**II Тур олимпиады по программированию, 2016-2017 уч.год**

**Галляува Ильнара Робертовна, 11 класс,**

**образовательное учреждение: 452558, РБ Мечетлинский район,с.Алегазово, ул.Советская, 4, МОБУ СОШ с.Алегазово. e-mail- gallyauva-i@yandex.ru**

Язык Pascal, выполнено в Lazarus IDE v1.2.4 Консольное приложение.

**Задание 1**

Разработайте программу, которая шифрует и дешифрует текст, используя шифр Цезаря для заданного сдвига номера каждой буквы алфавита на фиксированное число позиций от 1 до 20.

**Примеры входных и выходных данных:**

Входные данные

Шифрование - 1, дешифрование – 2: 1

Сдвиг (1-20): 1

Текст: Дорогу осилит идущий.

Выходные данные

Шифрование: Епспдф птймйу йефъйк.

Входные данные

Шифрование - 1, дешифрование – 2: 2

Сдвиг (1-20): 1

Текст: Епспдф птймйу йефъйк.

Выходные данные

Дешифрование: Дорогу осилит идущий.

*Оценка снижается на 30%, если программа правильно выполняет только одно действие из двух – шифрует или дешифрует.*

*Оценка снижается на 10%, если программа не позволяет использовать разный регистр букв.*

**Решение**

|  |
| --- |
| Фраза «Дорогу осилит идущий.» с использованием шифра Цезаря со сдвигом k=1 будет закодирована так: «Епспдф птймйу йефъйк.»  Если первая буква алфавита имеет код 1, вторая – код 2 и т.д. (в данном случае под кодом понимается порядковый номер буквы в заданном нами алфавите, он не совпадает с машинным кодом буквы), алгоритм шифрования может быть выражен формулой y=(x+k) mod n, где x – код исходного символа, k – величина сдвига, y – код символа замены, n – количество символов в алфавите, а запись ((**x+k) mod n**) обозначает остаток от деления **(x+k)**  на **n**, что необходимо для замыкания алфавита в кольцо. Мы используем алфавит без буквы ё, поэтому с учётом регистра у нас 64 буквы.  Сдвиг **k** служит ключом для дешифровки, а формула **x=(y-k+n) mod n** является формулой для дешифровки.  **Задача простая, но долго не получался результат из-за таких малостей как:**  1)Паскаль не различает большие и маленькие буквы (чтобы программа работала с русскими буквами включаем кодировку CP866), поэтому надо использовать машинный код букв, который учитывает регистр букв;  2)если работать с входным файлом, в котором задать текст для шифрования, то (оказалось после многочисленных проверок) машинные коды букв в самой программе отличаются от машинных кодов букв в текстовом документе (и выходной файл оказывался пустым); поэтому, если работать с файлом, надо во входном файле кроме текста для шифрования задать и алфавит;  Проще работать без файлов и алфавит задать внутри программы, а программа должна запрашивать тексты для шифровки и дешифровки.  3)Следующая трудность, которая не позволяла привести к удачному разрешению программы, это полный внутренний цикл. Внешний цикл составлен по длине исходного текста, поэтому вначале находим длину исходного предложения **t:=Length(s1).** Во внутреннем цикле машинный код каждой буквы из шифруемого текста сверяется с машинными кодами букв в алфавите  **if (ord(s1[i])=ord(s2[f]))**  (в нашем случае в алфавите 64 буквы и итерация будет происходить **64\*t** раз, где **t**-длина строки исходного текста; поэтому во внутреннем цикле надо поставить оператор **break** для выхода из внутреннего цикла в случае, когда найдено совпадение машинных кодов букв.  **Вариант I**  Таким образом, готовая программа имеет вид:  **program** project1;  const n=64; k=1;  **var**  s1,s2: string;  i,f:integer;  t:integer;  x,y,z:integer;  **begin** s2:='АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдежзийклмнопрстуфхцчшщъыьэюя'; //алфавит  //Организация шифрования  write('Введите текст для шифрования: ');  readln(s1); //ввод текста для шифровки  t:=Length(s1); i:=1;  **while i<= t do begin**  **for** f:=1 **to** n **do**  **if** (ord(s1[i])=ord(s2[f])) **then //**проверка совпадений машинных кодов в шифруемом тексте и в алфавите  **begin**  x:=f; y:=(x+k) mod n; s1[i]:=s2[y]; **break**;  **end**;  i:=i+1;  **end;**  writeln('Шифрование: ',s1); //вывод зашифрованного текста  writeln; writeln; //вывод двух пустых строк  //Организация дешифрования  write('Введите текст для дешифрования: ');  readln(s1);  t:=Length(s1); i:=1;  while i<= t do begin  for f:=1 to n do  if (ord(s1[i])=ord(s2[f])) then  begin  y:=f; x:=(y-k+n) mod n; s1[i]:=s2[x]; break;  end; i:=i+1;  end;  write('Дешифрование: ',s1); //вывод дешифрованного текста  readln;readln;  **end.**  C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\уменьшенные\1-1.png  **II вариант с добавлением выбора case**  **program project1;**  **const** n=64;  **var**  s1,s2: string;  i,f:integer;  k,m:integer;  t:integer;  x,y:integer;  **begin** s2:='АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдежзийклмнопрстуфхцчшщъыьэюя'; //алфавит  //Организация шифрования  **write ('Шифрование - 1, дешифрование 2: '); readln(m);**  **write ('Сдвиг (1-20): '); readln(k);**  **case m of**  **1:** begin write('Текст: ');  readln(s1); //ввод текста для шифровки  t:=Length(s1); i:=1;  while i<= t do begin  for f:=1 to n do  if (ord(s1[i])=ord(s2[f])) then //проверка совпадений машинных кодов в шифруемом тексте и в алфавите  begin  x:=f; y:=(x+k) mod n; s1[i]:=s2[y]; break;  end;  i:=i+1;  end;  writeln('Шифрование: ',s1); //вывод зашифрованного текста  end;  //Организация дешифрования  **2:** begin write('Текст: ');  readln(s1);  t:=Length(s1); i:=1;  while i<= t do begin  for f:=1 to n do  if (ord(s1[i])=ord(s2[f])) then  begin  y:=f; x:=(y-k+n) mod n; s1[i]:=s2[x]; break;  end; i:=i+1;  end;  write('Дешифрование: ',s1); //вывод дешифрованного текста  end;  **end;**  readln;readln;  **end.**    C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\2-1.png  C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\3-1.png |

