



Universidad
Francisco de
Vitoria

UFV Madrid

Universidad Francisco de Vitoria

Actividades prácticas: Estructuras básicas del lenguaje

(10 puntos / 15 puntos)

Introducción a la programación

El objetivo de esta práctica es comprobar que el alumno ha comprendido y utiliza correctamente las estructuras básicas del lenguaje (declaración de variables, operaciones aritméticas y lógicas, estructuras condicionales, estructuras repetitivas e invocación a funciones).

Ejercicio 1: robot dispensador de pegamento



Se dispone de un robot dispensador de pegamento, el cual dispone de un programa básico para aplicar pegamento en las esquinas de un rectángulo. Nuestra empresa, acaba de conseguir un contrato para la fabricación de un nuevo producto y hay que cambiar el programa del robot para que aplique el pegamento en un nuevo patrón.

Se debe crear un programa que dispense el pegamento acorde al nuevo *patrón* (antes de poner el programa en el robot, se va a probar en la pantalla del ordenador mediante dibujo). El *patrón* depende del tamaño tamaño del producto, siendo el tamaño N . Al robot hay que indicarle el tamaño N del producto (el tamaño se introduce por teclado, y debe ser un entero positivo mayor que cero. Si N es menor o igual a 0 se debe mostrar el mensaje “Valor no valido” y se termina el programa).

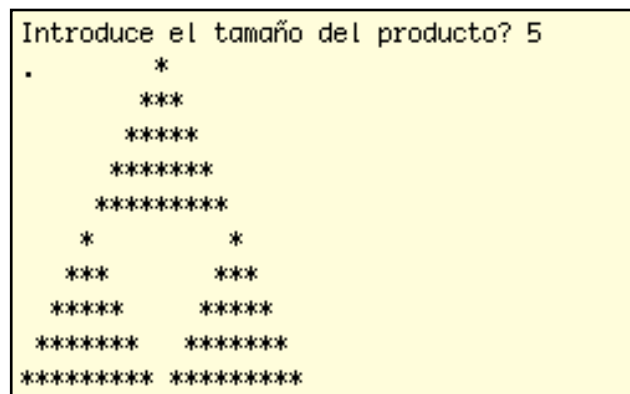
El patrón tiene las siguientes características:

- Está compuesto de 3 triángulos iguales
- Un triángulo de tamaño N debe tener N líneas
- Cada triángulo comienza con una un punto de pegamento (visualmente representado por $*$) y en cada línea se le añaden 2 puntos de pegamento respecto a la línea anterior
- La primera línea del patrón, además del punto de pegamento correspondiente $*$, debe de comenzar con un punto de pegamento de marca de anclaje (representado por $.$)

A continuación, a modo de ejemplo, se muestra un *patrón* de tamaño 3 y otro de tamaño 5.



Un ejemplo de funcionamiento del programa se pueden ver en la siguiente figura.



Ejercicio 2: centralita



En un automóvil, hay dos ECU, una de ellas con 7 sensores y otra con 9 sensores que generan señales registradas como dos números enteros positivos. Un primer número de 7 dígitos, y un segundo número de 9 dígitos (ninguno de ellos puede comenzar con el 0, y los dos números forma un solo código de forma numero1numero2).

$$d_{1n_1}d_{2n_1}d_{3n_1}d_{4n_1}d_{5n_1}d_{6n_1}d_{7n_1}d_{1n_2}d_{2n_2}d_{3n_2}d_{4n_2}d_{5n_2}d_{6n_2}d_{7n_2}d_{8n_2}d_{9n_2}$$

Otra ECU (ECU3) al recibir estos dos números hace una comprobación de los valores recibidos. si los valores recibidos de los sensores son incorrectos la ECU pone el automóvil en modo de protección, avisando al conductor.

La ECU3 para comprobar si los valores generados por los sensores son correctos hace una validación de los mismos, de acuerdo a las siguientes reglas.

1. Doblar el valor de los dígitos que ocupan las posiciones pares (de derecha a izquierda, siendo la posición más a la derecha la posición número 1). Si el valor resultado es un número de dos dígitos, hay que restarle 9 para convertirlo en un número de un dígito.
2. Suma los valores de una cifra del paso anterior.
3. Suma los valores de las posiciones impares empezando por la derecha.
4. Suma los resultados obtenidos en los pasos 2 y 3.
5. Si el resultado del paso 4 es divisible entre 10, slot valores son correctos.

