

**Pseudocódigo**

//Inicio

// Variables

i, j, x=0, pas[50] : enteros

//proceso

Desde i=1, Hasta i=6, i=i+1

Desde j=x, Hasta j≥0, j=j+1

si (j==0 o j==0)

pas[j]=0

Sino

pas[j]=pas[j]+pas[j+1]

x=x+1

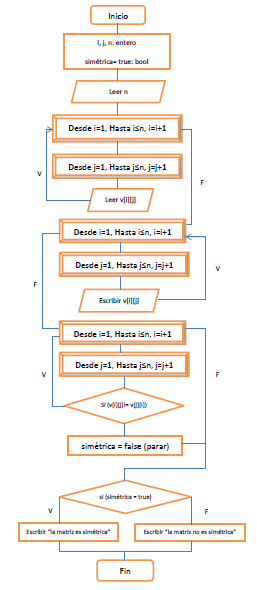
Desde j=1, Hasta j≤6-i, j=j+1

Desde j=0, Hasta j<x, j=j+1

//salida

escribir pas[j]

Fin

****

**Pseudocódigo**

// Inicio

// variables

i, j, n: entero

simétrica= true: bool

//entrada

Leer n

. Desde i=1, Hasta i≤n, i=i+1

Desde j=1, Hasta j≤n, j=j+1

Leer v[i][j]

//proceso

Desde i=1, Hasta i≤n, i=i+1

Desde j=1, Hasta j≤n, j=j+1

Escribir v[i][j]

Desde i=1, Hasta i≤n, i=i+1

Desde j=1, Hasta j≤n, j=j+1

Si (v[i][j]!= v[j][i])

simétrica = false (parar)

si (simétrica = true)

