

Vamos a practicar el uso de `argv` y `argc` Como en `hackerrank` no podemos obtener directo estos parametros. Por lo cual esta implementado el `main` principal para que lea del teclado simulando que se paso por parametro. y se lo pasa a nuestra funcion `MainConArgumentos (int argc, char argv)`.**

Validar los argumentos

1. Verificar que se han recibido 3 argumentos (extras al llamado del programa). Si no es asi imprimir ERROR 1 y salir.
2. Los argumentos que nos llegarían son NUMERO SIMBOLO NUMERO. Verificar que lo que llega siga este formato. Si no es asi devolver
 1. ERROR 2 si lo primero no es un numero
 2. ERROR 3 si lo segundo no es un simbolo (+-*/%)
 3. ERROR 4 si el tercero no es un numero.
3. Una vez verificado los argumentos, realice la operacion correspondiente, e imprima el resultado con 2 decimales de precision.
 1. Los Test principales usan numeros enteros (positivos o negativos), puede utilizar la funcion `atoi` para convertir un string a numero. Tambien puede probar con la funcion `sscanf` En estos casos recuerde convertir durante la operacion para tener un resultado flotante.
 2. Como **Ejercicio Extra**, los ultimos test (adicionales) validan teniendo en cuenta que los numeros ingresados pueden ser flotantes. Nuevamente, puede utilizar librerias existentes como `atof` y `sscanf`

Input Format

miprograma.exe X SIMBOLO Y

Constraints

1. miprograma.exe siempre estara ahi
2. Simbolo puede ser + - * / %
3. Los numeros son menores a 1000
4. Se utiliza el . como separador de decimales para los numeros no enteros

Output Format

Z.ff

ERROR N

Sample Input 0

```
4  
miprograma.exe 125 + 125
```

Sample Output 0

```
250.00
```

Sample Input 1

```
4  
miprograma.exe 14.5 - 7.2
```

Sample Output 1

```
7.30
```