

Ejercicios de C++

1. Uso de variables, funciones y flujos de control

1. Calcular la suma, resta, división y multiplicación de 2 números.

- Validar divisiones por 0.
- Utilice al menos 2 sobrecargas de función.

2. Calcular el perímetro y área de un círculo de radio r . Utilice funciones para el cálculo.

- Perímetro: $2\pi r$
- Área: πr^2

3. Calcular el perímetro y área de un triángulo isósceles de lados a y b . Utilice funciones para el cálculo.

- Perímetro: $2a + b$
- Área: $b * h/2$, con $h = \sqrt{a^2 - b^2/4}$

4. Verificar si un número entregado por teclado es par o impar.

- Utilizar operador mod %.

5. Verificar si un número entregado por teclado es primo.

- Un número se considera primo sólo si es divisible por si mismo (y uno).
- Utilizar operador mod %.

6. Calcular los divisores para un número entregado por teclado.

- Utilizar ciclo while y operador mod %.

7. Calcular $\sum_{i=1}^n (2i - 1)^3$, hasta $n = 10$.

- No olvidar incluir la directiva [<cmath>](#).
- Para calcular potencias, utilizar la función [pow](#).

8. Calcular $\sum_{i=-10}^n \sqrt{i^4 - i^2} + \ln |2i + 1|$, hasta $n = -15$

- La raíz cuadrada se calcula con la función [sqrt](#).
- El logaritmo natural se calcula con la función [log](#).
- El logaritmo común (base 10), se calcula con la función [log10](#).
- El valor absoluto se calcula utilizando la función [abs](#) o [fabs](#).

2. Manejo de caracteres (estilo C)

1. Dada dos variables del tipo char: `char a[] = "Hola"` y `char b[] = "Mundo"`, realizar las siguientes operaciones:

- Contar el número de caracteres utilizando [strlen](#).
- Comparar ambas variables y mostrar por pantalla cuál es mayor. Utilizar [strcmp](#).
- Unir ambas palabras y almacenar el resultado en la variable `char * c`. Utilizar [strcat](#).
- Imprimir número de vocales presentes en `char * c`. Debe **iterar** sobre `c`.

Utilizar “for range”. Disponible sólo al compilar con `-std=c++11`.

- Imprimir el caracter ubicado en la posición 9. ¿Qué arroja la salida?
¿Por qué?

2. Dada la variable `char teclado[7] = "qwerty"`:

- Buscar el caracter `q` y reemplazarlo por `a`.
- Buscar el caracter `w` y reemplazarlo por `z`.

Utilizar las funciones [strchr](#) y [strncpy](#).

3. Dada la variable `char hola[11] = "Hola Mundo"`, buscar el texto `Mundo` y reemplazarlo por `World`. Utilizar las funciones [strstr](#) y [strncpy](#).

3. Manejo de archivos planos

1. Crear archivo `ejemplo1.txt` utilizando la clase [ofstream](#) y agregar 100 líneas de texto. Ejemplo:

```
Línea 1
Línea 2
...
Línea 100
```

2. A partir del archivo creado anteriormente, leer su contenido con [ifstream](#) e imprimirlo por pantalla.
3. Crear un nuevo archivo llamado `ejemplo1-reverse.txt`, que contenga el texto del archivo `ejemplo1.txt`, pero invertido, es decir:

```
Línea 100
Línea 99
...
Línea 1
```

Para ello, se debe leer el contenido del archivo `ejemplo1.txt` y escribirlo, de manera inversa, en `ejemplo1-reverse.txt`.

Considere leer primero el contenido, **concatenarlo de forma inversa** en una variable de tipo `string` y luego escribir todo el contenido de esa variable en el archivo de destino.

4. Buscar la línea 50 del archivo `ejemplo1.txt` y reemplazar el número 50 por un número aleatorio generado a través de la función [rand](#).
5. Crear un nuevo archivo llamado `ejemplo2.txt` con el mismo contenido del archivo `ejemplo1.txt`, pero agregar 3 nuevas líneas entre las líneas 20 y 30. Ejemplo:

```
...
Línea 20
>> Línea agregada 1
```

```
>> Línea agregada 2
>> Línea agregada 3
Línea 30
...
```