

Práctica 2

Semestre de otoño

Curso 2019/20

PRIMERA FASE. Uso de estructuras condicionales e iterativas

1.- Codificar un programa que lea dos números enteros por teclado y escriba por pantalla el mayor de estos números, o en caso de que sean iguales, un mensaje indicándolo.

2.- Codificar un programa que lea una secuencia de números enteros por teclado hasta que el usuario introduzca el valor de 0. A continuación, se imprimirá por pantalla el valor de la media aritmética de la secuencia de números (sin considerar el 0).

3.- Codificar un programa que pida al usuario un número entero entre el 1 y el 9, y que a continuación muestre por pantalla la tabla de multiplicar de dicho número. Si el usuario entrega al programa un número fuera de rango, este deberá volver a preguntarle hasta que el dato recibido esté en el intervalo [1, 9].

4.- Codificar un programa que pida al usuario que escriba por teclado una secuencia de caracteres, que serán recibidos de uno en uno por el proceso. La recepción concluirá cuando se reciba el carácter '.' o el usuario haya introducido 20 caracteres. Una vez terminada la lectura de caracteres, se escribirá en la salida estándar el número de vocales en letra minúsculas que se recibieron.

Opcional : Modificar el programa anterior para que cuente y escriba el número **de cada una** de las cinco vocales (en letra minúscula) independientemente.

Se dispone de una constante de tipo *array* como la que se muestra a continuación:

```
const int Avances[10] = {5, 9, 5, -1, -1, 2, -1, 4, -1, 2};
```

Dicho *array* se almacena en la memoria interna de un robot que lo usa como una secuencia de instrucciones para mover su brazo mecánico a izquierda y derecha. Cuando el número es positivo el brazo se mueve a la derecha la misma cantidad de pasos que la cifra marcada por el número. Cuando el valor recibido es un -1 el robot mueve el brazo 3 pasos a la izquierda.

Queremos implementar un programa que analice el contenido del *array*, previamente a su instalación en el robot, para indicar cuál será la posición final del brazo considerándolo situado en un principio en su posición central.

Por ejemplo en el caso del *array* anterior, nuestro programa debería sacar por la salida estándar:

“El brazo del robot acabará situado 15 pasos a la derecha.”

Nota:

Probar el programa con diversos *arrays* para comprobar que el funcionamiento es correcto independientemente de la secuencia de entrada. La longitud del *array* no varía, siempre es 10.

Ejemplos:

{ 5, 9, 5, -1, -1, 2, -1, 4, -1, 2 }	→	15 pasos a la derecha.
{ 5, 1, 5, -1, -1, 2, -1, -1, -1, 2 }	→	Posición central.
{ 5, 1, 5, -1, -1, 2, -1, -1, -1, -1 }	→	5 pasos a la izquierda.

Opcional : Modificar el programa anterior para controlar el caso de incorporar un nuevo elemento al *array* – el número 0. En esta situación, el brazo realiza movimientos hasta que finaliza el *array* o hasta que encuentra el número 0.

Ejemplos:

{ 4, 7, 2, -1, 0, 2, -1, 3, -1, 1 }	→	10 pasos a la derecha.
{ 5, 3, -1, -1, -1, 1, -1, 0, 1, 3 }	→	3 pasos a la izquierda.
{ 4, 1, 3, 6, -1, -1, -1, -1, 1, -1 }	→	Posición central.

De acuerdo con lo especificado en la práctica 1:

1. Se creará mediante el explorador de archivos la carpeta '**P2**' dentro de la carpeta '**Progl**'. En esa carpeta radicarán todos los proyectos de la práctica 2.
2. Cada fase de la práctica dispondrá de un directorio específico. En esta práctica los proyectos de cada fase se denominarán '**P2F11**', '**P2F12**', '**P2F13**', '**P2F14**', '**P2F14O**', '**P2F2**' y '**P2F2O**'.
3. Los archivos fuente correspondientes con cada fase se denominará '**p2f11.c**', '**p2f12.c**', '**p2f13.c**', '**p2f14.c**', '**p2f14o.c**', '**p2f2.c**' y '**p2f2o.c**'.
4. La compilación, enlazado, ejecución y posible depuración de los programas se hará utilizando CodeBlocks.