A modo de ejemplo. Si los sensores generan los los números 4556737 y 586899855, el valor a comprobar sería 4556737586899855 y el proceso es el siguiente.

Pos ->	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	4	5	5	6	7	3	7	5	8	6	8	9	9	8	5	5
	↓		↓		↓		↓		↓		↓		↓		↓	
Paso 1	4*2		5*2		7*2		7*2		8*2		8*2		9*2		5*2	
	↓		↓		↓		↓		↓		↓		↓		↓	
	8		10		14		14		16		16		18		10	
	↓		↓		↓		↓		↓		↓		↓		↓	
Paso 2	8		1		5		5		7		7		9		1	
Paso 3		5		6		3		5		6		9		8		5
Paso 4																

90 es divisible entre 10. Valores correctos

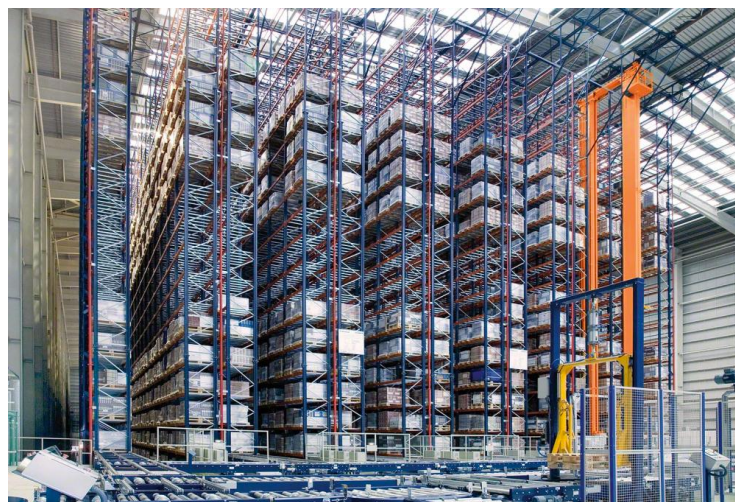
Se debe crear un programa que dados los dos números válidos. Los números se introducen por teclado, y deben ser enteros positivos con 7 y 9 dígitos respectivamente y no pueden comenzar por el dígito 0. Si los numero no tienen 7 o 9 dígitos respectivamente o comienzan por 0 se debe mostrar el mensaje **“Valor no valido”** y se termina el programa (**la comprobación se realiza después de introducir cada uno de los números**). Si la comprobación es correcta (acorde a las reglas descritas con anterioridad), el programa muestra en pantalla **“Valores correctos”**, en otro caso, mostrará el valor **“Entrando en modo de proteccion”**

Un ejemplo de funcionamiento del programa se pueden ver en la siguiente figura.

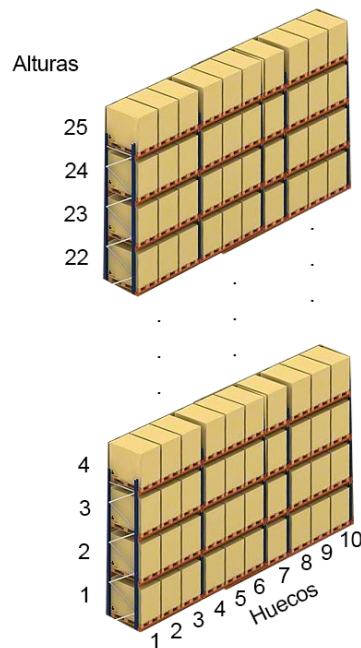
```
Introduce valor del primer codigo? 4556737
Introduce valor del segundo codigo? 586899855
Valores correctos
```

Ejercicio 3: almacén robotizado

Se quiere automatizar un almacén para que la recepción (almacenaje) y expedición (des-almacenaje) de mercancía sea asistida por un robot.

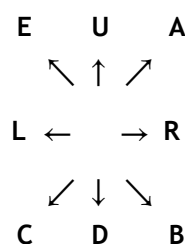


Para simplificar el ejercicio, se va a considerar que el almacén tiene un solo pasillo, con 25 alturas y en cada altura 10 huecos (la dimensión del hueco es 1m. x 1m.) y el robot inicialmente se encuentra en la posición fila 1 y hueco 1.



El robot puede avanzar o retroceder en horizontal y en vertical 1 metro o en diagonal $\sqrt{2}$ metros. Para representar los movimientos del robot se considera la información de la siguiente figura.

