Содержание

1 Задача 1…………………………………………………………………….…….…3

* 1. Текст задачи…………………………………………………………….…..…....3
	2. Описание алгоритма………………………………………………………...…...3
	3. Код программы…………………………………………………………….…….3
	4. Результат работы программы…………………………………………….……..4
1. Задача 2……………………………………………………………………….......5
	1. Текст задачи……………………………………………………………….……...5
	2. Описание алгоритма……………………………………………………………..6
	3. Код программы……………………………………………………….…………..7
	4. Результат работы программы…………………………………………………....7
2. Задача 3…………………………………………………………………………...8
	1. Текст задачи………………………………………………………………………8
	2. Описание алгоритма……………………………………………………….…….8
	3. Код программы…………………………………………………………………..9
	4. Результат работы программы…………………………………………………..11

Библиографический список…………………………………………………..……12

1 Задача 1

* 1. Текст Задачи

 Даны 4 точки, заданные координатами. Определить является ли данная фигура трапецией.

* 1. Описание алгоритма

 Фигура является трапецией, если две стороны ее параллельны, а две другие нет. Параллельность сторон определяется вычислением углового коэффициента по двум точкам, то есть стороны параллельны, если угловые коэффициента равны. Определение параллельности двух сторон вынесено в отдельную функцию принимающую точки исследуемых сторон и возвращающей значение истина если стороны параллельны и ложь в противном случае. В одноосновной программе, если две противоположные стороны параллельны, а две другие нет, то фигура является трапецией. Так как мы имеем 4 стороны, то рассматривается два варианта.

* 1. Код программы

Program Pr1;

uses crt;

Type point=record

x:real;

y:real;

end;

function IsParl(P1,P2,P3,P4:point):boolean;

begin

 IsParl:=(P2.y-P1.y)\*(P4.x-P3.x)=(P4.y-P3.y)\*(P2.x-P1.x);

end;

var

 P1,P2,P3,P4:point;

begin

clrscr;

writeln('Последовательно введите координаты точек:');

write('A: ');readln(P1.x,P1.y);

write('B: ');readln(P2.x,P2.y);

write('C: ');readln(P3.x,P3.y);

write('D: ');readln(P4.x,P4.y);

if (IsParl(P1,P2,P3,P4)and not IsParl(P2,P3,P1,P4))

 or (not IsParl(P1,P2,P3,P4) and IsParl(P2,P3,P1,P4))

 then write('Трапеция')

else

 write('Не трапеция ');

 readln;

end.

* 1. Результат работы программы

 Результат работы программы представлен на рисунке 1.





Рис.1 Результат работы программы.

1. Задача 2

2.1 Текст задачи

Найти сумму ряда:

$$4\sum\_{n=1}^{∝}\frac{(-1)^{n+1}}{2n-1}.$$

2.2 Описание алгоритма

 В цикле вычисляется значение члена ряда и плюсуется к сумме. Цикл продолжается до тех пор, пока значение члена ряда не будет меньше точности. Очевидно, что наиболее подходящий для данной задачи цикл это цикл с постусловием, так как ни значение члена ряда, ни необходимое число шагов цикла заранее не известны.

2.3 Код программы

Program Pr2;

uses crt;

var

i:integer;

sum,eps,r:real;

begin

 clrscr;

 writeln('Введите точность:');

 readln(eps);

 sum:=0;

 i:=1;

 repeat

 if ((i+1) mod 2=0) then

 r:=4/(2\*i-1)

 else

 r:=-4/(2\*i-1);

 sum:=sum+r;

 i:=i+1;

 until(abs(r)<eps);

 writeln('Сумма ряда ', sum);

 sum:=0;

 i:=1;

 while(true) do

 begin

 if ((i+1) mod 2=0) then

 r:=4/(2\*i-1)

 else

 r:=-4/(2\*i-1);

 sum:=sum+r;

 i:=i+1;

 if abs(r)<eps then break;

 end;

 writeln('Сумма ряда ', sum);

 sum:=0;

 for i:=1 to i+1 do

 begin

 if ((i+1) mod 2=0) then

 r:=4/(2\*i-1)

 else

 r:=-4/(2\*i-1);

 sum:=sum+r;

 if abs(r)<eps then break;

 end;

 writeln('Сумма ряда ', sum);

 readln;

end.

2.4 Результат работы программы

Результат работы программы представлен на рисунке 2.



Рис.2 Результат работы программы.

1. Задача 3
	1. Текст задачи

 Одномерный массив размерностью 2n, где n — целое положительное число, заполнен случайными положительными целыми числами. Изменить массив по следующему правилу:

в первую ячейку поместить модуль разности между значениями первой и второй ячеек, во вторую — модуль разности между значениями второй и третьей ячеек, … в последнюю — модуль разности между значениями последней и первой ячеек. Определить, сколько понадобится таких преобразований, прежде чем массив будет состоять из одних нулей.

* 1. Описание алгоритма

 Создаем массив из случайных положительных чисел, по алгоритму указанному в задании преобразуем массив до тех пор, пока все элементы массива не станут равными нулю. При этом считаем количество таких преобразований. Преобразование массива и проверка того что все элементы равны 0 вынесены в отдельную процедуру и функцию соответственно.

* 1. Код программы

program pr3;

{$R-}

uses crt;

type

 TArrayT=array[1..1] of integer;

 PArrayT=^TArrayT;

function chek(M:PArrayT; n:integer):boolean;

var

i:integer;

begin

 for i:=1 to n do

 if (M^[i]<>0) then

 begin

 chek:=false;

 exit;

 end;

 chek:=true;

end;

procedure convert(M:PArrayT; n:integer);

var

i,temp:integer;

begin

 temp:=M^[1];

 for i:=1 to n-1 do

 begin

 M^[i]:=abs(M^[i]-M^[i+1]);

 end;

 M^[n]:=abs(M^[n]-temp);

end;

procedure print(M:PArrayT; n:integer);

var

i:integer;

begin

 for i:=1 to n do

 begin

 write(M^[i]);

 write(' ');

 end;

 writeln('');

end;

 var

 M:PArrayT;

 n,i,size:integer;

 k:longint;

 begin

 clrscr;

 randomize;

 writeln('Введите n:');

 readln(n);

 size:=round(exp(n\*ln(2)));

 getmem(M,SizeOf(integer)\*(size+1));

 for i:=1 to size do

 begin

 M^[i]:=random(100);

 end;

 print(M,size);

 k:=0;

 while(not chek(M,size)) do

 begin

 convert(M,size);

 print(M,size);

 k:=k+1;

 end;

 writeln('Число преобразований ',k);

 readln;

 end.

* 1. Результат работы программы

Результат работы программы представлен на рисунке 2.



Рис.3 Результат работы программы.

Библиографический список

1. Острейковский В. А. Информатика : учебник для вузов / В. А. Острейковский. — М. : Высш. шк., 1999. 2. Вирт Н. Алгоритмы и структуры данных / Н. Вирт. — М. : Мир, 1989. 3. Епанешников А. М. Программирование в среде Turbo Pascal 7.0 / А. М. Епанешников, В. А. Епанешников. — 3-е изд., стер. — М. : ДИАЛОГ-МИФИ, 1996. 4. Культин Н .Б. Turbo Pascal в задачах и примерах / Н. Б. Культин. — СПб. : БХВ-Петербург, 2000. 5. Немнюгин С. А. Turbo Pascal / С. А. Немнюгин. — СПб. : Питер, 2000. 6. Шпак Ю. А. Turbo Pascal 7.0 на примерах / Ю. А. Шпак ; под ред. Ю. С. Ковтанюка. — Киев : Юниор, 2003.