Universidad Francisco de Vitoria

UFV Madrid

Universidad Francisco de Vitoria

Práctica 1: Estructuras básicas del lenguaje

Introducción a la programación

PRÁCTICA 1: ESTRUCTURAS BÁSICAS DEL LENGUAJE

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). Y con motivo del 35 aniversario del lanzamiento del *Super Mario Bros* original, *Mario* será nuestro protagonista, y el juego ***Super Mario Odyssey*** nuestra historia.

Parte 1: reprogramación de Cotorrete

El transporte que utiliza Mario para moverse por los diferentes mundos y llegar a alcanzar a Bowser es la nave Odyssey. El combustible de la Odyssey son las lunas que hay repartidas por los diferentes mundos que Mario puede recorrer.

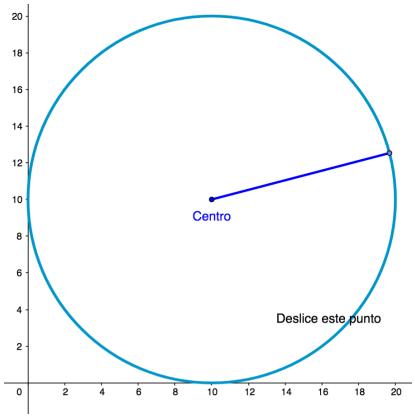
El problema es que, una vez que Mario ha recorrido todos los mundos desbloqueados, no tiene suficientes lunas para hacer volar la Odyssey y avanzar camino del Reino Luna para enfrentarse a Bowser y salvar a la Princesa Peach y a la hermana de Cappy de sus garras.

En este punto, Mario y Cappy se han enterado que sois expertos programadores, y os pide ayuda. Ya que están dentro de un juego, necesitan que modifiquéis el código del juego, añadiendo una funcionalidad de creación de lunas, para poder avanzar y lograr salvar a Peach y a Tiara.

Vuestro trabajo es reprogramar a Cotorrete, para que en lugar de dar pistas sobre la ubicación de las lunas, genere una luna del tamaño que le pida Mario (mayor o igual a 5), puesto que a mayor tamaño luna, mayor aporte de combustible para la Odyssey.

Vuestra labor va a consistir en crear un programa que cree una luna de tamaño N y la muestre por pantalla (el tamaño se introduce por teclado, y debe ser un entero positivo mayor o igual a 5. Si N es menor a 5 se debe mostrar el mensaje ***"No moon"*** y se termina el ejercicio - hay que volver al menú de selección de ejercicios).

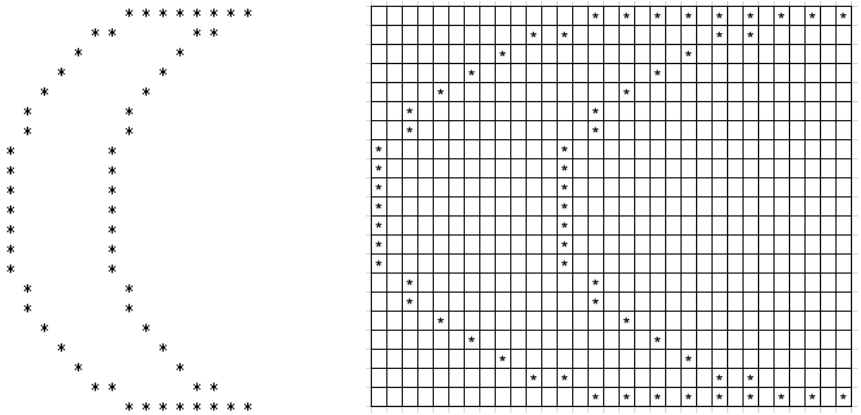
- Una luna está compuesta de dos arcos de radio N, cuyo centro se encuentra en el punto (N, N) del plano cartesiano. Para un N = 10 la representación de la circunferencia sería la siguiente.



- Los símbolos de la luna pueden ser un * o un espacio en blanco. En horizontal cada símbolo ocupa dos espacios y en vertical sólo un espacio
- Como la representación en la pantalla no es exacta, vamos a considerar que un punto pertenece a la circunferencia si tiene un error de ± 0.5.

