

I- INTRODUCCIÓN A LA PROGRAMACIÓN EN C++

1. Resuelve las siguientes expresiones de decimal a binario:

- a) 64
- b) 100
- c) 111
- d) 145
- e) 25,25

2. Resuelve las siguientes expresiones de binario a decimal:

- f) 001100
- g) 011100
- h) 111100
- i) 111111
- j) 111000

3. Resuelve las siguientes expresiones de hexadecimales a decimales:

- k) C
- l) 9F
- m) D52
- n) 67E
- ñ) ABCD

II- TIPOS DE DATOS

1. ¿Qué son los tipos de datos en c++?
 2. ¿Cuál es la diferencia entre un tipo de dato integer y un flotante?
 3. Explique cómo funciona un tipo de dato Booleano.
 4. Nombre los tipos de datos en c++.
 5. ¿Cuál es el rango de un dato tipo char?
 6. ¿Cuáles son los valores que acepta un dato booleano?
 7. ¿Cuál es el tamaño de un dato tipo int?
- II- ¿Cuál es el tamaño de un dato tipo char?

III-DISEÑO DE UN PROGRAMA BÁSICO

1. Realiza una nueva solución en visual studio c++.
2. Cree la estructura básica para un programa c++.
3. Realiza un programa que te permita ingresar los datos de un alumno utilizando distintos tipos de datos conforme sea conveniente.
 - Nombre del alumno.
 - Matricula del alumno.
 - Edad.
 - Semestre.
4. Realiza un programa que ingrese dos números y muestre el resultado de las siguientes operaciones.
 - Suma.
 - Resta.
 - Multiplicación.
 - División.

IV- CONTROL DE FLUJO

1. ¿Explica cómo funciona la condicional if?
2. ¿Explica cómo funciona la condicional if else?
3. ¿Explica cómo funciona un switch?
4. Escribe la sintaxis de la condicional if en c++
5. Escribe la sintaxis de la condicional if else en c++
6. Escribe la sintaxis del switch en c++
7. Realiza un programa en c++ que ayude a un trabajador a saber el tipo de descuento que le corresponde dependiendo el tipo de membresía que tiene el cliente.
 - Membresía Tipo 1: 5% descuento
 - Membresía Tipo2: 10% descuento
 - Membresía Tipo3 :20% descuento
8. Realiza un menú sencillo utilizando la estructura switch.

V- ARREGLOS

1. ¿Qué es un arreglo en c++?.
2. ¿Cómo funciona un arreglo unidimensional?
3. ¿Cómo funciona un arreglo bidimensional?
4. Declara un arreglo de una dimensión.
5. Declara un arreglo de dos dimensiones.
6. Realiza un programa utilizando arreglos para ingresar 3 números enteros, y calcular el menor y el mayor de los números ingresados.
7. Realiza un programa utilizando arreglos bidimensionales que te permita ingresar 4 números de teléfono con un máximo de 10 dígitos, estos se tendrán que guardar y posteriormente mostrar.

VI-ESTRUCTURAS

1. ¿Qué es una estructura en c++?
2. En un proyecto de Visual Studio cree una estructura que permita ingresar los siguientes datos de un alumno.
 - Nombre.
 - Matricula.
 - Semestre.
 - Correo.
 - Carrera.
3. A continuación, imprima los datos ingresados en la estructura anterior.
4. Realizar un programa que permita guardar los datos de un alumno y los datos de la escuela en la que estudia (estructuras anidadas). Para posteriormente visualizarlas.

Alumno

- Nombre.
- Matricula.
- Semestre.
- Correo.
- Carrera.

Escuela

- Nombre de la institución
- Dirección
- Teléfono

VII- CICLOS SIMPLES

1. ¿Qué son los ciclos?
2. ¿Cuáles son los tipos de ciclos?
3. ¿Cómo funciona el ciclo for?
4. ¿Cómo funciona el ciclo while?
5. ¿Cómo funciona el ciclo do while?
6. Escribe la sintaxis del ciclo for.
7. Escribe la sintaxis del ciclo while.
8. Escribe la sintaxis del ciclo do while.
9. Realiza un programa que permita saber la suma de los 100 primeros números naturales (con un while).
10. Utilizando un ciclo for realiza un programa para sumar los números enteros pares del 2 al 100.

