Министерство образования и науки

Московский технологический университет (МИРЭА)

|  |  |
| --- | --- |
| Дисциплина: | ***Микроэкономика*** |

Курсовая работа

**Основы алгоритмизации и программирования**

**Оглавление**

[Задача 1 3](#_Toc514839032)

[Задача 2 6](#_Toc514839033)

[Задача 3 9](#_Toc514839034)

[Задача 4 12](#_Toc514839035)

[Задача 5 15](#_Toc514839036)

[Задача 6 18](#_Toc514839037)

[Задача 7 24](#_Toc514839038)

[Список использованной литературы 26](#_Toc514839039)

# Задача 1

Смешали *V1* литров воды с температурой *t*1 градусов Цельсия с *V*2 литрами воды с температурой *t2* градусов Цельсия. Вычислите объем и температуру образовавшейся смеси.







unit Unit1;

interface

uses

 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

 TForm1 = class(TForm)

 Button1: TButton;

 Label1: TLabel;

 Edit1: TEdit;

 Label2: TLabel;

 Edit2: TEdit;

 Label3: TLabel;

 Label4: TLabel;

 Edit3: TEdit;

 Edit4: TEdit;

 Label5: TLabel;

 Label6: TLabel;

 Edit5: TEdit;

 Edit6: TEdit;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

 var

 v1, v2, t1, t2, v, t:real;

begin

 v1:=StrToCurr(Edit1.Text); //ввод v1

 v2:=StrToCurr(Edit3.Text); //ввод v2

 t1:=StrToCurr(Edit2.Text); //ввод t1

 t2:=StrToCurr(Edit4.Text); //ввод t2

 v:=v1+v2; //вычисление общего объема

t:=(v1\*t1+v2\*t2)/v; //вычисление температуры

Edit5.Text:=CurrToStr(v); //вывод итогового объема

 Edit6.Text:=CurrToStr(t); //вывод итоговой температуры

end;

end.

# Задача 2

В заданном массиве *A(N)* определите число соседств:

а) двух положительных чисел;

б) двух чисел разного знака;

в) двух чисел одного знака, причем абсолютная величина первого числа должна быть больше второго числа;

г) чётного числа и нечётного c нечётным индексом.





unit Unit1;

interface

uses

 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

 TForm1 = class(TForm)

 Button1: TButton;

 Label1: TLabel;

 Edit1: TEdit;

 Edit2: TEdit;

 Edit3: TEdit;

 Edit4: TEdit;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Label5: TLabel;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var

n, i:integer;

a: array [1..100] of integer;

b,c,d,e:integer;

begin

N:= 10; //число элементов

for i:=1 to n do //цикл по числу элементов

begin

 A[i]:= 20-random(10\*3); //генерируем элемент

 Memo1.Lines.Add(FloatToStrF(A[i],ffGeneral,4,4)); //выводим его на экран

end;

//обнуляем все переменные

b := 0;

c := 0;

d := 0;

e := 0;

for i:=1 to n do //просматриваем все элементы массива

begin

 if ((A[i-1] > 0)and(A[i] > 0)) then b:= b +1; //проверка двух положительных

 if ((A[i-1] > 0)and (A[i] < 0)or(A[i-1] < 0)and(A[i] > 0)) then c:= c + 1; //проверка двух чисел с разными знаками

 if ((A[i-1]\*A[i] > 0)and(abs(A[i-1]) > A[i])) then d:= d + 1; //числа с одним знаком, при этом модуль первого больше второго

 if (((A[i-1]mod 2) = 0)and((A[i] mod 2) = 1)and(i mod 2= 1))or((A[i-1]mod 2) = 1)and((A[i]mod 2) = 0)and((i-1 mod 2 = 1)) then e:= e + 1; //четное с нечетным, с учетом нечетного индекса

end ;

//вывод результатов

 Edit1.Text:=IntToStr(b);

 Edit2.Text:=IntToStr(c);

 Edit3.Text:=IntToStr(d);

 Edit4.Text:=IntToStr(e);

end;

end.

# Задача 3

Определите, имеется ли в заданном целочисленном массиве *X(N)* число, кратное заданным числам *А* и *В,* и не кратное числу С.







unit Unit1;

interface

uses

 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

 TForm1 = class(TForm)

 Button1: TButton;

 Edit1: TEdit;

 Edit2: TEdit;

 Edit3: TEdit;

 Edit4: TEdit;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Label5: TLabel;

 Memo1: TMemo;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var

n, i, k, h:integer;

r: array [1..100] of integer;

A, B, C :integer;

begin

N:= 10; //число элементов

for i:=1 to n do //цикл по числу элементов

begin

 r[i]:= 20-random(10\*3); //генерируем очередной элемент

 Memo1.Lines.Add(FloatToStrF(r[i],ffGeneral,4,4)); //выводим его на экран

end;

 A:= StrToInt(Edit1.Text); //ввод а

B := StrToInt(Edit2.Text); //ввод b

C:= StrToInt(Edit3.Text); //ввод с

 k:=0;

 h:=0;

 i:=1;

 while((k = 0) and (i<=N)) do //пока не нашли число или не достигли конца массива

 begin

 if ((r[i] mod A= 0) and (r[i] mod B= 0) and (r[i] mod C<>0)) then //проверка кратности

 begin

 k:= 1;

 h:=r[i];

 end

 else i:=i+1;

 end;

 Edit4.Text:=IntToStr(h); //вывод найденного числа

end;

end.

