



BOLETÍN DE EJERCICIOS TEMA 1

PROFESOR: MARTÍN GARCÍA FIGUEIRA
martin@ciclosmontecastelo.com

BLOQUE I – VARIABLES

1. Indica cuál es el error en las siguientes sentencias:

- a. `String` x = 5
- b. `char` c = "Pepe"
- c. `Int` n = "Cinco"
- d. `int` i;

i = i + 2.5

2. Indica que tipo de variable utilizarías para guardar los siguientes valores:

- a. True
- b. "s"c
- c. "23w-67"
- d. 45
- e. 278
- f. -90
- g. -130
- h. 13,9
- i. 25,7878785694321
- j. 3.567.900

3. Indica cuál es el problema (si es que hay alguno) en los siguientes ejemplos:

- a. `short` v1 = 379

`int` v2 = 3500



`v1 = v2`

b. `single x1 = 13.5`

`double x2 = x1`

`decimal x3 = x2`

c. `short d = 345`

`byte b = d`

d. `int raiz = Math.Sqrt(9)`

e. `decimal division = 9 / 5`

f. `short cantidad`

`Console.WriteLine("Introduzca cantidad: ")`

`cantidad = Console.ReadLine()`

g. `int y = 30000000`

h. `short x = Convert.ToInt16(y)`

4. El programa debe solicitar al usuario que ingrese un número. Este número es recogido en una variable de tipo **Double** llamada **radio**.

A continuación, se debe calcular en sendas variables, la **longitud** del círculo, el **área** y el **volumen** de la esfera. Para ello haced uso de una constante de tipo double denominada **PI** cuyo valor es 3.141516.

$\text{Longitud} = 2 * \text{PI} * \text{radio}.$

$\text{Area} = \text{PI} * \text{radio}^2$

$\text{Volumen} = (4/3) * \text{PI} * \text{radio}^3$

El programa debe mostrar estos resultados por pantalla.



5. Diseñar un programa que pide al usuario que introduzca el **precio** (**variable Double**) y la **cantidad** (**variable Short**) de un determinado producto.

A continuación, calcula el importe final que será $\text{precio} \times \text{cantidad}$ y muestra por pantalla el siguiente mensaje:

“El precio del producto es x euros y la cantidad que compra es y por lo tanto debe pagar z euros. Vuelva pronto.”

6. Realizar un programa que solicite al usuario dos números y calcule guardándolo en SENDAS VARIABLES: la **SUMA**, la **RESTA**, y la **MULTIPLICACIÓN** de esos dos números y muestre por pantalla los resultados con mensajes como:

“La suma es “, suma.

“La resta es “, resta.

“La multiplicación es “, multiplicación.

7. Una temperatura dada en grados Celsius (centígrados) puede ser convertida a una temperatura equivalente Fahrenheit de acuerdo a la siguiente fórmula :

$$f = (9/5) * C + 32$$

Escribir un programa que lea la temperatura en grados centígrados y la convierta en grados Fahrenheit.

8. La fórmula de los gases ideales es: $pV = nRT$, en donde :

P: es la **presión** en atmósferas.

V: es el **volumen** en litros.

n: es el **número de moles**.

R: Es una constante que vale 0.082

T: Es la **temperatura** en Kelvin



Crear un programa que lea por teclado el volumen, el número de moles, y la temperatura y calcule la presión, informando al usuario con el siguiente mensaje:

“Con un volumen de V litros ,y una temperatura de T kelvin, n moles de un gas ideal tienen una presión de P atmósferas”

9. Escribir un programa que permita intercambiar los contenidos de dos variables x e y de un cierto tipo de datos.
10. Escribir un programa que lea las horas trabajadas en un mes y la tasa a la que se paga cada hora y el IRPF que se aplica como impuesto

El programa debe calcular el salario bruto y el salario neto del trabajador.

El salario neto se calcula restando al salario bruto el IRPF asignado..