*Pista: utilizar la ecuación de pertenencia de un punto a una circunferencia para dibujar o no un **

A continuación, a modo de ejemplo, se muestra la luna de tamaño 10 que debe generar Cotorrera través de vuestro programa (se muestra también distribuido en una cuadricula para que se vea el número de espacios que separa cada elemento).



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

```
introduzca el tamaño de la luna? 5
     * * * * * *
   *       *
  *         *
*           *
*           *
*           *
*           *
*           *
  *         *
    *       *
      * * * * * *
```

Parte 2: Batalla contra los Broodals

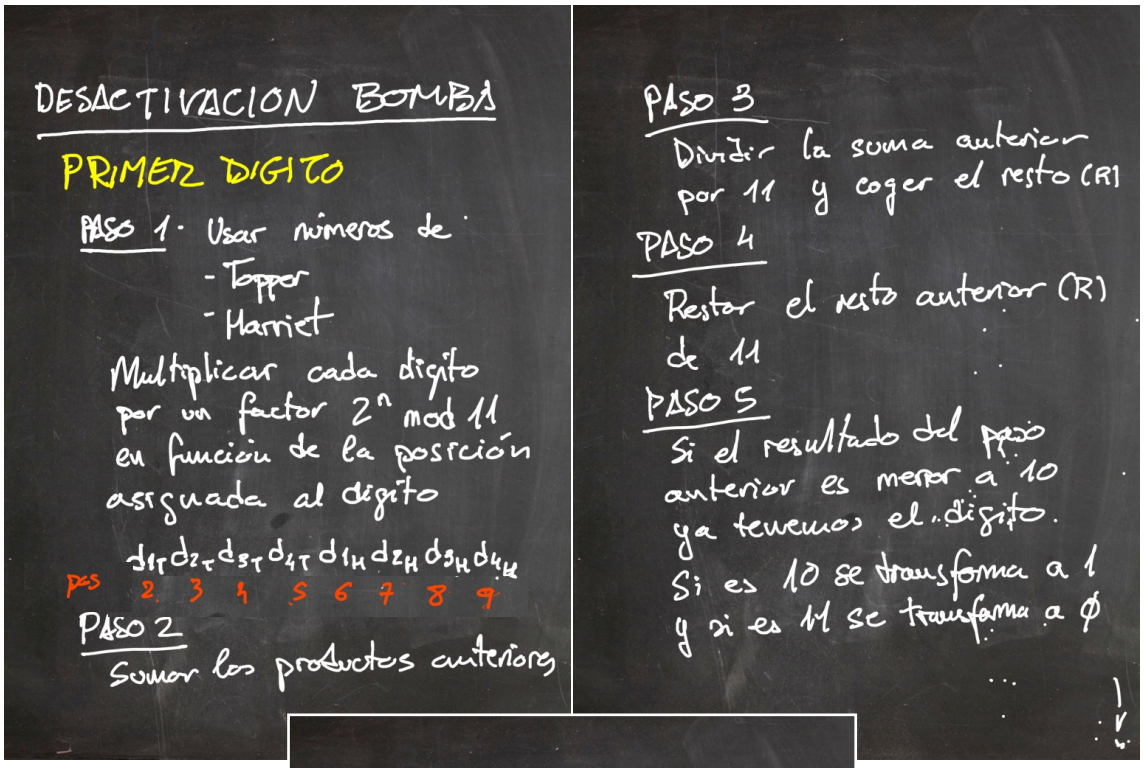
En la Cara oculta de la Luna, al llegar a la Torre Madriguera, Mario se encuentra con los Broodals (*Topper, Harriet, Spewart y Rango*) en actitud poco amistosa.