# Задача 4

Для заданной матрицы *A(N, N)* найдите сумму элементов, расположенных в строках с отрицательным элементом на главной диагонали.





unit Unit1;

interface

uses

 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

 TForm1 = class(TForm)

 Button1: TButton;

 Edit4: TEdit;

 Label5: TLabel;

 Memo1: TMemo;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var

n, i, j:integer;

r: array [1..100,1..100] of integer;

s :integer;

begin

N:= 10; //размерность матрицы

for i:=1 to n do //цикл по строкам

begin

for j := 1 to n do //цикл по столбцам

begin

 r[i,j]:= 20-random(10\*3); //генерируем элемент

 Memo1.Lines.Add(FloatToStrF(r[i,j],ffGeneral,4,4)); //выводим его на экран

end;

end;

 s:=0; //обнуляем сумму

 for i:=1 to n do //цикл по числу строк

 begin

 if(r[i,i]<0) then begin //если элемент на главной диагонали < 0

 for j:=1 to n do //запускаем цикл по столбцам

 s:=s+r[i,j]; //увеличиваем сумму

 end;

 end;

 Edit4.Text:=IntToStr(s); //выводим результат

end;

end.

# Задача 5

Найдите максимальное из чисел, встречающихся в заданной матрице более одного раза.









unit Unit1;

interface

uses

 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

 TForm1 = class(TForm)

 Button1: TButton;

 Memo1: TMemo;

 Label1: TLabel;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

implementation

{$R \*.dfm}

 const n = 2; //размер матрицы

var a: array [0..n, 0..n] of integer;

 i, j: integer;

 s: string;

function max(list: tstringlist): string; //функция сортировки пузырьком по убыванию

begin

 for i:=0 to list.Count-2 do

 for j:=i+1 to list.Count-1 do

 if strtoint(list.Strings[i])<strtoint(list.Strings[j]) then

 begin

 s := list.Strings[i];

 list.Strings[i] := list.Strings[j];

 list.Strings[j] := s;

 end;

 result := list.Strings[0];

end;

procedure TForm1.Button1Click(Sender: TObject);

 var list: tstringlist;

begin

 list := tstringlist.Create;

 randomize;

 memo1.lines.clear; //очистка вывода

 for i:=0 to n do //цикл по строкам

 begin

 for j:=0 to n do //цикл по столбцам

 begin

 a[i, j] := random(10); //генерируем элемент

 Memo1.Lines.Add(FloatToStrF(a[i,j],ffGeneral,4,4)); //выводим его на экран

 end;

 end;

 for i:=0 to n do //цикл по строкам

 for j:=0 to n do //цикл по столбцам

 if pos(inttostr(a[i, j]), s)>0 then //если число уже присутствует в строке поиска

 list.Add(inttostr(a[i, j])) //добавляем символ в список

 else s := s + inttostr(a[i, j]) + ' '; //иначе добавляем его в строку поиска

 label1.Caption := max(list); //сортируем список и ищем в нем максимальное число

end;

end.

# Задача 6

В заданном множестве точек на плоскости найдите четыре точки, которые могут служить вершинами ромба.

Ввести точки, затем нажать кнопку.







unit Unit1;

interface

uses

 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

 TForm1 = class(TForm)

 Button1: TButton;

 Edit1: TEdit;

 Edit2: TEdit;

 Edit3: TEdit;

 Edit4: TEdit;

 Edit5: TEdit;

 Edit6: TEdit;

 Edit7: TEdit;

 Edit8: TEdit;

 Edit9: TEdit;

 Edit10: TEdit;

 Edit11: TEdit;

 Edit12: TEdit;

 Label1: TLabel;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Label5: TLabel;

 Label6: TLabel;

 Label7: TLabel;

 Label8: TLabel;

 Edit13: TEdit;

 Label9: TLabel;

 Label10: TLabel;

 Edit14: TEdit;

 Edit15: TEdit;

 Edit16: TEdit;

 Edit17: TEdit;

 Edit18: TEdit;

 Edit19: TEdit;

 Edit20: TEdit;

 Label11: TLabel;

 Label12: TLabel;

 Edit21: TEdit;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

implementation

{$R \*.dfm}

 const nmax=20; //макс. кол.точек

type Point=record //тип - точка

 x,y:integer;

 end;