**Задание 2**

Разработайте программу, производящую расчет количества любых слов и различных букв в предлагаемой фразе, а также количество различных букв в словах.

**Пример входных и выходных данных:**

Входные данные

Строка текста: Viam supervadet vadens.

Выходные данные

Кол-во слов - 3. Всего кол-во различных букв - 12.

Viam - 4 различные буквы.

supervadet - 9 различных букв.

vadens - 6 различных букв.

*Оценка снижается на 20%, если программа не учитывает повтор одинаковых букв в слове.*

*Оценка снижается на 15%, если в качестве разделителя слов используется только пробел.*

*Оценка снижается на 10%, если программа не позволяет использовать разный регистр букв.*

|  |
| --- |
| **Решение на псевдокоде:**  **1.**Задаём два алфавита s2: 26 латинских и 33 русских буквы, итого 59 строчных букв.  **2.**Исходный текст s1 дублируем в s3 (нужен для урезания исходного текста при выделении букв).  **3.**В s7:=LowerCase(s1) и s7:=LowerCase(S[q]) меняем все заглавные буквы на строчные (маленькие), чтобы правильно подсчитать количество различных букв в исходном тексте.  **4**.В массив S выделим отдельные слова из текста.  **5.**При выделении букв в массив S ориентируемся на 7 знаков препинания ( , ? ! - . , : ; ) и пробел. Вначале подсчитываем количество все знаков препинания и пробелов, затем определяем минимальную позицию ближайшего знака (в массиве L записываем ближайшую позицию каждого вида знака для определения min позиции, содержание массива L меняется в каждой итерации после очередного урезания исходного текста s3) (это задача нахождения минимального числа в массиве чисел). Затем копируем слово, находящееся до ближайшего знака или пробела, в массив S, затем вырезаем это слово вместе со знаком; определяем нет ли за вырезанным знаком ещё какой-нибудь знак или пробел, если есть, то удаляем его. Затем операция повторяется: находим ближайший знак из 8-ми, копируем слово, стоящее до этого знака в массив S, вырезаем из строки s3 скопированное слово вместе со знаком, вырезаем сразу следующий за удалённым знаком знак (если он имеется) или пробел (н-р, запятая с пробелом или ?! или … или пробел-тире-пробел) и т.д. Концом этого цикла repeat-until является Length(s3)=0 –нулевая длина обрезаемой строки.  **6.**Последний цикл **case** нужен для правильного вывода окончаний (1 буква, 4 буквы, 9 букв).  **Программа правильно работает как с русским текстом, так и с латинским.**  **Решение**  **program project1;**  const n=59;  **var**  s1,s2,s3,s7: string;  A:array [1..n] of integer;//для подсчёта количества различных букв  B:array [1..n] of integer;//для подсчёта количества различных букв в словах  S:array [1..n] of string; //для выделения слов  L: array [1..8] of integer;//для определения позиции знака препинания  d,k,t,i,j:integer;  h,q,c,min,v:integer;  **begin**  s2:='абвгдеёжзийклмнопрстуфхцчшщъыьэюяabcdefghijklmnopqrstuvwxyz ';//33 русских и 26 латинских букв, всего 59  for i:=1 to n do A[i]:=0;  write('Введите текст: ');  readln(s1);  s3:=s1;//сохраним введённый текст  t:=Length(s1);//определяем количество символов в предложении  for i:=1 to n do S[i]:=''; //обнуление массива S  if (s1='.') or (s1='')or (s1=' ') then writeln('Неверный ввод. Введите текст заново. ') else  begin //Определение количества слов в тексте  d:=0;  for i:=1 to t do  if (s1[i]=' ') or (s1[i]=',') or (s1[i]=':') or (s1[i]=';') or (s1[i]='-') or (s1[i]='.') or (s1[i]='!') or (s1[i]='?') then d:=d+1;//количество всех знаков препинания  c:=1;  repeat //формирование массива L со словами  min:=t;  L[1]:=Pos('.',s3); L[2]:=Pos(' ',s3); L[3]:=Pos(',',s3); L[4]:=Pos(':',s3);  L[5]:=Pos(';',s3); L[6]:=Pos('!',s3); L[7]:=Pos('?',s3); L[8]:=Pos('-',s3);  for j:=1 to 8 do if (L[j]<min) and (L[j]<>0) then min:=L[j];//определение позиции ближайшего знака препинания, где условие (L[i]<>0) - это исключение отсутствия знака препинания  if min>1 //проверка, не стоит ли знак препинания первым  then begin  S[c]:=Copy(s3,1,min-1);//сохранение слова до первого знака препинания  Delete( s3,1,min);//вырезание из текста сохранённого слова вместе с первым знаком препинания  c:=c+1; //подсчёт количества слов  end else  Delete( s3,1,1);//вырезание знака препинания, если он стоит первым  until Length(s3)=0;//условием окончания будет служить нулевая длина строки  end;  s7:=LowerCase(s1); //заменим в s1 все заглавные буквы на строчные  //Подсчитаем количество различных букв в предложении  i:=1;  while i<= n do begin  for j:=1 to t do  if s2[i]=s7[j] then A[i]:=A[i]+1;  i:=i+1;  end;  v:=0;  for i:=1 to n do  if A[i]<>0 then v:=v+1;  k:=t-d;//количество всех букв  writeln('Количество слов - ',c-1,'.');  writeln('Количество всех букв в тексте - ',k,'.');  writeln('Количество различных букв в тексте - ',v,'.');  **for q:=1 to c do begin**  //Вывод количества различных букв в каждом слове  **case** Length(S[q]) **of**  1: begin  for j:=1 to n do B[j]:=0; //обнуляем массив  i:=1;  while i<= n do begin  for j:=1 to Length(S[q])do begin  s7:= LowerCase(S[q]); //переводим все буквы в слове в нижний регистр  if s2[i]=s7[j] then B[i]:=B[i]+1;  end; i:=i+1;  end;  h:=0;  for i:=1 to n do  if B[i]<>0 then h:=h+1;  writeln(S[q],' - ',h,' буква.');  end;  2..4:begin  for j:=1 to n do B[j]:=0; //обнуляем массив  i:=1;  while i<= n do begin  for j:=1 to Length(S[q])do begin  s7:= LowerCase(S[q]); //переводим все буквы в слове в нижний регистр  if s2[i]=s7[j] then B[i]:=B[i]+1;  end; i:=i+1;  end;  h:=0;  for i:=1 to n do  if B[i]<>0 then h:=h+1;  writeln(S[q],' - ',h,' различные буквы.');  end;  5..20:begin  for j:=1 to n do B[j]:=0; //обнуляем массив  i:=1;  while i<= n do begin  for j:=1 to Length(S[q])do begin  s7:= LowerCase(S[q]); //переводим все буквы в слове в нижний регистр  if s2[i]=s7[j] then B[i]:=B[i]+1;  end; i:=i+1;  end;  h:=0;  for i:=1 to n do  if B[i]<>0 then h:=h+1;  writeln(S[q],' - ',h,' различных букв.');  end;  **end;**  **end;**  readln;readln;  **end.**  Программа анализирует и несколько предложений с использованием 7-ми различных знаков препинания.  В программе допускается, что длина слова <=20 символов, количество слов в тексте <=59. Никаких оговорок насчёт этого в условии задачи не было. Подразумевалось, что будет одно предложение. Задача выполнена.  **Скриншот к задаче 2**  **C:\Documents and Settings\Admin\Рабочий стол\К задаче 2 упрощ программа-1.png** |

**Задание 3**

Разработайте программу вычисления числа Пи методом Монте-Карло с помощью отношения числа удачных выстрелов к общему числу выстрелов по круглой мишени. Используйте функцию генератора случайных чисел для имитации координат попаданий и не менее 10000000 испытаний.