11. El programa deberá solicitar al usuario que introduzca dos datos de tipo **double**: **espacio** en metros y **tiempo** en segundos. Con estos datos informará de la velocidad en metros por segundo según la mítica fórmula:

Velocidad = espacio / tiempo

El mensaje será: “La velocidad es de “ + velocidad + “metros/segundo”

12. Diseñar un programa que a partir del **lado** de un cuadrado calcule su **área**, su **perímetro** y su **diagonal**:

$$p = 4 * \text{lado}$$

$$d = \text{lado} \times \sqrt{2}$$

$$A = \text{lado}^2$$

13. Diseñar un programa que a partir de los lados de los catetos de un triángulo rectángulo calcule la hipotenusa con dos decimales según el teorema de Pitágoras:

$$h^2 = a^2 + b^2$$

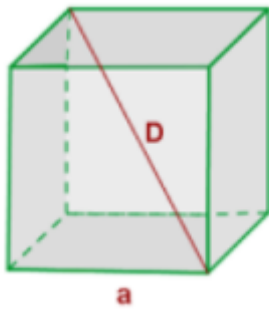
$$h = \text{Raiz}(a^2 + b^2)$$



14. Diseñar un programa que calcule la rebaja realizada sobre un determinado producto a partir del precio inicial y el porcentaje de descuento. El programa debe mostrar un mensaje informando:

“Al aplicar un ... por ciento de descuento el precio final del producto es de precio final y por lo tanto se ahorra ... euros.”

15. Realizar un programa que lea el lado de un cubo y calcule los siguientes datos:



$$D = \sqrt{3} \times a$$

$$A_L = 4 \times a^2$$

$$V = a^3$$



BLOQUE II – OPERADORES Y SENTENCIAS DE SELECCIÓN SIMPLES

1. Indica cuál es el valor que almacenan las variables booleanas en las siguientes expresiones:

- a.

```
int x = 5;
x += 10;
bool b = x < 15;
```
- b.

```
short y = 100;
short z = 50;
int resto = y % z;
bool b = resto == 1;
```
- c.

```
int z = 9;
bool b2 = z > 5;
bool bb = !b2;
```
- d.

```
bool b = true;
bool b2 = false;
bool b3 = b && b2;
bool b4 = b || b2;
bool b5 = !b3;
bool b6 = !b4;
```

2. Indica cuál es el valor final que almacena la variable i en las siguientes expresiones:

- a.

```
int i = 10;
i = i + 10;
i += 5;
```
- b.

```
int i = 11;
if (i % 2 == 0)
{
    i -= 5;
}
else
{
    i += 5;
}
```



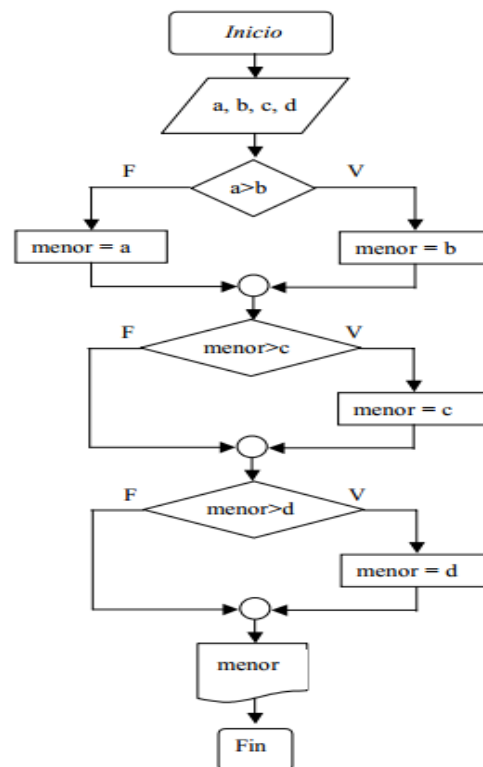
```
int i = 3;  
if (9 % i == 0)  
{  
    i += 10;  
}  
else  
{  
    i -= 10;  
}
```

c.

```
int i = 5;  
if (i > 0 && i < 5)  
{  
    i = 0;  
}  
else  
{  
    i += 5;  
}
```

d.