//является ли ромбом

function Square(a,b,c,d:Point):boolean; //проверка ромба

var s1,s2,s3,s4,s5,s6:Longint;

begin

s1:=sqr(a.x-b.x)+sqr(a.y-b.y); //длина ab

s2:=sqr(a.x-c.x)+sqr(a.y-c.y); //длина ac

s3:=sqr(a.x-d.x)+sqr(a.y-d.y); //длина ad

s4:=sqr(b.x-c.x)+sqr(b.y-c.y); //длина bc

s5:=sqr(b.x-d.x)+sqr(b.y-d.y); //длина bd

s6:=sqr(c.x-d.x)+sqr(c.y-d.y); //длина cd

Square:=((s1=s3)and(s1=s4)and(s1=s6)and(s2=s5)and(s2=2\*s1))

 or ((s1=s2)and(s1=s5)and(s1=s6)and(s3=s4)and(s3=2\*s1))

 or ((s2=s3)and(s2=s4)and(s2=s5)and(s1=s6)and(s1=2\*s2)); //проверка

end;

procedure TForm1.Button1Click(Sender: TObject);

var t:array[1..20] of Point;///массив точек

 n,i,j,k,l,p,q:byte;

 f:boolean;

begin

 n:=6; //число точек

//ввод исходных данных

t[1].x:=strtoint(edit1.text);

t[1].y:=strtoint(edit2.text);

 t[2].x:=strtoint(edit3.text);

t[2].y:=strtoint(edit4.text);

 t[3].x:=strtoint(edit5.text);

t[3].y:=strtoint(edit6.text);

 t[4].x:=strtoint(edit7.text);

t[4].y:=strtoint(edit8.text);

 t[5].x:=strtoint(edit9.text);

t[5].y:=strtoint(edit10.text);

 t[6].x:=strtoint(edit11.text);

t[6].y:=strtoint(edit12.text);

//выбираем по 4 разных точки

f:=false;//нет квадрата

i:=1;

//перебираем точки

while(i<=n-3)and not f do

 begin

 j:=i+1;

 while(j<=n-2)and not f do

 begin

 k:=j+1;

 while(k<=n-1)and not f do

 begin

 l:=k+1;

 while(l<=n)and not f do

 begin

 if Square(t[i],t[j],t[k],t[l])then //если условие выполнено

 begin

//выводим точки

 edit21.text:='есть';

 edit13.Text:=inttostr(t[i].x);

 edit14.Text:=inttostr(t[i].y);

 edit15.Text:=inttostr(t[j].x);

 edit16.Text:=inttostr(t[j].y);

 edit17.Text:=inttostr(t[k].x);

 edit18.Text:=inttostr(t[k].y);

 edit19.Text:=inttostr(t[l].x);

 edit20.Text:=inttostr(t[l].y);

 f:=true;

 end

 else inc(l);

 end;

 if not f then inc(k);

 end;

 if not f then inc(j);

 end;

 if not f then inc(i);

 end;

if not f then edit21.text:='нет';;

end;

end.

# Задача 7

Выясните, верно ли, что в заданном предложении есть пара соседствующих одинаковых символов.





unit Unit1;

interface

uses

 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

 TForm1 = class(TForm)

 Button1: TButton;

 Label1: TLabel;

 Edit1: TEdit;

 Edit2: TEdit;

 Label2: TLabel;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

 VAR

 st: string;

 i: byte;

 fl: boolean;

BEGIN

 st:=Edit1.Text; //ввод текста

 For i:=1 to Length(st)-1 do //просмотр всех символов

 If st[i]=st[i+1] then fl:=true; //если соседние совпадают

 if fl=true then edit2.Text:= 'Da' else edit2.Text:= 'Net'; //вывод результата

end;

end.

# Список использованной литературы

1. Архангельский, А.Я. Язык Pascal и основы программирования в Delphi / А.Я. Архангельский. - М.: Бином-Пресс, 2008. - 496 c.
2. Гавриков, М.М. Теоретические основы разработки и реализации языков программирования: Учебное пособие / М.М. Гавриков, А.Н. Иванченко, Д.В. Гринченков. - М.: КноРус, 2010. - 184 c.
3. Емельянов, В.И. Основы программирования на Delphi. / В.И. Емельянов. - М.: Высшая школа, 2005. - 231 c.
4. Желонкин, А. Основы программирования в интегрированной среде Delphi / А. Желонкин. - М.: Бином. Лаборатория знаний, 2004. - 236 c.
5. Кундиус, В.А. Теоретические основы разработки и реализации языков программирования / В.А. Кундиус. - М.: КноРус, 2013. - 184 c.
6. Черпаков, И.В. Основы программирования: Учебник и практикум для прикладного бакалавриата / И.В. Черпаков. - Люберцы: Юрайт, 2016. - 219 c.