APLICACIONES DEL CAPÍTULO IV.

4.1. Elaborar un programa para determinar las raíces de una ecuación cuadrática de la forma $Ax^2+Bx+C=0$ para el cual se utiliza las siguientes fórmulas:

$$x_1 = \frac{-B+\sqrt{B^2-4AC}}{2A}$$

$$x_2 = \frac{-B-\sqrt{B^2-4AC}}{2A}$$

El programa en MATLAB sería:

```
% raíces de una ecuación de la forma Ax2+Bx+C=0
clc;
A=input('Ingrese A: ');
B=input('Ingrese B: ');
C=input('Ingrese C: ');
x1=(-B+sqrt(B^2-4*A*C))/(2*A);
x2=(-B-sqrt(B^2-4*A*C))/(2*A);
fprintf('x1 =');disp(x1);
fprintf('x2 =');disp(x2);
```

Cuando se ejecuta el programa para valores de A=1, B=5 y C=-20 se obtiene lo siguiente:

```
Ingrese A: 1
Ingrese B: 5
Ingrese C: -20
x1 = 2.6235
x2 = -7.6235
```

4.2. Diseñar una función que calcule las raíces de una ecuación cuadrática de la forma $Ax^2+Bx+C=0$, los valores de A, B y C se deben ingresar como parámetros de la función.

```
function [x1,x2]=raicesEC(A,B,C)
      % Función que calcula las raíces de una
      % ecuación de la forma Ax2+Bx+C=0
      x1=(-B+sqrt(B^2-4*A*C))/(2*A);
      x2=(-B-sqrt(B^2-4*A*C))/(2*A);
end
```

Cuando se ejecuta la función para valores de A=3, B=2 y C=-6 se obtiene lo siguiente:

```
>> [r1,r2]=raicesEC(3,2,-6)
r1 =
  1.1196

r2 =
  -1.7863
```

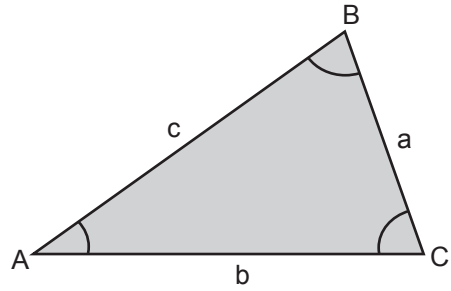
Cuando se ejecuta la función para valores de A=1, B=2 y C=20 se obtiene lo siguiente:

```
>> [r1,r2]=raicesEC(1,2,20)
r1 =
  -1.0000 + 4.3589i

r2 =
  -1.0000 - 4.3589i
```

Lo que significa que la función raicesEC calcula las raíces tanto con números reales como con números complejos.

4.3. Elaborar un programa para determinar el área del triángulo cuyos lados se deben ingresar desde el teclado; tener en cuenta que si los lados ingresados no corresponden a un triángulo debe salir el mensaje “Los lados no corresponden a un triángulo”.




```
% programa para calcular el área de un triángulo.
clc
a=input('Ingresa lado a: ');
b=input('Ingresa lado b: ');
c=input('Ingresa lado c: ');
if (a+b>c & a+c>b & b+c>a)
```

```
        p=(a+b+c)/2;
        area=sqrt(p*(p-a)*(p-b)*(p-c));
        fprintf('área=%1.2f\n',area);
    else
        fprintf('Los lados no corresponden a un triángulo');
    end
```

Cuando se ejecuta el programa para valores de a=3, b=4 y c=5 se obtiene lo siguiente:

```
Ingresa lado a: 3
Ingresa lado b: 4
Ingresa lado c: 5
área=6.00
```

Cuando se ejecuta el programa para valores de a=2, b=2 y c=100 se obtiene lo siguiente:

```
Ingresa lado a: 2
Ingresa lado b: 2
Ingresa lado c: 100
Los lados no corresponden a un triángulo
```

4.4. Diseñar una función que devuelva el área de un triángulo cuyos lados se deben ingresar como parámetros de la función; tener en cuenta que si los lados ingresados no corresponden a un triángulo debe salir el mensaje “Error: Los lados no corresponden a un triángulo”.

```
function area=areatriangulo(a,b,c)
    % Función que calcula el área de un triángulo
    % ingresando los lados a,b,c.
    if (a+b>c & a+c>b & b+c>a)
        p=(a+b+c)/2;
        area=sqrt(p*(p-a)*(p-b)*(p-c));
    else
        disp('Error: Los lados no corresponden a un triángulo');
    end
end
```

