>4) or (l<0)) do begin

writeln('Введите количество известных сторон (от 0 до 4)');

readln(l);

end;

if (k>0) then begin //если есть известные углы,то записываем их в массив

for i:= 1 to k do begin

writeln('Введите угол №',+i);

readln(a[i]);

end;

end;

if (l>0) then begin //если есть известные стороны,то записываем их в массив

for i:= 1 to l do begin

writeln('Введите сторону №',+i);

readln(b[i]);

end;

end;

if (k>0) then begin //если есть известные углы,то считаем кол-во различных

e:=1;

for i:=1 to k do begin

w:=0;

for g:= 1 to k do begin

if (a[i] = a1[g]) then

w:=w+1

end;

if (w=0) then begin

a1[e]:=a[i];

e:=e+1;

end;end;end;

e:=e-1;

if (l>0) then begin //если есть известные стороны,то считаем кол-во различных

w:=0;

for i:=1 to l do begin

for g:= 1 to l do begin

if (b[i] = b1[g]) then

w:=w+1;

end;

if (w=0) then begin

b1[j]:=b[i];

j:=j+1;

end;end;end;

j:=j-1;

if (( (k =1) and (a1[1]=90) and (l=4) and (j=1)) or

( (k>1) and (a1[1]=90) and (e=1) and (l>2) and (j=1))) then

writeln('Квадрат')

else if ( ((k =1) and (a1[1]=90) and (l=4) and (j=2)) or

((k>1) and (a1[1]=90) and (e=1) and (l>2) and (j=2)) or

((k>2) and (e=1) and (a1[1]=90) )) then

writeln('Прямоугольник')

else if ( (k>1) and (l>1) and (j=1)) then begin

if ((k>2) or (l>2)) then writeln('Ромб');

end

else if ( ((k>2) and (e=2)) or ((k=2) and (e=2) and (l=3) and (j=2)) or

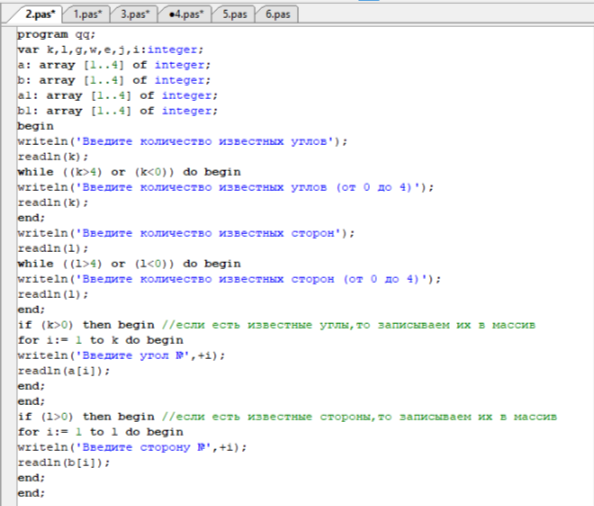
((l=4) and (j=2)) ) then

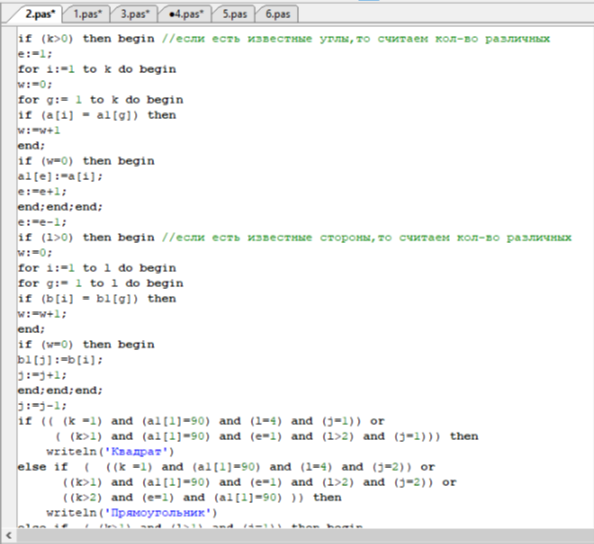
writeln('Параллелограмм')

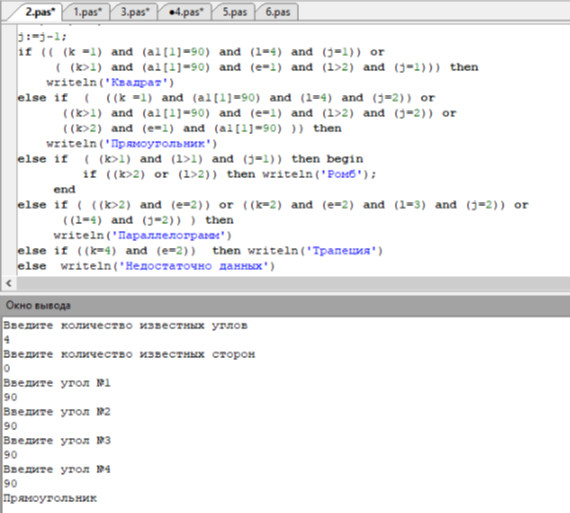
else if ((k=4) and (e=2)) then writeln('Трапеция')

else writeln('Недостаточно данных')

end.