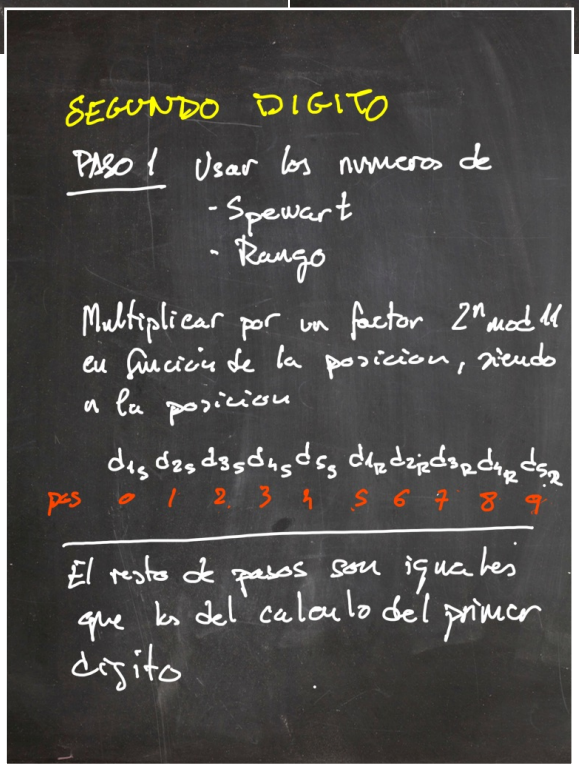
Bowser ha entrado en el código del juego y ha modificado la forma de luchar de los Broodals. Ahora en lugar de lanzar sombreros, bombas, veneno han puesto una mega-bomba en el castillo de Peach, para hacer que Mario vuelva al castillo y no acceda a la Torre Madriguera. El salto de Mario sobre s cabezas de los Broodals y a no tiene efecto, gracias a la reprogramación de Bowser. Así que no hay forma de detener a los Broodals, ni de desactivar la mega-bomba.

Pero un momento, por suerte, Cappy los oyó conversar sobre la forma de activar y desactivar la bomba. La bomba tiene un código de 18 dígitos de longitud para activarla y desactivarla. Para activarla, cada uno de los Broodals introduce un número entero positivo de cuatro o cinco cifras (ninguno de ellos puede empezar por 0). Topper (T) introduce el primer número de 4 dígitos, Harriet (H) el segundo número también de 4, a continuación Spewart (S) introduce un código de 5 dígitos y por último Rango (R) introduce otro número de 5 dígitos. Con todo ello se forma el código de activación de la bomba.

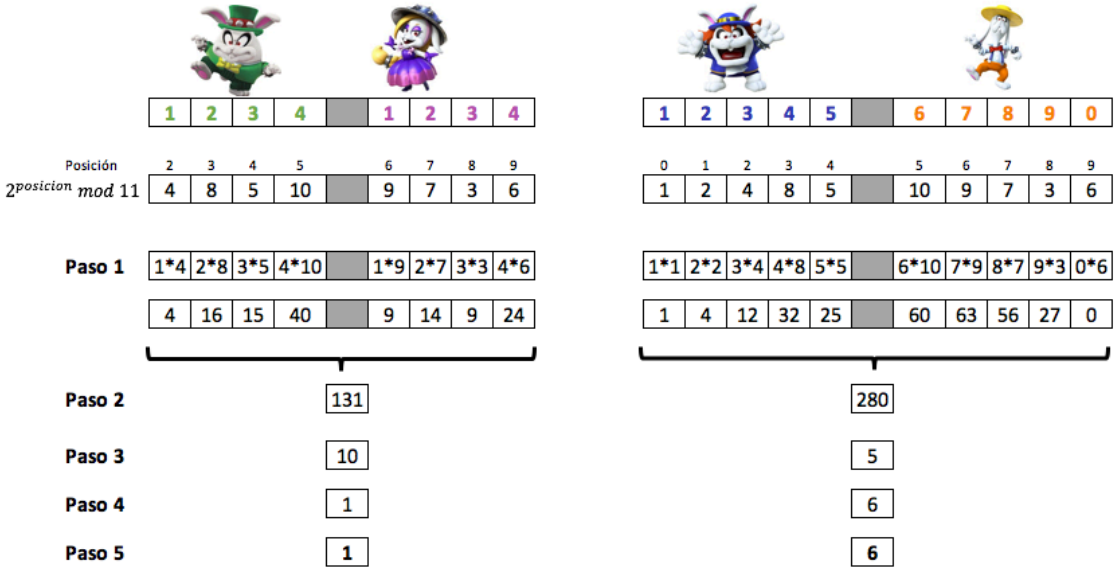
$$d_{1_T}d_{2_T}d_{3_T}d_{4_T}d_{1_H}d_{2_H}d_{3_H}d_{4_H}d_{1_S}d_{2_S}d_{3_S}d_{4_S}d_{5_S}d_{1_R}d_{2_R}d_{3_R}d_{4_R}d_{5_R}$$

Para poder desactivar la bomba, Cappy escuchó que hay que introducir dos dígitos. Rebuscando en la guarida de Bowser, Cappy encontró unos papeles donde parece que está escrita la forma de calcular los dos dígitos para desactivar la bomba.