Пример входных и выходных данных:

Входные данные

Input number of tests: 10000000

Выходные данные

Pi = 3.14113

*Оценка снижается на 20%, если программа не позволяет использовать 10000000 испытаний.*

**Решение I**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **program** project1;  **var** n,n1,i:integer;  x,y,p:real;  **begin**  assign(input,'in.txt');reset(input);  assign(output,'out.txt');rewrite(output);  readln(n);//задание числа бросаний  randomize; // генератор случайных чисел  //получение в цикле n случайных точек и проверка, попадают ли они в круг, если попадают, то подсчитывается число попаданий  n1 := 0;  for i:=1 to n do  begin  // получаем случайную точку с координатами (-1; 1):  x:=random\*2-1; y:=random\*2-1;  //проверяем, попадает ли точка с координатами (x;y) в круг радиуса r=1  if(x\*x+y\*y<=1) then n1:= n1 + 1;  end;  //получение случайного значения числа pi  p:=4\*n1/n;  //вывод числа pi в выходной файл  writeln('pi=',p:12:10);  end.   |  |  |  | | --- | --- | --- | | in.txt | out.txt | Время выполнения | | 100000000 | pi=3.1411603600 | 7 сек | | pi=3.1415056000 | 5 сек | | pi=3.1416079200 | 6 сек | | **10000000** | pi=3.1426376000 | 1 сек | | pi=3.1416912000 | 1 сек | | pi=3.1412386800 | 1 сек | | 500000000 | pi=3.1414991360 | 35 сек | | 500000000 | pi=3.1415886720 | 35 сек | | 500000000 | pi=3.1414595360 | 36 сек | | 700000000 | pi=3.1415000200 | 42 сек | | 800000000 | выполняет неправильно | 48 сек | | Методом Монте-Карло вычисления числа pi основан на следующей идее:   1. Круг с радиусом R=1 вписан в квадрат со стороной a=2R. 2. Задаём координаты случайной точки (этим самым бросаем в квадрат камешек...). Случайные число дробные x и y , <=2. Этот случайный «бросок» и роднит метод со столицей азартных игр, где все построено на случайности. 3. Точка может попасть в окружность или нет (условие попадания: x2 + y2<= 1). 4. Подсчитываем количество попаданий n1 при достаточно большом числе бросков n. 5. Если точки случайны, то отношение попавших точек к общему числу попыток равно отношению площадей круга и квадрата. n1 / n = Sкр / S кв = πR2 / (2R)2 6. Число π = 4 \* n1 / n. |
| **Решение II**  **Составим форму, демонстрирующую метод Монте – Карло наглядно попаданием красных камешков в круг, а зелёных – в квадрат.**  **unit** Unit1;  {$mode objfpc}{$H+}  **interface**  **uses**  Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, ExtCtrls,  StdCtrls;  **type**  { TForm1 }  TForm1 = **class(**TForm)  Button1: TButton;  Edit1: TEdit;  Label1: TLabel;  Label2: TLabel;  Label3: TLabel;  Notebook1: TNotebook;  **procedure** Button1Click(Sender: TObject);  **procedure** Edit1Change(Sender: TObject);  **procedure** Edit1KeyPress(Sender: TObject; **var** Key: char);  **private**  { private declarations }  **public**  { public declarations }  **end;**  **var**  Form1: TForm1;  **implementation**  {$R \*.lfm}  { TForm1 }  **procedure** TForm1.Edit1Change(Sender: TObject);  **begin**  **if** (Edit1.Text='') **then** Button1.Enabled:=False  **else** Button1.Enabled:=True;  **end;**  **procedure** TForm1.Edit1KeyPress(Sender: TObject; **var** Key: char);  **begin**  **case** Key **of**  '0'..'9',#8:;  **else** key:=#0;  **end;**  **end;**    **procedure** TForm1.Button1Click(Sender: TObject);  **const**  mash=100;  **var**  a,b:integer; //числа на координатных осях 0x и 0y  x0,y0:integer;// центр системы координат  x,y:real; // координаты точек графика (круга и квадрата)  x1,y1:real; // координаты сетки  s1,s2:char;// символы "x" и "y"  n,n1,i:integer;  p:real;  x2,y2:integer; // координаты случайных точек (камешков)  m1,m2:real; // координаты случайных чисел из интервала (-1;1)  **begin**  //Очистить рисунок при новом построении графика(две строки)  Notebook1.Canvas.Pen.Color:=clWhite;// цвет пера устанавливаем белый, чтобы отменить все старые цвета при перерисовывании нового графика  Notebook1.Canvas.Rectangle(0,0,ClientWidth,ClientHeight);//строим прямоугольник, закрывающий Notebook1 с прежним построением  s1:='x';s2:='y';  //находим центр  x0:=Notebook1.Width div 2;  y0:=Notebook1.Height div 2;  **with** Notebook1.Canvas **do** //Цикл операций для Notebook1.Canvas  **begin**  //рисуем оси  Pen.Color:=clBlack;// цвет осей  Pen.Width:=3;// толщина осей  MoveTo(x0,0); //начало оси 0y  LineTo(x0,ClientHeight);// конец оси 0y  MoveTo(0,y0); //начало оси ox  LineTo(ClientWidth,y0);// конец оси 0x  //Рисуем сетку в положительной области оси 0x  Notebook1.Canvas.Font.Size:=16; //размер цифр  Pen.Color:=clBlack;//устанавливаем цвет оси чёрный  Pen.Width:=1; //толщина оси  x1:=x0;y1:=y0; a:=0;b:=-1;  **repeat**  x1:=x1+1\*mash; y1:=y1+1\*mash;  MoveTo(trunc(x1),0); LineTo(trunc(x1),ClientHeight);  TextOut(trunc(x1)-1\*mash,y0+1\*mash div 5,FloatToStrF(a,ffGeneral,1,1)); //проставляем числа на положительной части оси 0x  MoveTo(0,trunc(y1)); LineTo(ClientWidth,trunc(y1));  TextOut(x0+1\*mash div 5,trunc(y1),FloatToStrF(b,ffGeneral,1,1));//проставляем числа на отрицательной части оси 0y  a:=a+1; b:=b-1;  **until** x1>=ClientWidth;  //Рисуем сетку в отрицательной области оси 0x  x1:=x0;y1:=y0; a:=-1; b:=1;  **repeat**  x1:=x1-1\*mash; y1:=y1-1\*mash;  MoveTo(trunc(x1),0); LineTo(trunc(x1),ClientHeight);  TextOut(trunc(x1-1\*mash div 5),y0+1\*mash div 10,FloatToStrF(a,ffGeneral,1,1)); //проставляем числа на отрицательной части оси 0x  MoveTo(0,trunc(y1)); LineTo(ClientWidth,trunc(y1));  TextOut(x0+1\*mash div 4,trunc(y1-1\*mash div 4),FloatToStrF(b,ffGeneral,1,1));//проставляем числа на положительной части оси 0y  a:=a-1; b:=b+1;  **until** x1<=0;  //Изображение стрелочек на осях  Pen.Color:=clBlack;//цвет  Pen.Width:=3; //толщина  //на оси 0x  MoveTo(trunc(2\*x0-1\*mash div 8),trunc(y0-1\*mash div 12)); Notebook1.Canvas.LineTo(2\*x0,y0); // верхняя  MoveTo(trunc(2\*x0-1\*mash div 8),trunc(y0+1\*mash div 12)); Notebook1.Canvas.LineTo(2\*x0,y0); // нижняя  //на оси 0y  MoveTo(trunc(x0-1\*mash div 12),trunc(0+1\*mash div 8)); Notebook1.Canvas.LineTo(x0,0); // левая  MoveTo(trunc(x0+1\*mash div 12),trunc(0+1\*mash div 8)); Notebook1.Canvas.LineTo(x0,0); // правая  //Надпись "х" и "y" (4 строки)  Font.Style:=[fsBold]; //стиль-полужирный  Font.Size:=16;//размер буквы  TextOut(trunc(2\*x0-1\*mash div 4),trunc(y0-1\*mash div 3),s1); //"x"  TextOut(trunc(x0-1\*mash div 5),trunc(0+1\*mash div 8),s2);//"y"  //Строим квадрат  Pen.Color:=clLime;//цвет салатовый  Pen.Width:=2; //толщина линии  Form1.Notebook1.Canvas.Rectangle(x0-1\*mash,y0-1\*mash,x0+1\*mash,y0+1\*mash);  //Строим круг  Pen.Color:=clFuchsia;//цвет ярко-розовый  Pen.Width:=2; //толщина линии  Form1.Notebook1.Canvas.Ellipse(x0-1\*mash,y0-1\*mash,x0+1\*mash,y0+1\*mash);  n:=StrToInt(Edit1.Text); // считываем число бросаний  randomize; // генератор случайных чисел  //получение в цикле n случайных точек и проверка, попадают ли они в круг, если попадают, то подсчитывается число попаданий  n1 := 0;  **for** i:=1 **to** n **do**  **begin**  // получаем случайную точку с координатами (-1; 1):  m1:=random\*2-1; m2:=random\*2-1; x2:=round(x0+m1\*mash); y2:=round(y0+m2\*mash);  //проверяем, попадает ли точка с координатами (m1;m2) в круг радиуса r=1  if(m1\*m1+m2\*m2<=1) then begin n1:= n1 + 1;  Form1.Notebook1.Canvas.Pixels[x2,y2]:=clRed;//точки красным цветом  end else Form1.Notebook1.Canvas.Pixels[x2,y2]:=clGreen;//точки зелёным цветом  //получение случайного значения числа pi  p:=(4\*n1/n);  //вывод числа pi в выходной файл  Form1.Label3.Caption:=FloatToStr(p);  **end;**  **end;**  **end;end.**  C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\7-1.pngC:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\8-1.png  C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\9-1.pngC:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\10-1.png  C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\11-1.pngC:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\12-1.png  Время выполнения 5 мин  C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\14-1.png  Время выполнения 18 мин | |