3. Explica lo que hace la siguiente figura y escribe el algoritmo en código en el Visual Studio.





4. Escribir un programa que decida si un número es par o impar e informe al usuario.
5. Escribir un programa que lea tres números y muestre por pantalla la media aritmética. Si la media es superior a 5 mandará un mensaje de “Alumno aprobado”. En cualquier otro caso el mensaje indica: “Alumno suspenso”.
6. Diseñar un programa que lea (es decir solicita al usuario que los introduzca por teclado) dos números reales, los guarde en dos variables de tipo **double** calcule su suma guardándola en una variable, **SI ES POSITIVA** mostrar por pantalla la raíz cuadrada de la suma (usar la clase `Math.sqrt(variable)`), en cualquier otro caso muestra un mensaje informando de que no se puede calcular la raíz.
7. Escribir un programa que solicite al usuario una contraseña que se guardará en una variable de tipo **String**. El programa debe tener almacenada la contraseña previamente en otra variable. Si las dos contraseñas son iguales manda un mensaje diciendo que la contraseña es correcta en otro caso indica que no lo es.
8. Diseñar un programa que lea dos números y calcule el mayor elevado al más pequeño (usar la clase `Math.pow (base,exponente)`).
9. Debido al sobrepeso **Homer** tiene que realizar su actividad laboral desde casa, para ello la central necesita realizar un programa que lo ayude a gestionar la seguridad de la central. Esta aplicación medirá la presión y la temperatura del núcleo. **Si la presión** es superior a 2 atm debe mostrar el mensaje “Abrir válvula de seguridad”. **Si la temperatura** supera los 500 Kelvin el mensaje debe ser “Reducir la temperatura. Si suceden ambas situaciones debe mostrar ambos mensajes. **Si no debe aparecer “Todo en orden”**.
10. Diseñar un programa que lea un número y si es múltiplo de 2 o de 3 mande un mensaje informando de tal situación, en caso de que no lo sea no hará nada.
11. Diseñar un programa que aplique un descuento del 10% si el precio de un producto es inferior a 100 y del 15% si es superior.
12. Diseñar un programa que pida un número del 1 al 5 y diga si es primo o no.



- 13.** Diseñar un programa que pida un número del 1 al 7 y diga el día de la semana correspondiente.
- 14.** Diseñar un programa que pida un número del 1 al 12 y diga el nombre del mes correspondiente.
- 15.** Diseñar un programa que pida 3 números y los muestre en pantalla de menor a mayor.
- 16.** Diseñar un programa que pida 3 números y los muestre en pantalla de mayor a menor.
- 17.** Diseñar un programa que pida 3 números y los muestre en pantalla de mayor a menor en líneas distintas. En caso de haber números iguales se pintan en la misma línea.
- 18.** Diseñar un programa que pida un número y diga si es positivo o negativo.
- 19.** Diseñar un programa que pida una letra y detecte si es una vocal.
- 20.** Diseñar un programa que pida tres números e indicar si el tercero es igual a la suma del primero y el segundo.
- 21.** Diseñar un programa que muestre un menú que contemple las opciones “Archivo”, “Buscar” y “Salir”, en caso de que no se introduzca una opción correcta se notificará por pantalla.
- 22.** Diseñar un programa que tome dos números del 1 al 5 y diga si ambos son primos.
- 23.** Diseñar un programa que tome dos números y diga si ambos son pares o impares.
- 24.** Diseñar un programa que tome tres números y diga si la multiplicación de los dos primeros es igual al tercero.
- 25.** Diseñar un programa que tome tres números y diga si el tercero es el resto de la división de los dos primeros.
- 26.** Diseñar un programa que muestre un menú donde las opciones sean “Equilátero”, “Isósceles” y “Escaleno”, pida una opción y calcule el perímetro del triángulo seleccionado.
- 27.** Diseñar un programa que pase de Kg a otra unidad de medida de masa,



mostrar en pantalla un menú con las opciones posibles:

- a. Hectogramos**
- b. Decagramos**
- c. Gramos**
- d. Decigramos**
- e. Centigramos**
- f. Miligramos**

28. Diseñar un programa que lea un importe bruto y calcule su importe neto, si es mayor de 15.000 se le aplicará un 16% de impuestos, en caso contrario se le aplicará un 10%.

29. Diseñar un programa que calcule el sueldo que le corresponde al trabajador de una empresa que cobra 40.000 euros anuales, el programa debe realizar los cálculos en función de los siguientes criterios:

- a.** Si lleva más de 10 años en la empresa se le aplica un aumento del 10%.
- b.** Si lleva menos de 10 años pero más que 5 se le aplica un aumento del 7%.
- c.** Si lleva menos de 5 años pero más que 3 se le aplica un aumento del 5%.
- d.** Si lleva menos de 3 años se le aplica un aumento del 3%.