A modo de ejemplo. Si se introducen los números *1234, 5678, 12345 y 67890* los dígitos de desactivación de la bomba serían 9 (primer dígito) y 6 (segundo dígito) y el proceso es el siguiente.



Se debe crear un programa que dados los cuatro códigos válidos. Los números se introducen por teclado, y deben ser los dos primeros un entero positivo con 4 dígitos y los dos siguientes un entero de 5 dígitos. Ninguno de los códigos puede comenzar por el dígito 0. Si cuando se introduce un código no es correcto se debe mostrar el mensaje ***"Código no valido"*** y se termina el ejercicio (hay que volver al menú de selección de ejercicios). La comprobación se realiza después de introducir cada uno de los códigos). Si los códigos son correctos el programa muestra en pantalla ***"Primer digito: X"*** y en la siguiente línea ***"Segundo digito: Y"*** siendo X e Y los valores calculados por el algoritmos creado por el Bowser para la desactivación de la bomba.

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

```
Topper, introduce el codigo de activacion (4 digitos)? 1234
Harriet, introduce el codigo de activacion (4 digitos)? 1234
Spewart, introduce el codigo de activacion (5 digitos)? 12345
Rango, introduce el codigo de activacion (5 digitos)? 67890

BOMBA activada ..........


Primer digito: 1
Segundo digito: 6
```

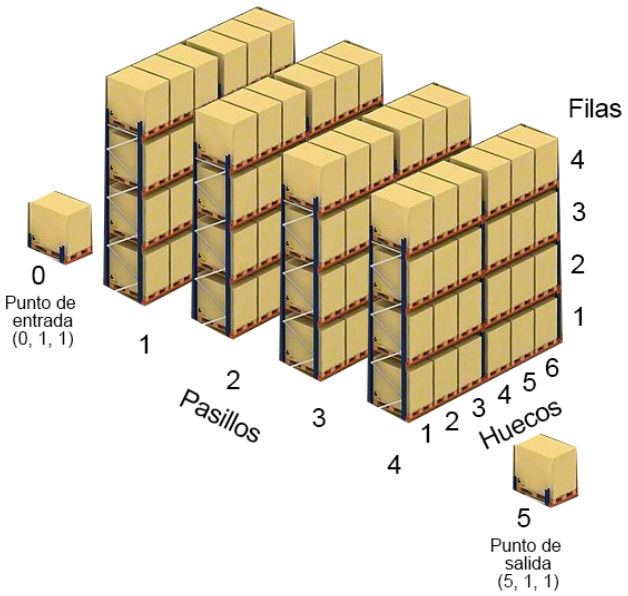
Para la realización de este ejercicio NO se puede hacer uso de array, ni se pueden utilizar 18 variables para almacenar cada uno de los dígitos de los códigos introducidos por los Broodals. Se penalizará la repetición de realizar el proceso 4 veces.

Parte 3: Crazy Cap

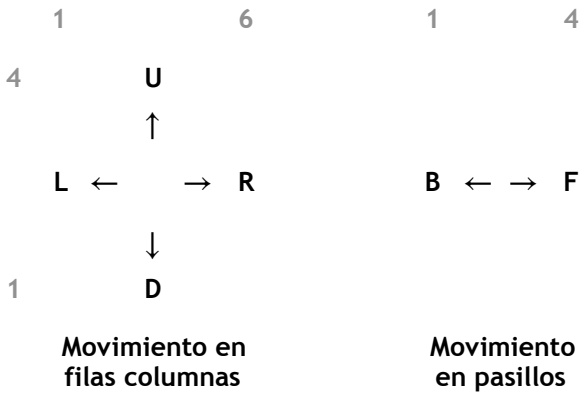
Crazy Cap es la tienda donde Mario puede adquirir nuevos trajes (y gorras) y todo tipo de objetos que le pueden ayudar en

el juego, o que le sirven para decorar la nave Odyssey. El propietario de la cadena de tiendas Crazy Cap quiere construir un almacén central automatizado para que la recepción (almacenaje) y expedición (des-almacenaje) de mercancía sea asistida realizado por un robot similar a R.O.B. pero sin armamento, y puede volar en vertical.