**Задание 4**

Разработайте программу, которая, используя рекурсивную функцию, вычисляет факториал числа и определяет максимально возможное значение факториала для используемого алгоритма и системы программирования.

**Примеры входных и выходных данных:**

Входные данные

5

Выходные данные

5!=120

Максимально возможные значения:

12! = 479 001 600

или

15! = 1 307 674 368 000

или

21! = 51 090 942 171 709 440 000

*Оценка снижается на 10%, если программа не вычисляет правильно максимально возможное значение факториала.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Решение**  program project1;  **const d=1000000;**  **var** a:array of longint;//массив динамический  t,x,c,n,k,i,s,r:integer;  **begin**  **//**связь с входным и выходным файлами  assign(input,'in.txt');reset(input);  assign(output,'out.txt');rewrite(output);    readln(n); //вводим из входного файла значение n, факториал которого будем находить  t:=n\*2 div 6;//рассчитываем величину массива a[i]  SetLength(a,t);//задаём длину массива  a[0]:=1; //записываем в младшую ячейку массива 1  for i:=1 to t do a[i]:=0;//в остальные переменные массива a[i] записываем нули  r:=0;//начальное значение старшего разряда  //Цикл нахождения n!=1\*2\*3\*…\*n постепенным умножением 1 на все числа, начиная с единицы и разбиение числа по ячейкам массива  **for k:=2 to n do begin**  for i:=0 to t do begin  s:=a[i]\*k+r;  a[i]:=s mod d;  r:=s div d;  end;**end;**  //Вывод членов массива в одну строку слева направо  i:=t;  while a[i]=0 do  i:=i-1;  if **n>11** then writeln('Максимально возможные значения: ');  write(n,' ! = ',a[i]);  write(a[i]);  for k:=i-1 downto 0 do **begin**  x:=a[k];  c:=100000;  while c>0 do begin  write(x div c);  x:=x mod c;  c:=c div 10;  end;  **end;**  close(input); close(output);  **end.**  **Результат работы программы**   |  |  | | --- | --- | | **in.txt** | **out.txt** | | **5** | **5!=120** | | **16** | **Максимально возможные значения:**  **16!= 20922789888000** | | **105** | **Максимально возможные значения:**  **105 != 108139675824029090050410130580032964972064610777**  **490257914417663657322653190990515332698453652680824033**  **97763989348702965799387290781343681609728000000000000**  **0000000000000** | | **10** | **10 ! = 3628800** | | **11** | **11 ! = 39916800** | | **12** | **Максимально возможные значения:**  **12 ! = 479001600** | | **13** | **Максимально возможные значения:**  **13 ! = 6227020800** | | **Заявленные переменные:**  Задача относится к разряду длинных чисел, которые не помещаются в переменную стандартного типа. Поэтому выводимое число мы разбиваем на группы по 6 чисел и выводим эти группы в массив a[i]. Фактически мы получившееся число записываем в системе счисления с основанием d=1000000.  t - число членов массива;  x - переменная используется для вывода шести чисел в ячейки массива a[i], вначале x:=a[k], затем операцией write(x div c) выводится остаток от деления на **с**; таким образом мы выводим в ячейки массива a[i] число, начиная с конца;  c - переменная используется для вывода шести чисел в ячейки массива a[i]:  x:=a[k];  c:=100000;  while c>0 do begin  write(x div c);  x:=x mod c;  c:=c div 10;  end;  n - число, факториал которого находим (н-р, n=5 если ищем 5!);  k – счётчик цикла для n чисел;  i - счётчик цикла для t чисел;  s – результат умножения длинного числа на короткое;  r – число, переносимое в старший разряд. |

**Обычное решение**

Вообще, **обычный алгоритм** вычисляет n! при n=12 ещё правильно. А вот дальше…

**program** project1;

var n,i,factorial:longint;

**begin**

assign(input,'in.txt');reset(input);

assign(output,'out.txt');rewrite(output);

read(n); factorial:=1;

for i:=1 to n do factorial:=factorial\*i;

write(n,' ! = ',factorial); close(input); close(output);

**end.**

**Результат работы программы обычного алгоритма**

|  |  |
| --- | --- |
| **in.txt** | **out.txt** |
| **5** | **5 ! = 120** |
| **10** | **10 ! = 3628800** |
| **11** | **11 ! = 39916800** |
| **12** | **12 ! = 479001600** |
| **13** | **13 ! = 1932053504 уже с ошибкой** |

**Задание 5**

Разработайте программу, которая после выбора функций выводит графики трех функций (y=kx, y=k\*sin(x), y=k\*x^2 ) в одной системе координат.

**Примеры входных и выходных данных:**

Входные данные

Какие функции выбираем (1, 2, 3)? 2, 3

Введите k2, x2, k3, x3

Выходные данные

Изображение графиков функций y=k\*sin(x) и y=k\*x^2.

Допускается выбор функций по значениям k не равным 0.