Cuando se ejecuta la función para valores de a=3, b=4 y c=5 se obtiene lo siguiente:

```
>> ar=areatriangulo(3,4,5)
```

```
ar =
  6
```

Cuando se ejecuta el programa para valores de a=2, b=2 y c=100 se obtiene lo siguiente:

```
>> areatriangulo(2,2,100)
Error: Los lados no corresponden a un triángulo
```

4.5. Diseñar una función que devuelva el factorial doble de un número "n" que se debe ingresar como parámetro de la función; tener en cuenta que si "n" es negativo debe salir el mensaje "Error: No se puede calcular el factorial doble de un número negativo", las condiciones para determinar el factorial doble son:

- Si "n" es par, entonces se multiplican los pares hasta 2, es decir: 8!! = 8.6.4.2
- Si "n" es impar, entonces se multiplican los pares hasta 1, es decir: 7!! = 7.5.3.1
- Si "n" es cero o uno el resultado es uno, es decir: 0!! = 1 y 1!! = 1.

```
function y=factorialdoble(n)
      % Función que calcula el factorial doble de n.
      if (n>=0)
            y=1;
            if rem(n,2)==0
                  k=2;
            else
                  k=1;
            end
            for (i=k:2:n)
                  y=y*i;
            end
      else
            disp('Error: el parámetro no puede ser negativo');
      end
end
```

Cuando se ejecuta la función para valor de n=8 se obtiene lo siguiente:

```
>> a=factorialdoble(8)
a =
  384
```

Cuando se ejecuta la función para valor de n=0 se obtiene lo siguiente:

```
>> a=factorialdoble(0)
a =
  1
```

Cuando se ejecuta la función para valor de n=-7 se obtiene lo siguiente:

```
>> a=factorialdoble(-7)
Error: el parámetro no puede ser negativo
```

4.6. Diseñar una función para realizar el cambio de base 10 a otra base especificada menor igual a 9.

```
function n=cambiobaseN10(num,base)
      % Función que cambia número de base 10 a base menor igual a 9
      inv=0;
      while (num>0)
            inv=10*inv+rem(num,base);
            num=floor(num/base);
      end
      n=0;
      while (inv>0)
            n=10*n+rem(inv,10);
            inv=floor(inv/10);
      end
end
```

Cuando se ejecuta la función para valor de num=1834 en base 5 se obtiene lo siguiente:

```
>> c=cambiobaseN10(1834,5)
c =
  24314
```

Cuando se ejecuta la función para valor de num=27 en base 2 se obtiene lo siguiente:

```
>> c=cambiobaseN10(27,2)
c =
  11011
```

4.7. Diseñar una función para redondear un número “n” a “d” cifras decimales.

```
function y=redondear(n,d)
      % Función que redondea un número "n" a "d" decimales
```

```
        n1=floor(n*10^d);
        n2=floor(n*10^(d+1));
        r=floor(rem(n2,10));
        if (r>=5)
              n1=n1+1;
        end
        y=n1/10^d;
return
```

Cuando se ejecuta la función para valor de n=3,2567 y d=2 (redondear a dos decimales) se obtiene lo siguiente:

```
>> redondear(3.2567,2)
ans =
  3.2600
```

Cuando se ejecuta la función para valor de n=134,32243 y d=3 (redondear a dos decimales) se obtiene lo siguiente:

```
>> redondear(134.32243,3)
ans =
  134.3220
```

4.8. Diseñar una función para invertir un número entero que se debe ingresar como parámetro de la función.

```
function inv=invertirnumero(n)
      % Función que invierte los dígitos un número "n"
      inv=0;
      while (n>0)
            inv=10*inv+rem(n,10);
            n=floor(n/10);
      end
return
```

Cuando se ejecuta la función para valor de n=15467 se obtiene lo siguiente:

```
>> invertirnumero(15467)
ans = 76451
```

4.9. Diseñar una función para determinar si un número entero que se ingresa como parámetro es capicúa (utilizar la función "***invertirnumero***" diseñado en el ejemplo

anterior). La función debe devolver uno (1) si es capicúa y cero (0) si no los es.

```
function x=capicua(n)
      % Función que verifica si "n" es capicúa o no
      or=n;
      inv=invertirnumero(n); % función diseñado ejemplo anterior
      if (or==inv)
            x=1;
      else
            x=0;
      end
return
```