BLOQUE III – SENTENCIAS DE REPETICIÓN

1. Escribe los bucles necesarios para generar las siguientes salidas:

- a) 1,3,5,7,9,11
- b) 2,4,6,8,10
- c) 5,10,15,20,25
- d) 100 90 80 70 60 50

2. Indica con qué valor sale la variable de control en las siguientes situaciones y cuantas veces se repite el algoritmo.

```
int i = 5;

while (i <= 20)
{
    i += 10;
}
```

a)

```
for (int j = 3; j < 8; j += 2)
{
    //...
}
```

b)

```
int i = 10, j=20;

while (i < j)
{
    i += 2;
}
```

c)

3. Describe la salida que genera cada una de las siguientes expresiones:



a)

```
int i = 10;

while (i <= 100)
{
    Console.Write(i + " ");
    i = i + 10;
}
```

b)

```
int i = 100;

while (i > 80)
{
    Console.Write(2 * i);
    i -= 2;
}
```

c)

```
int i = 50, j = 100;

while (i < j)
{
    Console.Write(i + " ");

    j -= 10;

    i += 10;
}
```

d)

```
int i, j, n;

i = 100;
j = 1;
n = 50;

while (i >= n)
{
    Console.Write(2 * j);

    j += 2;
    i -= 10;
}
```



```
int x = 0, i = 0;

while (i < 20)
{
    if (i % 5 == 0)
    {
        x += i;
    }
    i += 1;
}

Console.WriteLine(x);
```

e)

```
int x = 100, i = 1, s = 8;

while (i < x)
{
    s += 1;
    i *= 2;
}

Console.WriteLine(s);
```

f)

```
int x = 100, i = 1, s = 1;

for (i = 2; i < x; i++)
{
    s += i;
    i *= 10;
}

Console.WriteLine(s);
```

g)



```
int i = 1;

for (i = 1; i < 10; i++)
{
    if (i % 2 == 1)
    {
        Console.WriteLine(i);
    }
}

Console.WriteLine(i);
```

h)

4. Explica los fallos de los siguientes algoritmos:

```
int x = 0, i = 1;

while (i < 20)
{
    if (i % 5 == 0)
    {
        x += i;
    }
}

Console.WriteLine(x);
```

a)

```
bool x = true;
while (x)
{
    Console.WriteLine("!");
}
```

b)

```
int x = 100, i = 0, s = 0;

while (i < x)
{
    s += 1;
    i *= 2;
}

Console.WriteLine(s);
```

c)



5. Muestra los números del 320 al 160, contando de 20 en 20 utilizando un bucle `while`
6. Muestra los números múltiplos de 5 y 7 entre 0 y 100 utilizando un bucle `for`.
7. Muestra la tabla de multiplicar de un número introducido por teclado.
8. Escribe un programa que muestre en tres columnas, el número, el cuadrado y el cubo de los 5 primeros números enteros a partir de uno que se introduce por teclado.
9. Bucle `for` descendente que escribe números reales y su raíz cuadrada. Se inicializa la variable `n` a 16, en cada iteración del bucle se decrementa `n` en 2,5.
El mensaje resultado de la ejecución deberá ser el siguiente:

N	raiz(n)
16	4
13.5	3.67423
11	3.31662
8.5	2.91548
6	2.44949
3.5	1.87083
1	1

10. Realizar un programa que calcule la suma de los `m` primeros números, siendo `m` un número **no negativo** que es introducido **POR TECLADO** por el usuario. El programa pedirá que el usuario introduzca un número hasta que un número positivo sea introducido, si no repite el mensaje solicitando el dato.