El nuevo almacén centralizado y robotizado del Crazy Cap se encuentra distribuido en pasillos, y cada pasillo se encuentra dividido en filas y huecos tal como se presenta en la siguiente figura, si bien el almacén puede tener cualquier número de pasillos, filas y huecos (al menos 1 de cada y se puede configurar).



R.O.B puede avanzar o retroceder en horizontal una unidad de movimiento (un pasillo o un hueco) y en vertical una unidad de movimiento (una fila). Para representar los movimientos de R.O.B se considera la información de la siguiente figura (lo números representan los pasillos, filas o huecos acorde a la distribución de la figura anterior).



En el almacén R.O.B puede realizar sólo dos operaciones, la de almacenar una mercancía (ir desde el lugar donde se encuentre al punto de entrada, coger la mercancía y llevarla al lugar indicado para su almacenaje), y la de sacar una mercancía del almacén (ir del lugar donde se encuentra la mercancía, recogerla y llevarla al punto de salida). Una vez realizada la operación R.O.B se queda esperando

la siguiente operación en el punto donde se detenga después de realizar la operación anterior.

R.O.B. para poder desplazarse entre los diferentes pasillos debe estar ubicado en la fila 1 y hueco 1 de un pasillo (si el punto de destino de R.O.B. se encuentra en el mismo pasillo en el que está parado R.O.B. esperando una orden, no hay que bajar a la fila 1 y hueco 1 para poder moverse). Una vez ubicado en el correspondiente pasillo, el robot primero se mueve en vertical para situarse en la fila y luego en horizontal para colocarse en el hueco indicado y almacenar o recoger la mercancía.

Este ejercicio debe permitir primero realizar la configuración del almacén, pidiendo por teclado el número de pasillos, filas y columnas que tendrá el almacén (valores todos ellos superiores a 0).

```
Numero de pasillos? 0
El numero de pasillos debes ser un número positivo mayor que 0
Numero de pasillos? 10
Filas de cada pasillo? 8
Numero de huecos de cada fila? 0
El numero de pasillos debes ser un número positivo mayor que 0
Numero de huecos de cada fila? 15
```

A continuación, una vez definido el almacén, el ejercicio debe permitir simular los movimiento de R.O.B a petición de los *Toad* que se encargan del almacén (la posición inicial de R.O.B. es el punto de entrada - pasillo 0, fila 1, hueco 1). Para hacer está simulación el programa debe mostrar las operaciones a realizar (1- recoger, 2 - almacenar, 3 - salir). Con la opción 3 se termina el ejercicio (hay que volver al menú de selección de ejercicios) y con las otras dos opciones el programa solicita el pasillo, la fila y el hueco (números positivos mayores a 0 y válidos en la configuración del almacén) de la mercancía que va a recoger o a almacenar. Si los datos introducidos no son válidos se informa al *Toad* y se vuelve a pedir el dato no válido.

```
Operación a realizar
[1] Recoger
[2] Almacenar
[3] Salir
Opcion? 1
Donde coger?
Pasillo? -1
Pasillo no valido
Pasillo? 12
Pasillo no valido
Pasillo? 3
Fila?-1
Fila no valida
Fila?10
Fila no valida
Fila?3
Hueco?-1
Hueco no valido
Hueco?0
Hueco no valido
Hueco?16
Hueco no valido
Hueco?15
```

```
Operación a realizar
[1] Recoger
[2] Almacenar
[3] Salir
Opcion?
2
Donde almacenar?
Pasillo? 3
Fila?3
Hueco?4
```

Si los datos son correctos se procede a simular el movimiento de R.O.B. El ejercicio debe mostrar cada uno de los movimientos realizados por R.O.B. y las posiciones donde se encuentra en cada momento. El movimiento de R.O.B. se describe a continuación en función del tipo de movimiento.