Cuando se ejecuta la función para valor de n=345 se obtiene lo siguiente:

```
>> capicua(345)
ans =
   0
```

Cuando se ejecuta la función para valor de n=545 se obtiene lo siguiente:

```
>> capicua(545)
ans =
   1
```

4.10. Diseñar una función que devuelva la suma de todos los divisores de un número "n" que se debe ingresar como parámetro de la función.

```
function suma=sumadivisores(n)
      % Función que determinar la suma de los divisores de "n"
      suma=0;
      for (i=1:n)
            if (rem(n,i)==0)
                  suma=suma+i;
            end
      end
end
```

Cuando se ejecuta la función para valor de n=24 se obtiene lo siguiente:

```
>> sumadivisores(24)
```

```
ans =
  60
```

4.11. Diseñar una función que devuelva el valor de "E" de la siguiente serie, como parámetro de la función se debe ingresar "n". (7 puntos)

$$E = \frac{1}{2} - \frac{2}{3} + \frac{3}{4} - \frac{4}{5} + \frac{5}{6} - \frac{6}{7} + ... \pm \frac{n}{n+1}$$

```
function e=serieE(n)
      signo=1;
      e=0;
      for (i=1:n)
            e=e+signo*i/(i+1);
            signo=-signo; %Intercambia el signo entre + y -.
      end
end
```

Cuando se ejecuta la función para valor de n=4 se obtiene lo siguiente:

```
>> serieE(4)
ans =
  -0.2167
```

4.12. Elaborar un programa para ingresar desde el teclado los elementos de una lista o vector y luego calcular el valor máximo y mínimo sin utilizar la función ***max*** ni ***min*** del MATLAB.

```
%Programa para ingresar datos de un vector desde el teclado.
clc
n=input('Ingrese número de elementos: ');
for (i=1:n)
      fprintf('Ingr     ese dato (%1.0f): ',i);
      X(i)=input('');
end
ma=X(1);
me=X(1);
for (i=2:n)
      if (X(i)>ma)
            ma=X(i);
      end
```

```
        if (X(i)<me)
              me=X(i);
        end
end
fprintf('Valor máximo = %1.0f\n',ma);
fprintf('Valor mínimo = %1.0f\n',me);
```

Cuando se ejecuta el programa para n=6 e introducir los valores: 3, 12, 5, -2, 8 y 9 se obtiene lo siguiente:

```
Ingrese número de elementos: 6
Ingrese dato (1): 3
Ingrese dato (2): 12
Ingrese dato (3): 5
Ingrese dato (4): -2
Ingrese dato (5): 8
Ingrese dato (6): 9

Valor máximo = 12
Valor mínimo = -2
```

Este mismo programa, pero utilizando las funciones ***max*** y ***min*** del MATLAB sería más fácil de programar, y los resultados son los mismos, el código en MATLAB sería:

```
%Programa para ingresar datos de un vector desde el teclado.
clc
n=input('Ingrese número de elementos: ');
for (i=1:n)
      fprintf('Ingrese dato (%1.0f): ',i);
      X(i)=input('');
end
ma=max(X);
me=min(X);
fprintf('Valor máximo = %1.0f\n',ma);
fprintf('Valor mínimo = %1.0f\n',me);
```

4.13. Diseñar una función que permita ingresar desde el teclado los elementos de una lista o vector y los almacene en un array unidimensional.

```
function [X]=leervector(n)
      %Función para ingresar datos de un vector desde el teclado.
```

```
        for (i=1:n)
              fprintf('Ingrese dato (%1.0f): ',i);
              X(i)=input('');
        end
end
```

Cuando se ejecuta la función para n=6 e introducir los valores: 4, 7, 2, 8, 1 y 0 se obtiene lo siguiente:

```
>> A=leervector(6)
Ingrese dato (1): 4
Ingrese dato (2): 7
Ingrese dato (3): 2
Ingrese dato (4): 8
Ingrese dato (5): 1
Ingrese dato (6): 0

A =
  4 7 2 8 1 0
```