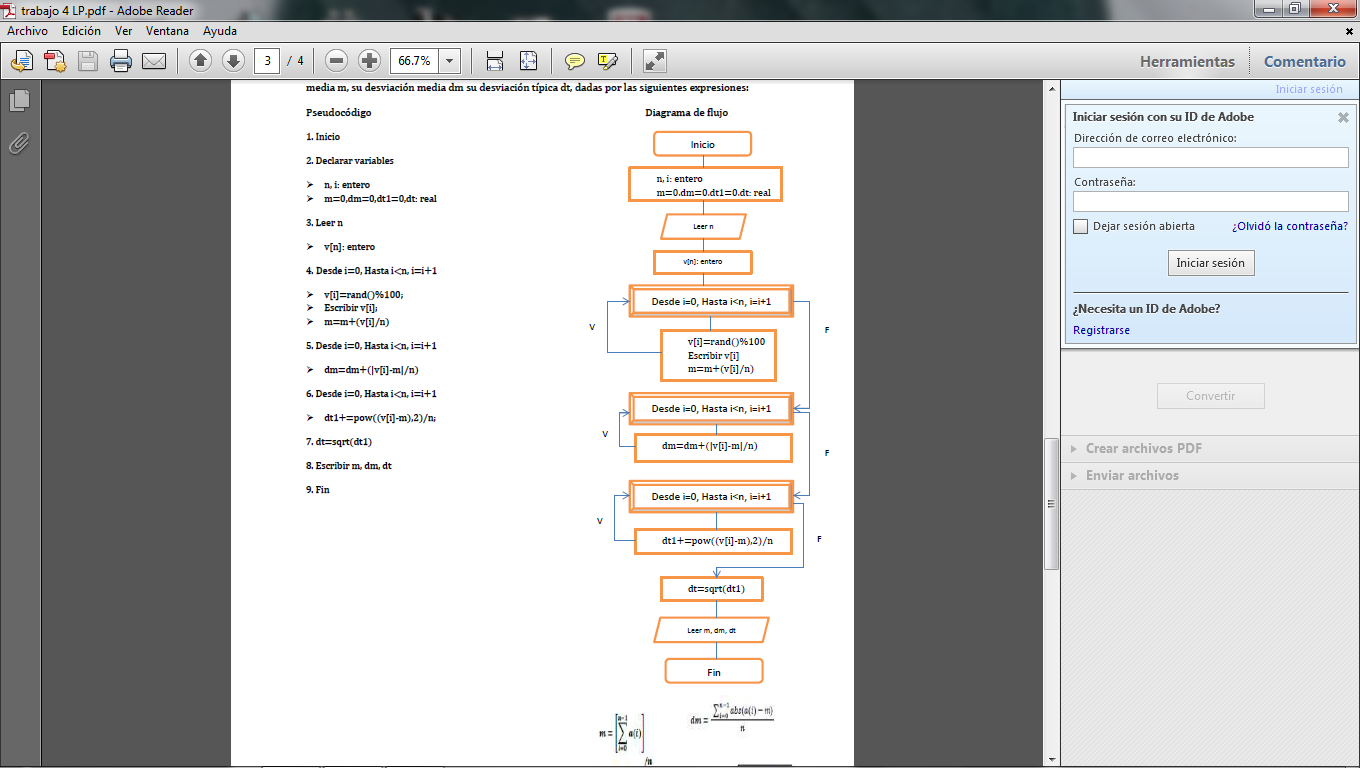
//salida

Escribir “la matriz es simétrica”

Sino

Escribir “la matriz no es simétrica”

Fin

Pseudocódigo

// Inicio

// Declarar variables

n, i: entero

m=0,dm=0,dt1=0,dt: real

//entrada

Leer n

v[n]: entero

//proceso

Desde i=0, Hasta i<n, i=i+1

v[i]=rand()%100;

Escribir v[i];

m=m+(v[i]/n)

Desde i=0, Hasta i<n, i=i+1

dm=dm+(|v[i]-m|/n)

Desde i=0, Hasta i<n, i=i+1

dt1+=pow((v[i]-m),2)/n;

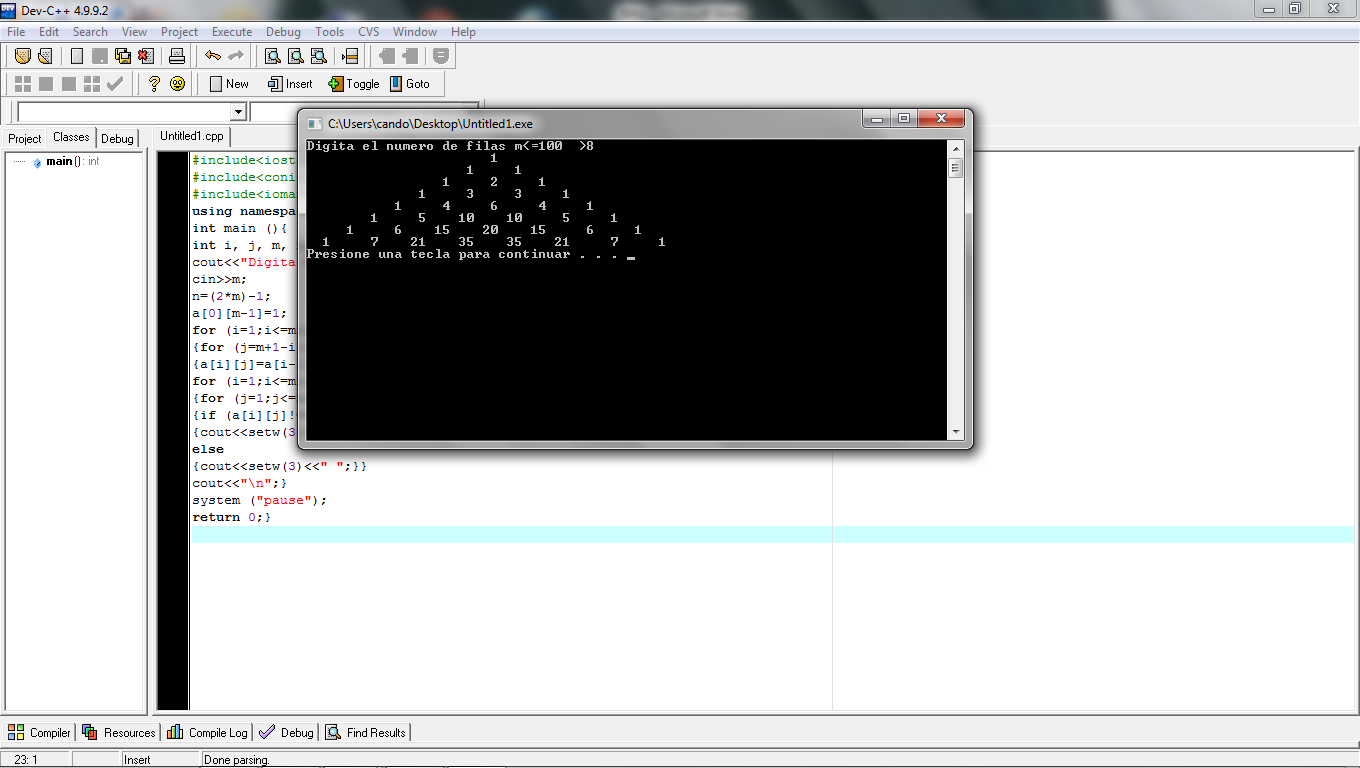
dt=sqrt(dt1)