*Оценка снижается на 30%, если программа искажает графики (соотношение высоты и ширины графика).*

*Оценка снижается на 20%, если программа не выводит координатные оси.*

*Оценка снижается на 10%, если программа не выводит числовые метки на координатных осях.*

|  |
| --- |
| **Решение в форме** **Form1**  **(из соображения наглядности и практического применения)**  C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\15-1.png  **unit Unit1**;  {$mode objfpc}{$H+}  interface  **uses**  Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, ExtCtrls,  StdCtrls;  **type**  { TForm1 }  TForm1 = class(TForm)  Button1: TButton;  Edit1: TEdit;  Edit2: TEdit;  Edit3: TEdit;  Edit4: TEdit;  Image1: TImage;  Label1: TLabel;  Label2: TLabel;  Label3: TLabel;  Label4: TLabel;  Label5: TLabel;  Label6: TLabel;  Label7: TLabel;  Label8: TLabel;  Label9: TLabel;  Notebook1: TNotebook;  RadioButton1: TRadioButton;  RadioButton2: TRadioButton;  RadioButton3: TRadioButton;  procedure Button1Click(Sender: TObject);  procedure Edit1Change(Sender: TObject);  procedure Edit1KeyPress(Sender: TObject; var Key: char);  procedure Edit2Change(Sender: TObject);  procedure Edit2KeyPress(Sender: TObject; var Key: char);  procedure Edit3Change(Sender: TObject);  procedure Edit3KeyPress(Sender: TObject; var Key: char);  procedure Edit4Change(Sender: TObject);  procedure Edit4KeyPress(Sender: TObject; var Key: char);  **private**  { private declarations }  **public**  { public declarations }  **end;**  **var**  Form1: TForm1;  implementation  {$R \*.lfm}  { TForm1 }  **procedure TForm1.Button1Click(Sender: TObject);**  **var**  a,b:integer; //числа на координатных осях 0x и 0y  mash:integer; // масштаб  k1,k2,k3:real;// коэффициенты графиков  x0,y0:integer;// центр системы координат  x,y:real; // координаты точек графика  x1,y1:real; // координаты сетки  s1,s2:char;// символы "x" и "y"  **begin**  //Очистить рисунок при новом построении графика(две строки)  Notebook1.Canvas.Pen.Color:=clWhite;// цвет пера устанавливаем белый, чтобы отменить все старые цвета при перерисовывании нового графика  Notebook1.Canvas.Rectangle(0,0,ClientWidth,ClientHeight);//строим прямоугольник, закрывающий Notebook1 с прежним построением  mash:=StrToInt(Edit4.Text); // считываем масштаб  s1:='x';s2:='y';  //находим центр  x0:=Notebook1.Width div 2;  y0:=Notebook1.Height div 2;  **with Notebook1.Canvas do** //Цикл операций для Notebook1.Canvas  **begin**  //рисуем оси  Pen.Color:=clBlack;// цвет осей  Pen.Width:=3;// толщина осей  MoveTo(x0,0); //начало оси 0y  LineTo(x0,ClientHeight);// конец оси 0y  MoveTo(0,y0); //начало оси ox  LineTo(ClientWidth,y0);// конец оси 0x  **//Рисуем сетку в положительной области оси 0x**  Notebook1.Canvas.Font.Size:=7; //размер цифр  Pen.Color:=clBlack;//устанавливаем цвет оси чёрный  Pen.Width:=1; //толщина оси  x1:=x0;y1:=y0; a:=0;b:=-1;  **repeat**  x1:=x1+1\*mash; y1:=y1+1\*mash;  MoveTo(trunc(x1),0); LineTo(trunc(x1),ClientHeight);  TextOut(trunc(x1)-1\*mash,y0+1\*mash div 2,FloatToStrF(a,ffGeneral,1,1)); //проставляем числа на положительной части оси 0x  MoveTo(0,trunc(y1)); LineTo(ClientWidth,trunc(y1));  TextOut(x0+1\*mash div 2,trunc(y1),FloatToStrF(b,ffGeneral,1,1));//проставляем числа на отрицательной части оси 0y  a:=a+1; b:=b-1;  **until x1>=ClientWidth;**  **//Рисуем сетку в отрицательной области оси 0x**  x1:=x0;y1:=y0; a:=-1; b:=1;  **repeat**  x1:=x1-1\*mash; y1:=y1-1\*mash;  MoveTo(trunc(x1),0); LineTo(trunc(x1),ClientHeight);  TextOut(trunc(x1),y0+1\*mash div 2,FloatToStrF(a,ffGeneral,1,1)); //проставляем числа на отрицательной части оси 0x  MoveTo(0,trunc(y1)); LineTo(ClientWidth,trunc(y1));  TextOut(x0+1\*mash div 2,trunc(y1),FloatToStrF(b,ffGeneral,1,1));//проставляем числа на положительной части оси 0y  a:=a-1; b:=b+1;  **until x1<=0;**  **//Изображение стрелочек на осях**  Pen.Color:=clBlack;//цвет  Pen.Width:=3; //толщина  **//на оси 0x**  MoveTo(trunc(2\*x0-1\*mash div 2),trunc(y0-1\*mash div 3)); Notebook1.Canvas.LineTo(2\*x0,y0); // верхняя  MoveTo(trunc(2\*x0-1\*mash div 2),trunc(y0+1\*mash