- 11. Calcular la suma de la serie $1/13 + 1/23 + \dots + 1/n3$. Donde n es un número positivo que se introduce por teclado.**
- 12. Calcular la suma de los divisores de un número n no negativo introducido por teclado.**
- 13. Diseñar un programa que lea un número introducido por teclado y decida si ese número es perfecto.**

Un número es perfecto si es igual a la suma de sus divisores excepto el mismo. Por ejemplo, el 6 es un número perfecto porque la suma de sus divisores da $1+2+3=6$.
- 14. Calcular la suma de los n primeros números que sean múltiplos de 5, siendo n un número no negativo introducido por teclado.**
- 15. Diseñar un programa que calcule el factorial de un número introducido por teclado.**

Ayuda: El factorial de un número n es: $n*(n-1)*(n-2)*\dots*1$

Por ejemplo el factorial de 4 es: $4*3*2*1$



Contadores

- 16. Diseñar un programa que cuente cuántos números múltiplos de 3 hay por debajo de un número n dado por teclado por el usuario.**
- 17. Diseñar un programa que cuente los divisores de un número n introducido por teclado.**
- 18. Diseñar un programa que lea un número positivo y nos diga si ese número es primo o no.**
Utilizar para resolverlo un bucle for. Nota: Un número es primo si solamente es divisible entre 1 y sí mismo. Si el número introducido es negativo o cero el programa debe volver a pedir al usuario que introduzca un número positivo.
- 19. Diseñar un programa que detecte (muestre por pantalla) el primer número que sea múltiplo de 3, 5 y 7 en un intervalo de dos números n y m solicitados por teclado.**



Bucles Anidados:

21. Mostrar la tabla de multiplicar de los 10 primeros números.

22. Realizar un programa que muestre la siguiente salida:

a)

* * * * *

* * * * *

* * * * *

* * * *

* * *

* *

*

b)

*

* *

* * *

* * * *

* * * * *



c)

```
*  
  
* *  
  
* * *  
  
* * * *  
  
* * * * *  
  
* * * *  
  
* * *  
  
* *  
  
*
```

d)

```
1 2 3 4 5 6 7  
  
1 2 3 4 5 6  
  
1 2 3 4 5  
  
1 2 3 4  
  
1 2 3  
  
1 2  
  
1
```



e)
1 1 1 1 1 1 1

2 2 2 2 2 2

3 3 3 3 3

4 4 4 4

5 5 5

6 6

7

- 23.** Realizar un programa que muestre todos los números perfectos por debajo de uno dado.
- 24.** Realizar un programa que cuente cuántos números primos hay por debajo de uno dado.
- 25.** Realizar un programa que genere números aleatorios entre -5 y 5. Para comprobar que realmente son aleatorios haremos un bucle que genera 10 números aleatorios y los muestra por pantalla. A continuación haremos que este bucle se repita 10 veces mostrando por pantalla los números generados en forma de matriz:

-1 , 3 , 4 , -2 , -3 , -1 , 0 , 4 , 1 , 0

-3 , 1 , -1 , -0 , -1 , 1 , 5 , 0 , 2 , 3

-3 , 1 , 0 , -1 , -4 , 5 , 1 , 2 , -1 , -4

...

El programa debe mostrar por cada fila a continuación la siguiente información:



- Cantidad de números negativos
- Cantidad de números positivos.
- Porcentaje de números entre un rango , por ejemplo -2 y 2
- El valor más repetido.



Práctica final:

- 25. Programar el juego de adivina el número. Para ello el programa genera un número aleatorio y el usuario debe de acertar de que número se trata.**

Existen 3 niveles:

- Nivel 1: El número generado está entre 1 y 15. El usuario tiene 3 intentos.
- Nivel 2: Número generado entre 1 y 30 . El usuario tiene 4 intentos.
- Nivel 3: Número generado entre 1 y 50. El usuario tiene 6 intentos.



BLOQUE IV – ARRAYS

En todos estos ejercicios tras crear el array debemos mostrarlo para comprobar el resultado.