VIII- CICLOS ANIDADADOS

1. ¿Que son los ciclos anidados?
2. ¿Qué utilidad tienen los ciclos anidados?
3. Escribe la sintaxis del ciclo anidado for.
4. Escribe la sintaxis del ciclo anidado while.
5. Escribe la sintaxis del ciclo anidado do while.
6. Realiza un programa en c++ que permita ingresar 15 calificaciones que solo acepten de rango 0 a 100.
7. Realiza un programa en c++ que permita ingresar 15 calificaciones de alumnos en un intervalo de 0 a 10 de lo contrario se deberá volver a pedir la calificación.

IX- FUNCIONES I

1. Realiza las siguientes actividades:

a) Crear un proyecto vacío en C++.

b) Cree las siguientes funciones como se indiquen:

- Una función de tipo **void** sin parámetros que solamente imprima algo en pantalla.
- Una función de tipo **void** que recibirá un texto de parámetro y que dicha función imprima en pantalla si dicho texto contiene algún número.
- Una función de tipo **int** que calcule y retorne la factorial de un número.

X- FUNCIONES II

1. Realiza las siguientes actividades:

a) Crear un proyecto vacío en C++.

b) Cree las siguientes funciones como se indiquen:

- Una función que reciba un número y que eleve al cuadrado dicho número. El parámetro se tiene que enviar usando el paso por referencia, para corroborar imprima dicho número antes y después de mandarla a llamar.
- Una función que comparta el mismo nombre que el anterior sólo con la diferencia de que se envíe un 2do número que sea el exponente para elevar dicho número, en lugar de elevar directamente al cuadrado.

XI- ARCHIVOS I

1. Realiza las siguientes actividades:
 - a) Crear un proyecto vacío en C++.
 - b) Escriba un código que te pida ingresar 5 nombres y estos guárdalos en un arreglo.
 - c) Cree una variable de tipo **fstream** para que guarde en un archivo txt la lista de nombres utilizando la configuración **trunc**.
 - d) Después de guardar el archivo vuelva a abrirlo para luego cargar los nombres en otro arreglo.
 - e) Vuelva a guardar este segundo arreglo, pero en lugar de **trunc** utilice la configuración **app**.
 - f) ¿Qué diferencia hay entre guardar usando **trunc** y usando **app**?

XII- ARCHIVOS II

1. Realiza las siguientes actividades:

a) Crear un proyecto vacío en C++.

b) Desarrolle código que agregue y elimine productos de una tienda. Puede usar la siguiente estructura de ejemplo:

```
struct {  
    char nombre[50],  
    int precio,  
};
```

c) Cuando finalice el programa guarde la lista en un archivo de texto y cuando se vuelva a abrir que cargue dicha información. Para eso cree otra opción que muestre en pantalla la lista con toda la información.

XIII- PUNTEROS I

1. Realiza las siguientes actividades:
 - a) Crear un proyecto vacío en C++.
 - b) Declara una variable de tipo **int** y asigne cualquier valor.
 - c) Declara un puntero del tipo **int** y asigne la dirección de la variable anterior.
 - d) Utilizando el puntero modifique el valor de la variable **int** y verifique si este valor cambió.
 - e) Revise la dirección de la variable y el valor del puntero. ¿La dirección del puntero es la misma que la de la variable?

XIV- PUNTEROS II

1. Realiza las siguientes actividades:
 - a) Crear un proyecto vacío en C++.
 - b) Ingrese 5 números y guárdelos en un arreglo de tipo **int**.
 - c) Cree un puntero que apunte al arreglo y muestre la lista de números a través del puntero.
 - d) Revise si la dirección de cada variable del arreglo coincide con la que tenga el puntero en cada iteración.

XV- PUNTEROS III

1. Realiza las siguientes actividades:
 - a) Crear un proyecto vacío en C++.
 - b) Ingrese 20 números y guárdelos en 4 arreglos tipo **int** o un arreglo bidimensional.
 - c) Declare un arreglo de punteros y asígnele la dirección de los arreglos en cada puntero.
 - d) Muestre los valores a través del arreglo de punteros.
 - e) Revise si la dirección de cada variable del arreglo coincide con la que tenga el puntero en cada iteración.