//salida

Escribir m, dm, dt

Fin

**1 Escribir una función que permita visualizar el triángulo de pascal:**

***Codification del triangle Pascal***

#include<iostream>

#include<conio.h>

#include<iomanip>

using namespace std;

int main (){

int i, j, m, n, a[100][100];

cout<<"Digita el numero de filas m<=100 >";

cin>>m;

n=(2\*m)-1;

a[0][m-1]=1;

for (i=1;i<=m;i++)

{for (j=m+1-i;j<=i-m+n;j++)

{a[i][j]=a[i-1][j-1]+a[i-1][j+1];}}

for (i=1;i<=m;i++)

{for (j=1;j<=n;j++)

{if (a[i][j]!=0)

{cout<<setw(3)<<a[i][j];}

else

{cout<<setw(3)<<" ";}}

cout<<"\n";}

system ("pause");

return 0;}

**3. Escribir un programa que genere aleatoriamente los datos de un vector, lo visualice, y calcule su media m, su desviación media dm su desviación típica dt.**

#include<iostream>

#include <cstdlib>

#include<iomanip>

#include<cmath>

using namespace std;

int main(){

int n;

double m=0,dm=0,dt;

cout<<"ingrese cantidad de numeros del vector\n";cin>>n;

cout<<endl;

int v[n];

for(int i=0;i<n;i++){

v[i]=rand()%10+1;

cout<<setw(4)<<v[i];

m+=v[i]/(n\*1.0);

}

cout<<endl;

for(int i=0;i<n;i++)

dm+=sqrt(pow((v[i]-m),2))/n;

dt=sqrt(dm);

cout<<"\n LA MEDIA ES: : "<<m<<endl;