El movimiento preferido por el robot es en diagonal, puesto que avanza de forma más rápida hacia su destino. En caso que el camino más corto no sea en diagonal, selecciona un movimiento en vertical o en horizontal (tiene preferencia el movimiento vertical).



Escribir un programa que permita manejar el robot indicando a que fila y hueco debe desplazarse desde la posición en la que se encuentre actualmente, teniendo en cuenta que la posición inicial del robot inicialmente se encuentra en la posición fila 1 y hueco 1. Si la posición indicada no es válida (se lee por teclado como 2 números enteros que representa la fila y la columna. No válida es que este fuera del almacén o en la misma posición del robot) se muestra el mensaje **"Destino no válido"**, si la posición es válida el robot se mueve en el almacén, mostrando los movimientos en la pantalla. Una vez en el destino el programa debe mostrar el mensaje **"Destino alcanzado"** y queda a la espera del siguiente destino (fila, hueco). Si se introduce la fila 0 y el hueco 0 el ejercicio termina.

Un ejemplo de funcionamiento del programa se pueden ver en la siguiente figura.

```
Fila destino? 7
Hueco destino? 3
Destino. Fila: 7. Hueco: 3
Mov      Fila      Hueco
=====
-         1         1
A         2         2
A         3         3
U         4         3
U         5         3
U         6         3
U         7         3
Destino alcanzado
Fila destino? █
```

Para el formato de salida se debe utilizar las siguiente cadenas de texto y especificador de formato:

"Mov\tFila\t\tHueco\n"

"=====\\n"

"%c\t%2d\t\t%2d\\n"

Entrega

Estos ejercicios son de realización obligatoria para todos los alumnos. **Se debe realizar de forma individual.**

Plazo de entrega: **23:59 horas del día 16 de junio de 2021.**

Habrà que entregar a través de Canvas un fichero de nombre **ejercicios1.c** con el siguiente contenido:

- Un programa principal que contenga un menú para seleccionar la parte de la práctica a ejecutar (solo debe permitir los valores del 1 al 5). Si se introduce un valor diferente a los valores correctos, se debe indicar el error mediante el mensaje “ERROR: Opcion introducida no valida” y se debe mostrar otra vez el menú. No hay que contemplar el caso de leer de forma errònea una letra.

```
1.- Ejercicio 1
2.- Ejercicio 2
3.- Ejercicio 3
4.- Ejercicio opcional
5.- Salir

Introduzca una opcion (1-5) █
```

- Si alguna de las partes no se implementa, el programa debe indicarlo en pantalla con el siguiente mensaje: “Parte no realizada”
- Cada una de las partes se implementará en un procedimiento diferente, de acuerdo a los siguiente nombres:

- o **void ejercicio1();**
- o **void ejercicio2();**
- o **void ejercicio3();**

Se considerará **plagio** entregar un código idéntico total o parcialmente al de otro compañero. Tampoco se admitirá un código idéntico a otro presente en cualquier página de Internet, incluso aun citando la fuente. **El profesor dispone y usará herramientas de detección automática de plagio.**

Puntuación

Estos ejercicios obligatorios se puntuará de 0 a 10, nota que se contabilizará en el 15% de la nota final de la asignatura (proyectos y actividades teórico-prácticas) en el caso de la evaluación continua (dentro de ese 15% estos ejercicios son el 10%). Las prácticas no entregadas se computarán como un 0.

Elementos a evaluar:

- **Incluir el nombre del autor como primer comentario del programa.** Si para la correcta compilación del programa hay que incluir algún parámetro opcional al compilador, se debe incluir en una comentario debajo del nombre de alumno la correcta llamada al compilador.
- La no existencia de errores sintácticos en el código (es decir, **el código debe compilar**). En caso contrario, **una práctica cuyo código fuente dé errores de compilación se considerará suspensa directamente.**
- Correcto funcionamiento de los distintos ejercicios entregados. Es decir, se valorará positivamente que no haya errores lógicos.
- Limpieza y claridad en el código.
- Presencia de comentarios en el código.
- **Ver documento “Errores a evitar.pdf”**

De forma numérica los pesos de cada parte de la practica son:

Ejercicio 1: robot dispensador de pegamento	3.5
Ejercicio 2: centralita	3.5
Ejercicio 3: robot	3
Total	10

La puntuación máxima a obtener puede ser un 10 si se realizan todos los apartados, los ejercicios ejecutan de la forma descrita en el enunciado, y no se comete ninguno de los errores descritos en el documento “*Errores a evitar.pdf*”