Creación e Inicialización de arrays

- 1. Crear el siguiente array: {1,2,3,4,5,6,7,8,9}**
- 2. Crear un array con capacidad para 20 valores de tipo float.**
- 3. Crear un array con capacidad para 10 valores de tipo int.**
A continuación, modificar las posiciones: 0, 2, 5, y 9 para darle valores negativos
- 4. Crear un array con capacidad para 5 valores de tipo String.**
A continuación, guardar en él los nombres de los 5 mejores libros que hayas leído.
- 5. Crear un array con capacidad para 4 valores de tipo short.**
A continuación, ir solicitando al usuario que vaya dando valores a las diferentes posiciones.
- 6. Crear un array con capacidad para 50 valores de tipo double.**
Modificar la última posición para darle valor -1.
- 7. Crear un array con capacidad para n valores de tipo Integer, siendo n una variable cuyo valor debe de ser solicitado al usuario.**
Darle valor a la primera y última posición.
- 8. Crear un array con capacidad para 100 valores de tipo int.**
A continuación, darle valores aleatorios entre 0 y 10 a todas las posiciones.
- 9. Crear un array con capacidad para 100 valores de tipo int.**
A continuación, guardar en las posiciones pares los valores : 1,2,3,4,...
- 10. Crear un array con capacidad para 100 valores de tipo int.**
A continuación, modificar todas las posiciones impares con números aleatorios entre 10 y 35.
- 11. Crear un array con 50 valores aleatorios cuyos valores están entre el rango -10 y 50.**
A continuación, mostrar los valores negativos junto con la posición que ocupan.



Suma

- 12.** Crea un array de 100 valores aleatorios entre -10 y 10 y suma todos los elementos de un array.
- 13.** Crea un array de 100 valores aleatorios entre -10 y 10 y suma todos los elementos de un array que ocupan posiciones con índice par.
- 14.** Crea un array de 100 valores aleatorios entre -10 y 10 y suma los cuadrados de los valores de un array.
- 15.** Crea un array de 100 valores aleatorios entre -10 y 10 y suma aquellos valores de un array que no son negativos.

Contadores

- 16.** Crea un array de 100 valores aleatorios entre -10 y 10 e indica cuantos valores negativos hay en un array.
- 17.** Crea un array de 100 valores aleatorios entre -10 y 10 e indica el porcentaje de números impares que hay en un array.
- 18.** Pide por pantalla un array de 10 elementos e indica cuántas posiciones vacías hay en un array de strings.
- 19.** Crea un array de 100 valores aleatorios entre 500 y 2000 y muestra todos los números de un array que son superiores a 1000

Búsquedas

- 20.** Crea un array de 100 valores aleatorios entre 500 y 2000 y muestra el primer número superior a 1000 y la posición que ocupa en un array.
- 21.** Pide por pantalla un array de 10 elementos e indica la primera posición vacía de un array de strings.
- 22.** Crea un array de 100 valores aleatorios entre -10 y 10 e indica la posición del primer número par que hay en un array.
- 23.** Crea una matriz de nombres de frutas y pide al usuario que ingrese el nombre de una fruta. Utiliza la función `Array.indexOf()` para buscar el índice de la fruta ingresada por el usuario en la matriz de frutas.



- 24. Crea una matriz de nombres de frutas y ordénalos alfabéticamente de forma normal y de forma inversa usando Sort() y Reverse()**

Cálculo del mínimo

- 25. Crea un array de 100 valores aleatorios entre -10 y 10 y calcula el valor más pequeño almacenado en una matriz.**

Comparación

- 26. Algoritmo que compara dos arrays y nos dice si son iguales o no.**
27. Repite el ejercicio anterior utilizando Array.Equals()

Copias

- 28. Algoritmo que crea una copia de un array.**
29. Repite el ejercicio anterior utilizando Array.Copy()

Redimensión

- 30. Algoritmo que permita meter dos tamaños de matriz por pantalla. Crear la matriz con el primer tamaño y redimensionarla con el segundo.**

Problemas

- 31. Crear un array con capacidad para n valores de tipo Integer, siendo n un dato introducido por el usuario. A continuación, realizar las siguientes operaciones:**
- a)** Llenar todas las posiciones con un número negativo
 - b)** Guardar en las 5 primeras posiciones del array y las 5 últimas con números aleatorios
 - c)** El resto de posiciones actualizarlas al número que se os ocurra
 - d)** Mostrar por pantalla el valor de la primera y de la última posición
 - e)** Crear un array copia del array generado.
 - f)** Ordenar el array copia de menor a mayor.
 - g)** Calcular la suma de los valores almacenados en las 10 primeras posiciones



de ambos arrays.