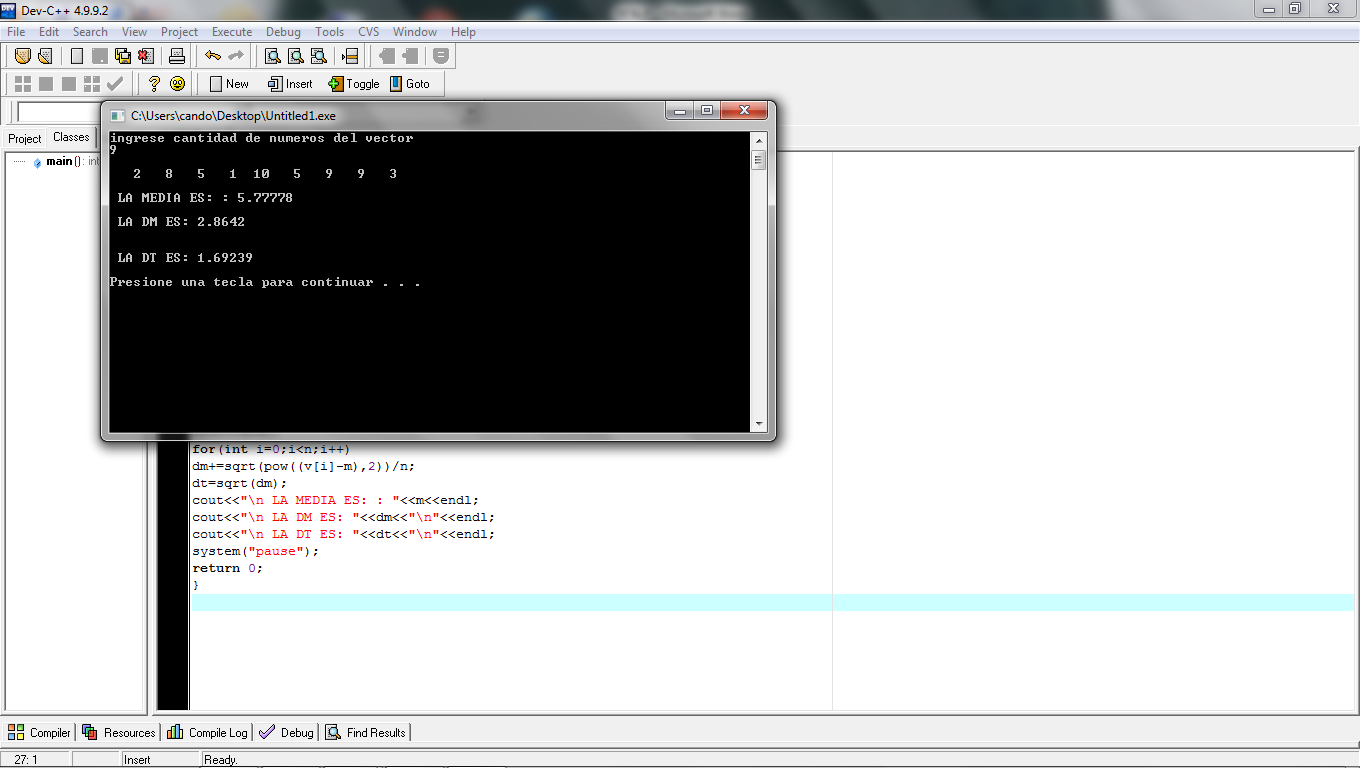
cout<<"\n LA DM ES: "<<dm<<"\n"<<endl;

cout<<"\n LA DT ES: "<<dt<<"\n"<<endl;

system("pause");

return 0;

}



2. Escribir una función que reciba como parámetros una matriz cuadrada de orden **n** y decida si es simétrica. Una matriz cuadrada de orden **n** es simétrica si **a [i][j] == a [j][i]** para todos los valores de los índices i, j.

#include<iostream>

#include<cmath>

#include<iomanip>

using namespace std;

int main(){

int fil,col,orden;

int A[20][20];

cout<<"ingrese el orden de la matriz:";cin>>orden;

void simetrico(int[][20],int dimen);

for(fil=0;fil<orden;fil++)

for(col=0;col<orden;col++)

{cout<<"A["<<setw(2)<<fil+1<<"]["<<setw(2)<<col+1<<"]:";

cin>>A[fil][col]; } simetrico(A,orden);

return 0;} void simetrico(int matriz[][20],int dimen)

{#define true 1

#define false 0

int fil,col,ok;

ok = true;

cout<<"matriz ingresada:"<<endl<<endl;

for(col=0;fil<dimen;col++)

{cout<<endl;

for(col=0;col<dimen;col++)

{cout<<setw(4)<<matriz[fil][col];

if(matriz[fil][col]!=matriz[col][fil])

ok=false; }}

cout<<endl<<endl;

if(ok)

cout<<"si es simetrico"<<endl;

else

cout<<"no es simetrico"<<endl;

}

