



Objetivos de la Guía

- Recordar el uso while
- Recordar el uso de for
- Aprender sobre el uso de funciones

> A - Marcelo y los ciclos en la vida

Marcelo estaba girando en círculos. Mientras giraba en círculos hacia operaciones matemáticas en su cabeza. Mientras giraba se dio cuenta que muchas de las cosas que él hacía en su cabeza se pueden programar.

* Parte 1: Sumar, sumar y sumar

La primera parte consiste en leer 1 entero por pantalla. Luego, se debe realizar la suma de los números que hay entre 1 y ese entero. El resultado de esa suma debe ser mostrado por pantalla.

* Input:

* Output:

Tip 1

El resultado del ejemplo se debe a que se suma $1 + 2 + 3 + 4 + 5$.

* Parte 2: Dividir, dividir y dividir?

La segunda parte consiste en dividir un número n por otro número d hasta el resultado de la división de 0. Marcelo como es curioso, quiere además determinar cuantas veces se pudo realizar la división.

Para realizar esto, se deberá leer por pantalla 2 enteros n y d . n corresponde al número que se quiere dividir y k es el número que se utilizará para dividir

* Input:

* Output:

Tip 1

Recuerda que la división es entre variables de tipo `int`.

Tip 2

El resultado del ejemplo se debe a que:

- Se divide $17/2$, como la división es entero eso da 8
- Ahora, se divide $8/2$, esto da 4
- Ahora, se divide $4/2$, esto da 2
- Ahora, se divide $2/2$, esto da 1
- Ahora, se divide $1/2$, como la división es entero eso da 0

Por lo tanto, se realizaron 5 ciclos



* Parte 3: Función rara

Para esta tercera parte, Marcelo quiere terminar su vicio con los ciclos, realizando el ciclo de un cálculo raro. Este calculo consiste en lo siguiente, el cálculo parte en un entero n :

- Si n es par se le suma 1 a n
- Si n es impar se divide n en 2

Lea por pantalla el número n por pantalla y muestre por pantalla cuantos pasos del cálculo se realizaron hasta que n vale 0.

* Input:

* Output:

Tip 1

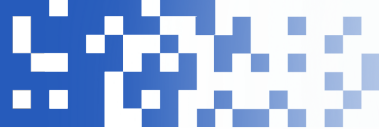
Para verificar si un número es par o impar, utiliza la operación %. Por ejemplo, $3 \% 2$ es 1 y $4 \% 2$ es 0.

Tip 2

El resultado del ejemplo se debe a que:

- 5 es impar, entonces se divide en 2, dando 2
- 2 es par, entonces se le suma 1, dando 3
- 3 es impar, entonces se divide en 2, dando 1
- 1 es impar, entonces se divide en 2, dando 0

Por lo tanto, se realizaron 4 ciclos



> B - Juan Pablo y las funciones de diferentes tipos

Juan Pablo le gusta escribir funciones, porque es una persona muy ordenada. Por lo tanto, cuando programa siempre se pregunta ¿Y cómo puedo hacer una función de esto?. Entonces, cuando llega a casa te invita a que hagas las funciones con él a través de Discord.

* Parte 1: Mayor, menor o igual

Juan Pablo quiere saber si un número es mayor, menor o igual. Por lo tanto quiere escribir tres funciones llamadas `mayor`, `menor` e `igual`. Estas funciones deben recibir dos enteros a y b . Debe retornar `true` si a es mayor/menor/igual a b , debe retornar `false` en caso contrario.

Para probar sus funciones, se deberá leer dos enteros a y b , y mostrar el resultado de cada función, cada una en una línea separada.

* Input:

2 3

* Output:

0
1
0

Tip 1

Recuerde que los tipos de datos booleanos se muestran por pantalla como 1 o 0

Tip 2

La sintaxis de las funciones es `tipo_retorno nombre(parametros)`. Por ejemplo:

```
#include <iostream>
using namespace std;

// esta función retorna el número que se recibe sumado $1$
int suma_1(int a) {
    return a;
}

int main() {
    int n;
    cin >> n;
    cout << suma_1(n) << endl;
    return 0;
}
```

Tip 3

El output se debe a que 2 no es mayor que 3, 2 es menor que 3 y 2 no es igual a 3.



* Parte 2: Ejercicio A, pero con pasos extra

Juan Pablo habla con Marcelo, entonces Juan Pablo quiere hacer el código de Marcelo más ordenado. Por lo tanto quiere realizar funciones que realicen lo explicado en la Parte 1 y Parte 3 del ejercicio A.

Se debe leer un entero n y mostrar por pantalla el resultado de ambas funciones

* Input:

* Output: