

**PRÁCTICA 1**

1) ¿Qué tipo de variable escogería y por que para representar lo siguiente?

Considera además los modificadores de tipos: signed/unsigned, short/long

- Edad de una persona
- Peso de un producto
- DNI
- Iniciales de un nombre
- Coordenadas de un punto en el plano
- Precio de un producto
- Nombre y apellido de una persona
- Dimensiones de una habitación

2) Comentar si las siguientes declaraciones de variables son correctas o no, de existir errores indicar claramente cuales son.

- a) `int x = 1000;`
- b) `var char letra= "a";`
- c) `float x = 3,5;`
- d) `int a,b = 20,40;`
- e) `int x, y, z=45;`
- f) `unsigned double d = 45;`
- g) `int a=10; b=15;`
- h) `int 10="diez";`
- i) `int a=4, b=a;`
- j) `long float lf=123456.789;`

3) Uso de **macros** - Crear macros para lo siguiente

- a) Obtener el doble de un número
- b) Determinar si un numero está entre el 0 y el 100
- c) Determinar si un número es divisible por 5
- d) Determinar si dos números son consecutivos, por ejemplo, 5 y 4 o 33 y 34, pero no 40 y 42 y para cada una crear y ejecutar un programa mostrando su uso.

4) Crear un programa que permita sumar los primeros n numeros enteros, donde n>0 es introducido por el usuario.

5) Crear programas que respondan a cada consigna. Usar macros tanto como sea posible.

- a) Solicitar al usuario que ingrese por pantalla dos enteros y luego indique cual es el mayor de los dos, cuantos son positivos y cuantos son pares.
- b) Repita el ejercicio pero ahora para tres enteros.

6) Escriba programas que realicen lo siguiente:

- a) reciba dos enteros positivos n y m, asegurarse que el usuario ingrese tales enteros respetando que n<m  
- genere la salida n, n+1, n+2, ..., m-2, m-1,m  
Ejemplo n=2, m=10, debe generarse: 2, 3, 4, 5, 6, 7, 8, 9, 10
- b) reciba tres enteros positivos n, p y m, asegurándose que n<m  
- genere la salida n, n+p, n+2.p, n+3.p, .... k (k<=m y es de la forma n+NRO.p)  
Ejemplo n=2, p=3, m=10, debe generarse: 2, 5, 8  
Ejemplo n=1, p=4, m=21 , debe generarse 1,5,9,13,17,21

7) a) Calcule mediante bucles **for** las siguientes sumatorias y productorias.

$$\sum_{n=1}^{100} \frac{1}{n}$$

$$\sum_{k=1}^{30} \frac{1}{k^2}$$

$$\sum_{j=1}^{25} \frac{1}{j}$$

$$\sum_{i=2}^{10} (i+1)i$$

$$\prod_{i=1}^8 i$$

$$\prod_{k=1}^{10} \frac{1}{k^2}$$

$$\prod_{m=1}^5 (m^2 + 3m)$$

$$\prod_{n=0}^{10} (n+1)^2$$

b) Seleccione tres de ellos ítems y reescribalos usando **while**.

### 8) Calculadora para enteros

Crear un programa que reciba argumentos por línea de comandos.

El primer argumento (que no sea el nombre del programa en sí) debe ser un número entero, el segundo un operador (+,-,x,/) y el tercero otro número entero.

El programa deberá:

- analizar que la cantidad de argumentos sea correcta, en caso contrario finaliza el programa informando tal situación al usuario
- analizar que cada argumento representa lo requerido, es decir que el 1ro y el 3ro sean números enteros y el 2do sea uno de los operadores mencionados, en caso contrario finalizar e informar
- mediante el uso de **switch** determinar tipo de operación a realizar y resolverla
- cubrir el caso de división por cero

Ayuda: Considerar el uso de la función atoi de stdlib.h (investigar)

### 9) Año bisiesto

Según el calendario gregoriano, es el año cuyo número es divisible por 4, excepto los seculares (divisibles por cien) de número no divisible por 400. (Ejemplo: 1700, 1800, 1900, son no bisiestos, 1996 y 2000 fueron bisiestos). Construya un programa que determine si un año ingresado es bisiesto.

Referencia: [https://es.wikipedia.org/wiki/A%C3%B1o\\_bisiesto](https://es.wikipedia.org/wiki/A%C3%B1o_bisiesto)

10) Realice un programa que convierta un número binario en su equivalente decimal.

11) Realice un programa que convierta un número decimal a su equivalente en binario.

12) Crear un programa que permita el pasaje de temperaturas en grados Celcius a Farenheit y viceversa.

13) Escriba un programa que cumpla con los siguientes requisitos:

- El programa debe generar un número n aleatorio entre 0 y 500, luego el usuario, mediante ingreso de números, deberá adivinar dicho valor.
- En cada intento el programa debe responder "el número es mayor" o "el número es menor", según corresponda.
- El usuario dispondrá como máximo de 15 intentos.
- En caso de éxito deberá presentarse un cartel diciendo: "*Adivinó, felicitaciones!*".
- En caso de fracasar deberá mostrar: "El número era n".

14) Crear un programa, pero ahora Ud deberá pensar un número y la computadora intentará adivinarlo mediante consultas estilo "es este número?" y Ud deberá responder: "si" o bien "es menor" o bien "es mayor" (por ejemplo puede introducir 0, -1 y 1 para cada uno de estos casos a fin de evitar el uso de texto). El programa debe seguir la misma estrategia que Ud utilizó en el ejercicio anterior para intentar acertar el número.

15) Empleando bucles for anidados intente generar las siguientes salidas por pantalla

| a)    | b)     | c)         | d)      |
|-------|--------|------------|---------|
| *     | ++++++ | \$\$\$\$\$ | @.@.@.@ |
| **    | ++++++ | \$\$\$\$   | .@.@.@. |
| ***   | ++++++ | \$\$\$     | @.@.@.@ |
| ****  | ++++++ | \$\$       | .@.@.@. |
| ***** |        | \$         |         |

16) Crear un programa que calcule los divisores de un número dado.

17) Crear un programa que determine si un número entero es primo.

18) Crear un programa que cuente cuantas cifras tiene un número entero.

19) Crear un programa que calcule el factorial de un número.

20) Crear un programa que permita al usuario ingresar una lista de enteros positivos tan larga como quiera, solo debe indicar que finaliza la entrada al ingresar un 0.

Durante la carga se debe contabilizar la racha de enteros consecutivos más larga, al finalizar la carga mostrar dicho valor.

Ejemplo:

Carga: 4 5 6 9 3 **6 7 8 9** 2 1 4 5 0(fin) – Racha más larga 4

Carga: 4 3 8 **1 2 8 9** 2 8 4 1 6 10 0(fin) – Racha más larga 2

21) Crear un programa que muestre los **números de Armstrong** comprendidos entre 1 y 1000.

Un **número de Armstrong** o **número narcisista**, es todo aquel número **número** que es la suma de cada uno de sus mismos dígitos elevado al **número** total de dígitos.

Por ejemplo el **número** 153 es de **Armstrong** ya que este posee 3 dígitos y la suma de cada uno de sus dígitos elevado a 3 es igual a si mismo,  $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$ .

Pista: son 5 en total en el rango indicado.

22) Crear un programa que muestre los **números perfectos** comprendidos entre 1 y 10000.

Un **número perfecto** es un número natural que es igual a la suma de sus divisores positivos propios (es decir, excluido dicho número).

Ejemplo: el 6 es un número perfecto porque sus divisores propios son 1, 2 y 3.  $6 = 1 + 2 + 3$

Pista: hay 3 números perfectos más en el rango indicando.