**Задание 3**

Разработать программу, которая генерирует неравномерный двоичный код, допускающий однозначное декодирование. Количество возможных кодов определяется по первым двум стартовым значениям из некоторого количества бит. Первое значение из двух начинается с 0, второе с 1. Максимальная длина неравномерного кода определятся наибольшей длиной одного из первых двух кодов.

**Пример ввода:**

Стартовые значения кода: 0, 100

**Пример вывода:**

Возможные коды: 100, 101, 110 111, 0

program qq;

var x,y,s:string;

i,k:integer;

procedure ss(a:string);

begin

if((a<>x) and (a<>y) and (length(a)<k)) then begin //рекурсивная функция

//если наше число не равно ни x ни н и его длинна не больше максимальной ,то

ss(a+'1');//продолжаем рекурсивную

ss(a+'0');

end

else if ((length(a)=k) or (a=x) or (a=y)) then //иначе если число совпадает с ведённым

// или его длина максимальная то выводи его

write(' '+a);

end;

begin

writeln('Введите x'); //считываем число в двоичной записи

//и записываем в строку x

readln(x);

for i:=1 to length(x) do

if ((x[i]<>'1') and (x[i]<>'0')) then begin

writeln('Введите x');

readln(x);

end;

writeln('Введите y');//считываем числj в двоичной записи

//и записываем в строку x

readln(y);

for i:=1 to length(y) do

if ((y[i]<>'1') and (y[i]<>'0')) then begin

writeln('Введите y');

readln(y);

end;

if(length(x)>length(y)) then //определяем наибольшую длинну

k:=length(x)

else

k:=length(y);

s:='1';

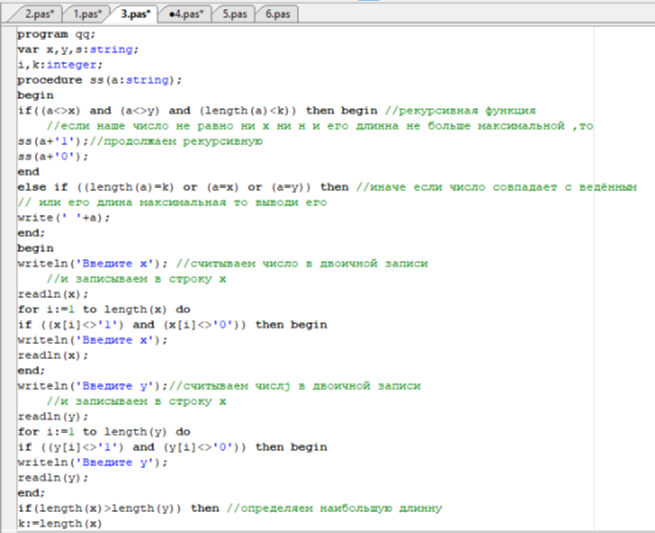
write('Возможные коды:');

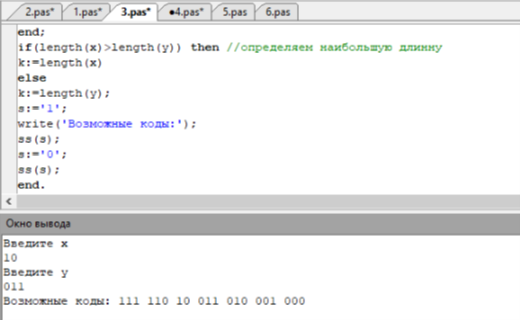
ss(s);

s:='0';

ss(s);

end.





**Задание 4**

Разработать программу, которая по заданным IP-адресу узла и маске подсети определяет десятичные значения байтов в адресе сети (сетевом адресе).