- h) Sumarle 10 a todas las posiciones del array copia.
- i) Poner a cero todas aquellas posiciones que almacenen un número mayor que la media en ambos arrays.
- j) Solicitar al usuario un valor y buscarlo en ambos arrays mostrar por pantalla la posición que ocupa. En caso de no encontrarlo avisa de que no 'está en la matriz'

32. Crear un array con capacidad para n valores. Llenarlo con valores aleatorios entre 0 y 10.

- a) Calcular la media de los elementos del Array.
- b) Calcular el porcentaje de valores que están por encima de la media:
 - i. Si la media es inferior a 5 muestra un mensaje informado de que la nota es **suspenso**.
 - ii. Si la media está entre 5 y 7 muestra un mensaje informado de que la nota es **aprobado**.
 - iii. Si la media está entre 7 y 9 muestra un mensaje informado de que la nota es **notable**.
 - iv. Si la media está entre 9 y 10 muestra un mensaje informado de que la nota es **sobresaliente**.
 - v. En cualquier otro caso manda un mensaje de error.

33. Diseñar un programa que calcule y muestre la desviación con respecto a la media de cada elemento de un array de enteros de tamaño n.

Ayuda: Para calcular la desviación de un elemento : $\text{Abs}(\text{valor elemento} - \text{media})$.

Ahora, modifícalo para que calcule la desviación media de un array. Es decir, la media de todas las desviaciones.

34. Declarar dos arrays de 5 elementos, pedir los datos del primero por teclado al usuario.



A continuación, darle valores al segundo array de manera que cada posición almacenará el mismo valor que la misma posición del primer array más 10.
Mostrar ambos arrays por pantalla.

35. Escribir un programa que calcule el módulo de un array de n valores.

El módulo es la raíz cuadrada de la suma de los cuadrados de los elementos de un array.

$$A = \{a, b, c, d\}$$

$$\text{Módulo de } A = \sqrt{a^2 + b^2 + c^2 + d^2}.$$

36. Diseñar un programa que detecte la posición del primer número negativo en un array.

Mostrará por pantalla el valor y el índice.

37. Diseñar un programa que busque un número introducido por el usuario.

- Es decir, si el usuario introduce un 3 el programa buscará si en el array hay alguna posición que almacene ese valor, en cuyo caso muestra por pantalla el índice.
- Si hay más de una aparición las muestra todas, si no hay ninguna informa de dicha circunstancia.

38. Declarar una matriz con capacidad para 100 strings y presentar un menú al usuario con las siguientes opciones:

- Introducir nombres:** Debe solicitar al usuario que introduzca su nombre y guardarlo en la primera posición vacía de la matriz.
- Mostrar nombres:** Muestra todos los registros guardados en la matriz.
- Contar nombres:** Muestra el número de nombres guardados en la matriz.
- Mostrar espacio:** Muestra el número de posiciones vacías de la matriz.



- e) **Buscar nombre:** Solicita al usuario que introduzca el nombre que desea buscar y le muestra la posición en la que se encuentra, en caso de que no encuentre debe mostrar un mensaje informando.
- f) **Eliminar nombre:** Solicita que al usuario que introduzca el nombre que quiere eliminar, lo busca y lo elimina. Para eliminarlo deberá establecer la posición que ocupaba el nombre a null y debería reordenar la matriz para que no queden posiciones vacías por el medio.
- g) Salir.

39. Crear un programa que permita generar n valores enteros entre dos límites que son introducidos por el usuario.

A continuación, el programa muestra la siguiente información:

- a) Número de elementos generados por encima de la mitad. Por ejemplo si el rango es (10,20) , serían números mayores a 15.
- b) Número de veces que fue generado el límite inferior.
- c) Posición del primer valor generado que es superior al valor que representa la mitad del rango.

40. Diseñar un programa que permita calcular el valor más alto de una matriz.

41. Utilizando la librería Array:

- a) Declara dos matrices de tipo Integer e inicializa la primera a unos valores cualesquiera
- b) Realiza una copia de los valores de la primera en la segunda.
- c) Ordena la matriz original.
- d) Utilizando un Foreach recorre ambas matrices para mostrar los valores almacenados
- e) Solicitar al usuario un valor, buscarlo en la matriz ordenada y mostrar la posición por pantalla.



- 42. Diseñar un programa que cuente la cantidad de números primos que hay en un determinado array.**
- 43. Almacenar 25 números aleatorios en un array.** Mostrar por pantalla los valores de la matriz situando un * delante de aquellos valores que aparezcan repetidos más de una vez.