div 3)); Notebook1.Canvas.LineTo(2\*x0,y0); // нижняя  **//на оси 0y**  MoveTo(trunc(x0-1\*mash div 3),trunc(0+1\*mash div 2)); Notebook1.Canvas.LineTo(x0,0); // левая  MoveTo(trunc(x0+1\*mash div 3),trunc(0+1\*mash div 2)); Notebook1.Canvas.LineTo(x0,0); // правая  //Надпись "х" и "y" (4 строки)  Font.Style:=[fsBold]; //стиль-полужирный  Font.Size:=12;//размер буквы  TextOut(trunc(2\*x0-1\*mash),trunc(y0-1\*mash),s1); **//"x"**  TextOut(trunc(x0-1\*mash),trunc(0+1\*mash div 2),s2);**//"y"**  **//Строим график**  Pen.Color:=clRed;  Pen.Width:=2;  **if** RadioButton1.Checked **then begin**  k1:=StrToFloat(Edit1.Text); // считываем коэффициент первого графика  **if k1<>0 then begin**  x:=-100; y:=k1\*x;  MoveTo(trunc(x),trunc(y));  **repeat**  x:=x+0.1; y:=-k1\*x;  LineTo(trunc(x\*mash)+x0,trunc(y\*mash)+y0);  **until x>=100; end;**  **end**  **else begin**  **if** RadioButton2.Checked **then begin**  k2:=StrToFloat(Edit2.Text); // считываем коэффициент второго графика  **if** k2<>0 **then begin**  x:=-100; y:=k2\*sin(x);  MoveTo(trunc(x),trunc(y));  **repeat**  x:=x+0.1; y:=-k2\*sin(x);  LineTo(trunc(x\*mash)+x0,trunc(y\*mash)+y0);  **until x>=100; end;**  **end**  **else**  **if** RadioButton3.Checked **then begin**  k3:=StrToFloat(Edit3.Text); // считываем коэффициент третьего графика  **if** k3<>0 **then begin**  x:=-100; y:=k3\*x\*x;  MoveTo(trunc(x),trunc(y));  **repeat**  x:=x+0.1; y:=-k3\*x\*x;  LineTo(trunc(x\*mash)+x0,trunc(y\*mash)+y0);  **until x>=100; end;**  **end;**  **end;**  **end;**  **end;**  **//Процедуры: на пустые окна Edit, на ограничения для ввода в окна**  //если **изменения в Edit1** будут пустой строкой, то заблокировать кнопку Button1  **procedure** TForm1.**Edit1Change**(Sender: TObject);  **begin**  **if** (Edit1.Text='') **then** Button1.Enabled:=False  else Button1.Enabled:=True;  **end;**  //разрешение на ввод в Edit1 цифр, знака "-", запятой и действие кнопки Backspase  **procedure** TForm1.Edit1KeyPress(Sender: TObject; var Key: char);  **begin**  **case** **Key of**  '0'..'9',#8,'-',',':;  **else** key:=#0;  **end;**  **end;**  **procedure** TForm1.Edit2Change(Sender: TObject);  **begin**  **if** (Edit2.Text='') **then** Button1.Enabled:=False  **else** Button1.Enabled:=True;  **end;**  **procedure** TForm1.Edit2KeyPress(Sender: TObject; var Key: char);  **begin**  **case Key of**  '0'..'9',#8,'-',',':;  **else** key:=#0;  **end;**  **end;**  **procedure** TForm1.Edit3Change(Sender: TObject);  **begin**  **if** (Edit3.Text='') **then** Button1.Enabled:=False  **else** Button1.Enabled:=True;  **end;**  **procedure** TForm1.Edit3KeyPress(Sender: TObject; var Key: char);  **begin**  **case Key of**  '0'..'9',#8,'-',',':;  **else** key:=#0;  **end;**  **end;**  **procedure** TForm1.Edit4Change(Sender: TObject);  **begin**  **if** (Edit4.Text='') **then** Button1.Enabled:=False  **else** Button1.Enabled:=True;  **end;**  **procedure** TForm1.Edit4KeyPress(Sender: TObject; var Key: char);  **begin**  **case Key of**  '0'..'9',#8,'-',',':;  **else** key:=#0;  **end;**  **end;**  **end.**    **Дополнительные пояснения:**   1. Ограничение на количество вводимых в окно **Edit** символов вводится в свойстве **MaxLength** (в нашем случае «**5»** для коэффициентов **k1, k2, k3** и **«2»** для масштаба **mash**) при выделенном на форме соответствующем окне **Edit**. 2. Графики созданы в ворде, сохранены как рисунок, а затем введены на компонент **Image1** с помощью свойства **Picture** компонента **Image1,** с использованием свойства **AutoSize** и выбора **True**  согласуем размеры рисунка с компонентом **Image1**. 3. k1:=StrTo**Float**(Edit1.Text); считывание строки в окне Edit1 и перевод в десятичное число 4. mash:=StrTo**Int**(Edit4.Text); аналогичная операция для целого числа 5. выбор одной радиокнопки из трёх:   **if** RadioButton1.Checked **then …**  **else begin**  **if** RadioButton2.Checked **then …**  **else**  **if** RadioButton3.Checked **then** …  **end**  **C:\Documents and Settings\Admin\Рабочий стол\для рисунков к решениям 2-го тура\16-2.png** |