**Пример ввода:**

IP-адрес узла: 209.135.197.15

Маска подсети: 255.255.120.0

**Пример вывода:**

Сетевой адрес: 209.135.64.0

program qq;

var ip, ms,x,y,l:string;

k,i,a,b,j:integer;

t:real;

begin

writeln('Введите IP-адрес узла(в формате 255.255.255.255)');

Readln(ip);

writeln('Введите Маску подсети(в формате 255.255.255.255)');

Readln(ms);

write('Сетевой адрес: ');

for i:= 1 to 4 do begin

x:='';

l:='';

y:='';

if(i<4) then begin //если точка встречалась меньше четырёх раз,то

a:=strtoint(copy(ip,1,pos('.',ip)-1));//вырезаем первое число до точки из узла

delete(ip,1,pos('.',ip));

b:=strtoint(copy(ms,1,pos('.',ms)-1));//вырезаем первое число до точки из маски

delete(ms,1,pos('.',ms));

end

else begin //иначе берём число целиком

a:=strtoint(ip); //перводим из строковой переменной в действительную

b:=strtoint(ms);

end;

while(a>1) do begin //переводим в двоичную систему счисления

x:=inttostr(a mod 2)+x;

a:=a div 2;

end;

x:=inttostr(a)+x;

while(b>1) do begin

y:=inttostr(b mod 2)+y;

b:=b div 2;

end;

y:=inttostr(b)+y;

while (length(x)<8 ) do//если длинна меньше восьми то приписываем в начало нули

x:='0'+x;

while (length(y)<8 ) do

y:='0'+y;

for k:=1 to 8 do

l:=l+inttostr(strtoint(x[k])\*strtoint(y[k])); // проводим поразрядную конъюкцию

b:=7;

t:=0;

for j:= 1 to 8 do begin

t:=t+(strtoint(l[j])\*exp(b\*ln(2)));//переводим в десятичную систему счисления

b:=b-1;

end;

if(i<4) then

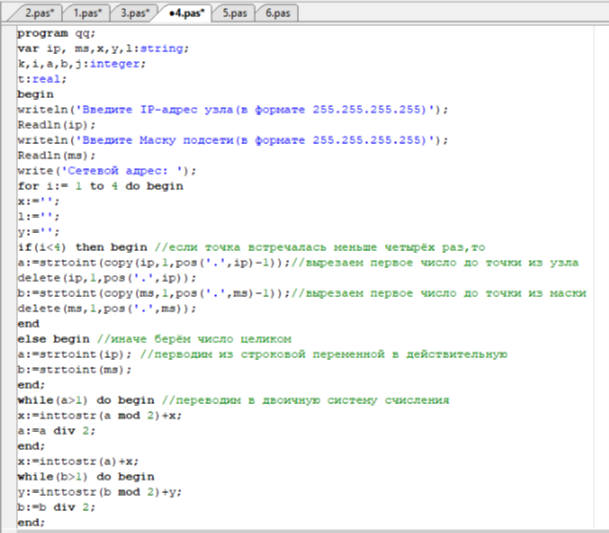
write(t,'.')

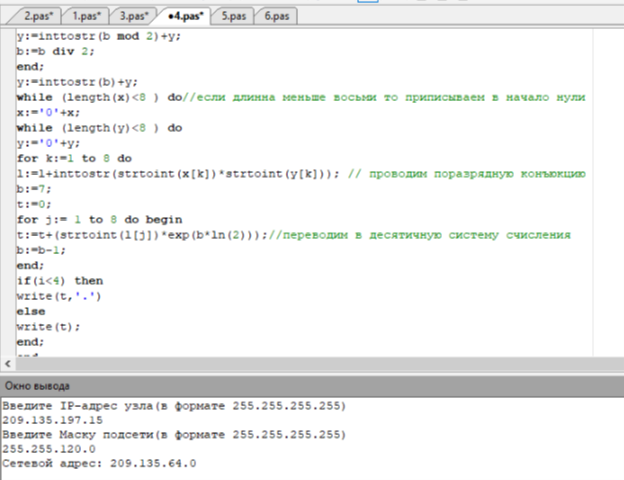
else

write(t);

end;

End.





**Задание 5**

Разработать программу, которая определяет информационный объем всех результатов наблюдений влажности воздуха на метеорологической станции в байтах для некоторого фиксированного количества измерений. Результатом одного измерения влажности воздуха является целое число от 0 до 100 процентов, которое записывается при помощи минимально возможного количества бит.

**Пример ввода:**

Количество измерений: 160

**Пример вывода:**

Информационный объем результатов наблюдений в байтах: 140

program kk;

var r:integer;

begin

writeln('Количество измерений:');

readln(r);

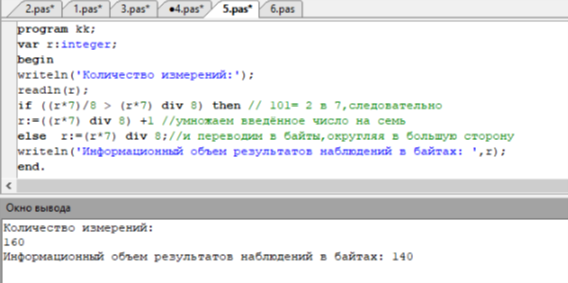
if ((r\*7)/8 > (r\*7) div 8) then // 101= 2 в 7,следовательно

r:=((r\*7) div 8) +1 //умножаем введённое число на семь

else r:=(r\*7) div 8;//и переводим в байты,округляя в большую сторону

writeln('Информационный объем результатов наблюдений в байтах: ',r);

end.