4.14. Elaborar el mismo programa del ejemplo 4.12, pero utilizando las funciones leervector(n) diseñado en el ejemplo 4.13 y las funciones ***max*** y ***min*** del MATLAB; observen que el código del programa es mucho más pequeño.

```
%Programa para ingresar datos de un vector desde el teclado.
clc
n=input('Ingrese número de elementos: ');
X=leervector(n);
ma=max(X);
me=min(X);
fprintf('Valor máximo = %1.0f\n',ma);
fprintf('Valor mínimo = %1.0f\n',me);
```

Cuando se ejecuta el programa para n=6 e introducir los valores: 3, 12, 5, -2, 8 y 9 se obtiene lo siguiente:

```
Ingrese número de elementos: 6
Ingrese dato (1): 3
Ingrese dato (2): 12
Ingrese dato (3): 5
Ingrese dato (4): -2
```

```
Ingrese dato (5): 8
Ingrese dato (6): 9
Valor máximo = 12
Valor mínimo = -2
```

4.15. Diseñar una función para invertir el orden de una array unidimensional, por ejemplo si el array es A=[5,2,8,9,1,2,0,6]; la función debe devolver el array B=[6,0,2,1,9,8,2,5] en orden invertido.

```
function [X]=invertirlista(Y)
      % Función que invierte el orden de los datos de Y.
      n=length(Y);
      q=n;
      for (i=1:n)
            X(i)=Y(q);
            q=q-1;
      end
end
```

Cuando se ejecuta la función se obtiene lo siguiente:

```
>> A=[5,2,8,9,1,2,0,6];
>> B=invertirlista(A)
B =
  6 0 2 1 9 8 2 5
```

4.16. Elaborar un programa para ingresar desde el teclado los elementos de una matriz y luego obtener la suma de cada columna (no usar la función ***sum*** del MATLAB) que se debe almacenar en un vector.

```
%Programa para ingresar datos de una matriz desde el teclado.
clear X;
clear Y;
clc
f=input('Ingrese número de filas: ');
c=input('Ingrese número de columnas: ');
for (i=1:f)
      fprintf('\nIngrese datos de la fila %1.0f\n',i);
      for (j=1:c)
            fprintf('Ingrese elemento(%1.0f,%1.0f): ',i,j);
            X(i,j)=input('');
```

```
        end
    end
    for (i=1:c)
        Y(i)=0;
        for (j=1:f)
            Y(i)=Y(i)+X(j,i);
        end
    end
    disp(X);
    fprintf('\nLa suma de columnas es:\n');
    disp(Y);
```

Cuando se ejecuta el programa para f=3 y c=4 e introducir los valores de la siguiente matriz:

```
1 2 3 4
5 2 3 2
1 4 2 3

Ingrese número de filas: 3
Ingrese número de columnas: 4

Ingrese datos de la fila 1
Ingrese elemento(1,1): 1
Ingrese elemento(1,2): 2
Ingrese elemento(1,3): 3
Ingrese elemento(1,4): 4

Ingrese datos de la fila 2
Ingrese elemento(2,1): 5
Ingrese elemento(2,2): 2
Ingrese elemento(2,3): 3
Ingrese elemento(2,4): 2

Ingrese datos de la fila 3
Ingrese elemento(3,1): 1
Ingrese elemento(3,2): 4
Ingrese elemento(3,3): 2
Ingrese elemento(3,4): 3


1 2 3 4
5 2 3 2
```

```
1 4 2 3

La suma de columnas es:
7 8 8 9
```

4.17. Diseñar una función que permita ingresar desde el teclado los elementos de una matriz y los almacene en un array bidimensional.

```
function [X]=leermatriz(f,c)
      % Función que lee desde el teclado una matriz de fxc.
      for (i=1:f)
            fprintf('\nIngrese datos de la fila %1.0f\n',i);
            for (j=1:c)
                  fprintf('Ingrese elemento(%1.0f,%1.0f): ',i,j);
                  X(i,j)=input('');
            end
      end
end
```

Cuando se ejecuta la función para f=3 y c=4 e introducir los valores de la siguiente matriz:

1 2 3 4
5 2 3 2
1 4 2 3

```
>> A=leermatriz(3,4)

Ingrese datos de la fila 1
Ingrese elemento(1,1): 1
Ingrese elemento(1,2): 2
Ingrese elemento(1,3): 3
Ingrese elemento(1,4): 4

Ingrese datos de la fila 2
Ingrese elemento(2,1): 5
Ingrese elemento(2,2): 2
Ingrese elemento(2,3): 3
Ingrese elemento(2,4): 2

Ingrese datos de la fila 3
Ingrese elemento(3,1): 1
Ingrese elemento(3,2): 4
```

```
Ingrese elemento(3,3): 2
Ingrese elemento(3,4): 3

A =
  1 2 3 4
  5 2 3 2
  1 4 2 3
```

4.18. Elaborar el mismo programa del ejemplo 4.16, pero utilizando las funciones ***leermatriz(f,c)*** diseñado en el ejemplo 4.17 y la función ***sum*** del MATLAB; observen que el código del programa es mucho más pequeño.

```
%Programa para ingresar datos de una matriz desde el teclado.
clear X;
clear Y;
clc
f=input('Ingrese número de filas: ');
c=input('Ingrese número de columnas: ');
X=leermatriz(f,c);
Y=sum(X);
disp(X);
fprintf('\nLa suma de columnas es:\n');
disp(Y);
```

Su aplicación se muestra a continuación:

```
Ingrese número de filas: 3
Ingrese número de columnas: 4

Ingrese datos de la fila 1
Ingrese elemento(1,1): 1
Ingrese elemento(1,2): 2
Ingrese elemento(1,3): 3
Ingrese elemento(1,4): 4

Ingrese datos de la fila 2
Ingrese elemento(2,1): 5
Ingrese elemento(2,2): 2
Ingrese elemento(2,3): 3
Ingrese elemento(2,4): 2
```

```
Ingrese datos de la fila 3
Ingrese elemento(3,1): 1
Ingrese elemento(3,2): 4
Ingrese elemento(3,3): 2
Ingrese elemento(3,4): 3

      1 2 3 4
      5 2 3 2
      1 4 2 3

La suma de columnas es:
      7 8 8 9
```

4.19. Diseñar una función para calcular el promedio de todos los elementos impares múltiplos de tres de una matriz de nxm.

```
function prom=promImpares3(A)
      % Función que devuelve el promedio de todos los elementos
      % impares múltiplos de 3 de una matriz A.
      [n,m]=size(A);
      k=0;
      suma=0;
      for (i=1:n)
            for (j=1:m)
                  if (rem(A(i,j),2)==1)&&(rem(A(i,j),3)==0)
                        suma=suma+A(i,j);
                        k=k+1;
                  end
            end
      end
      prom=suma/k;
end
```

Su aplicación se muestra a continuación:

```
>> A=[12,10,9;21,13,8;11,15,6]
A =
  12 10 9
  21 13 8
  11 15 6
```

```
>> promImpares3(A)
ans =
  15
```

Si observamos la matriz ingresada, los elementos impares múltiplos de 3 son: 9, 21 y 15 cuya suma es 45 por tanto el promedio sería 15.

4.20. Diseñar una función para generar una matriz de mxn con elementos consecutivos que empieza con el uno.

```
function [X]=matrizconsecutivos(f,c)
      % Función que genera una matriz de fxc con números consecut.
      k=1;
      for (i=1:f)
            for (j=1:c)
                  X(i,j)=k;
                  k=k+1;
            end
      end
end
```

Su aplicación se muestra a continuación:

```
>> matrizconsecutivos(3,4)
ans =
  1  2  3  4
  5  6  7  8
  9 10 11 12
```

4.21. Elaborar un programa para ingresar dos números, el programa debe tener un sistema de menú con las siguientes opciones:

1. Sumar.
2. Restar.
3. Multiplicar.
4. Dividir.

Entonces con los números ingresados y la opción seleccionada se debe realizar la operación correspondiente:

```
%Programa con opciones para hacer las 4 operaciones aritméticas.
clc
a=input('Ingrese dato1: ');
```

```
b=input('Ingrese dato2: ');
disp(' ');
disp('MENÚ DE OPCIONES');
disp('1. Sumar');
disp('2. Restar');
disp('3. Multiplicar');
disp('4. Dividir');
opcion=input('Elija una opción (1-4): ');
switch (opcion)
      case 1
            r=a+b;
      case 2
            r=a-b;
      case 3
            r=a*b;
      case 4
            r=a/b;
      otherwise
            disp('Opción equivocada');
end
if (opcion>=1 & opcion<=4)
      fprintf('\nResultado = %1.2f\n',r)
end
```