**Задание 6**

Разработать программу, которая подсчитывает количество нуклеотидов для заданной в программе строки. Подсчитать количество нуклеотидов A, C, G, T, разделив пробелами получаемые значения.

**Пример ввода:**

Дана половинка ДНК, в строковой переменной в программе присвоено значение последовательности нуклеотидов:

AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC

**Пример вывода:**

20 12 17 21

program gg;

var s:string;

i,a,c,g,t:integer;

begin

s:='AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC';

for i:=1 to length(s) do begin //перебираем все элементы строки и считаем

if (s[i]='A') then // кол-во A,C,G,T

a:=a+1

else if (s[i]='C') then

c:=c+1

else if (s[i]='G') then

g:=g+1

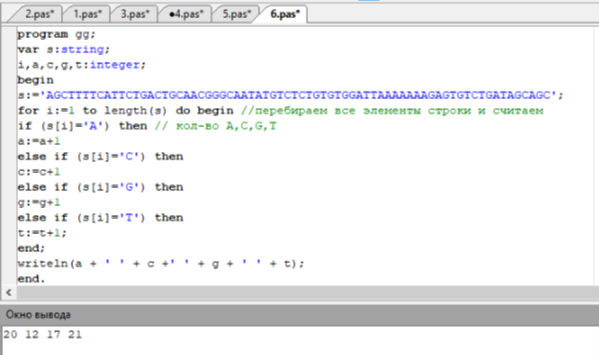
else if (s[i]='T') then

t:=t+1;

end;

writeln(a + ' ' + c +' ' + g + ' ' + t);

End.



**Задание 7**

Разработать программу, которая выполняет построение комплементарной цепочки ДНК для заданной строки ДНК: AAAACCCGGT. Для построение комплементарной цепочки ДНК нужно прочитать цепочку задом наперед, заменив аденин на тимин, а цитозин на гуанин.

**Пример ввода:**

Строковой переменной в программе присвоено значение:

AAAACCCGGT

**Пример вывода:**

ACCGGGTTTT

program gg;

var s,S1:string;

i:integer;

begin

s:='AAAACCCGGT';

for i:=1 to length(s) do begin//перебираем все элементы строки и заменяем

if (s[i]='A') then//согласно условию

s[i]:='T'

else if (s[i]='C') then

s[i]:='G'

else if (s[i]='G') then

s[i]:='C'

else if (s[i]='T') then

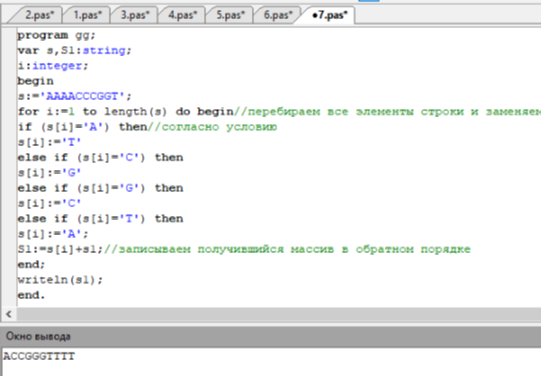
s[i]:='A';

S1:=s[i]+s1;//записываем получившийся массив в обратном порядке

end;

writeln(s1);

End.



program gg;

var str2,str1:string;

i,k:integer;

begin

str1:='GAGCCTACTAACGGGAT';

str2:='CATCGTAATGACGGCCT';

for i:=1 to length(str1) do begin

if (str1[i]<>str2[i]) then //если элементы двух массивов под одним и тем же индексом

//не совпадают,то

k:=k+1;

end;

writeln ('Расстояние Хэмминга: '+k);

end.

**Задание 8**

Разработать программу, которая выполняет подсчет количества точечных мутаций.

**Пример ввода:**

Переменной Str1 в программе присвоено значение:

GAGCCTACTAACGGGAT

Переменной Str1 в программе присвоено значение:

CATCGTAATGACGGCCT

**Пример вывода:**

Расстояние Хэмминга: **7**

program gg;

var str2,str1:string;

i,k:integer;

begin

str1:='GAGCCTACTAACGGGAT';

str2:='CATCGTAATGACGGCCT';

for i:=1 to length(str1) do begin

if (str1[i]<>str2[i]) then //если элементы двух массивов под одним и тем же индексом

//не совпадают,то

k:=k+1;

end;

writeln ('Расстояние Хэмминга: '+k);

end.