Cuando se ejecuta el programa para a=8 y b=5 y elegimos la opción 3, se tiene:

```
Ingrese dato1: 8
Ingrese dato2: 5
MENÚ DE OPCIONES
1. Sumar
2. Restar
3. Multiplicar .
4. Dividir
Elija una opción (1-4): 3
Resultado = 40
```

Otra forma de hacer el mismo programa sería usando el if-elseif:

```
%Programa con opciones para hacer las 4 operaciones aritméticas.
clc
```

```
a=input('Ingrese dato1: ');
b=input('Ingrese dato2: ');
disp(' ');
disp('MENÚ DE OPCIONES');
disp('1. Sumar');
disp('2. Restar');
disp('3. Multiplicar');
disp('4. Dividir');
opcion=input('Elija una opción (1-4): ');
 if (opcion==1)
      r=a+b;
elseif (opcion==2)
      r=a-b;
elseif (opcion==3)
      r=a*b;
elseif (opcion==4)
      r=/b;
else
      disp('Opción equivocada');
end
if (opcion>=1 & opcion<=4)
      fprintf('Resultado = %1.2f\n',r)
end
```

4.22. Elaborar un programa para ingresar las edades de "n" personas y para leer las "n" edades se debe utilizar la función leervector(n) diseñado en el ejemplo 4.13; el programa debe calcular la cantidad de personas que corresponden a cada una de las siguientes categorías:

- Niños de 1 a 12 años.
- Adolescentes de 13 a 17 años.
- Jóvenes de 18 a 35 años.
- Adultos de 35 a 60 años.
- Mayores de 60 a más años.

```
%Programa para seleccionar por edades.
```

```
clc
n=input('Ingrese número de personas: ');
edad=leervector(n);
N=0;
A=0;
J=0;
Ad=0;
M=0;
for (i=1:n)
      if (edad(i)>=1 && edad(i)<=12)
            N=N+1;
      elseif (edad(i)>=13 && edad(i)<=17)
            A=A+1;
      elseif (edad(i)>=18 && edad(i)<=35)
            J=J+1;
      elseif (edad(i)>=35 && edad(i)<=60)
            Ad=Ad+1;
      elseif (edad(i)>60)
            M=M+1;
      end
end
fprintf('\nNiños = %1.0f\n',N);
fprintf('Adolescentes = %1.0f\n',A);
fprintf('Jóvenes = %1.0f\n',J);
fprintf('Adultos = %1.0f\n',Ad);
fprintf('Mayores = %1.0f\n',M);
```

Cuando ejecutamos el programa para 18 personas sería:

```
Ingrese número de personas: 18
Ingrese dato (1): 45
Ingrese dato (2): 67
Ingrese dato (3): 12
Ingrese dato (4): 9
Ingrese dato (5): 25
```

```
Ingrese dato (6): 36
Ingrese dato (7): 12
Ingrese dato (8): 70
Ingrese dato (9): 1
Ingrese dato (10): 45
Ingrese dato (11): 15
Ingrese dato (12): 26
Ingrese dato (13): 29
Ingrese dato (14): 11
Ingrese dato (15): 20
Ingrese dato (16): 50
Ingrese dato (17): 6
Ingrese dato (18): 55

Niños = 6
Adolescentes = 1
Jóvenes = 4
Adultos = 5
Mayores = 2
```


Otra forma de hacer el mismo programa sería usando cambiando las condiciones. Sería bueno que analice dichas condiciones y el real funcionamiento de la sentencia "***if-elseif***"

```
%Programa para seleccionar por edades.
clc
n=input('Ingrese número de personas: ');
edad=leervector(n);
N=0;
A=0;
J=0;
```

```
Ad=0;
M=0;
for (i=1:n)
      if (edad(i)<=12)
            N=N+1;
      elseif (edad(i)<=17)
            A=A+1;
      elseif (edad(i)<=35)
            J=J+1;
      elseif (edad(i)<=60)
            Ad=Ad+1;
      else
            M=M+1;
      end
end
fprintf('\nNiños = %1.0f\n',N);
fprintf('Adolescentes = %1.0f\n',A);
fprintf('Jóvenes = %1.0f\n',J);
fprintf('Adultos = %1.0f\n',Ad);
fprintf('Mayores = %1.0f\n',M);
```