- [1] Recoger. R.O.B. se mueve desde la posición donde se encuentre parado hasta la fila 1 y hueco 1 del pasillo donde se encuentra*. A continuación se dirige al pasillo indicado para la recogida, y a la fila y hueco indicados, mostrando el mensaje "Recogida de material". Por último, se dirige al punto de entrega y muestra el mensaje "Entrega de material". R.O.B. se queda parado en este punto hasta que este listo para atender la siguiente operación.

- [2] Almacenar. R.O.B. se mueve desde la posición donde se encuentre parado hasta la fila 1 y hueco 1 del pasillo donde se encuentra†. Luego se dirige al punto de entrada para coger el material que debe almacenar y muestra el mensaje "Recogida de material". Para terminar se mueve al punto de almacenaje indicado por el *Toad* y muestra el mensaje "Almacenado de material". R.O.B se queda parado en este punto hasta que este listo para atender la siguiente operación.

```
Operación a realizar
[1] Recoger
[2] Almacenar
[3] Salir
Opcion? 1
Donde coger?
Pasillo? 3
Fila?2
Hueco?4
Posicion Robot - pasillo: 0, fila: 1, hueco: 1
mov     pasillo         fila            hueco
=======================================
-       0               1               1
F       1               1               1
F       2               1               1
F       3               1               1
U       3               2               1
R       3               2               2
R       3               2               3
R       3               2               4
Recogida de material
D       3               1               4
L       3               1               3
L       3               1               2
L       3               1               1
F       4               1               1
F       5               1               1
F       6               1               1
F       7               1               1
F       8               1               1
F       9               1               1
F       10              1               1
F       11              1               1
Entrega de material
```

```
Operación a realizar
[1] Recoger
[2] Almacenar
[3] Salir
Opcion? 2
Donde almacenar?
Pasillo? 4
Fila?3
Hueco?1
Posicion Robot - pasillo: 11, fila: 1, hueco: 1
mov     pasillo         fila            hueco
=======================================
B       10              1               1
B       9               1               1
B       8               1               1
B       7               1               1
B       6               1               1
B       5               1               1
B       4               1               1
B       3               1               1
B       2               1               1
B       1               1               1
B       0               1               1
Recogida de material
F       1               1               1
F       2               1               1
F       3               1               1
F       4               1               1
U       4               2               1
U       4               3               1
Almacenado de material
```

* *Si R.O.B. se encuentra en el pasillo de destino, no es necesario ir a la fila 1 y hueco 1 para moverse. Irá directamente a la fila y hueco indicados como destino*

† *Si R.O.B. se encuentra en el pasillo de destino, no es necesario ir a la fila 1 y hueco 1 para moverse. Irá directamente a la fila y hueco indicados como destino*

PRÁCTICA 1: ESTRUCTURAS BÁSICAS DEL LENGUAJE

Entrega

Esta práctica es de realización obligatoria para todos los alumnos. **Se debe realizar de forma individual**.

Plazo de entrega: **23:59 horas del día 11 de noviembre de 2020**.

Habrá que entregar a través de Moodle un fichero de nombre **practica1.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 errónea una letra.

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

Introduzca una opcion (1-4)
```

- 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:
  - **void ejercicio1();**
  - **void ejercicio2();**
  - **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.**

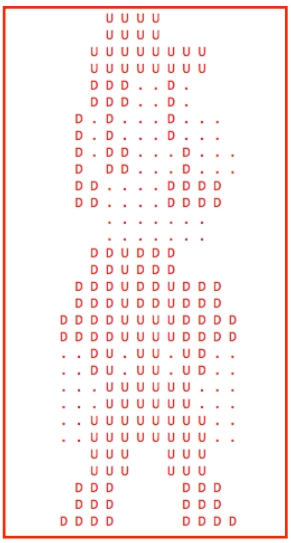
Puntuación

Esta práctica de entrega obligatoria se puntuará de 0 a 10, nota que se contabilizará en el 30% de la nota final de la asignatura en el caso de la evaluación continua (dentro de ese 30% esta práctica cuenta el 50%. Para poder considerar la nota de la práctica se debe obtener una nota igual o superior a 5). 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 de errores de compilación se considerará suspensa directamente.
- Correcto funcionamiento de los distintos procedimientos entregados. Es decir, se valorará positivamente que no haya errores lógicos.
- Limpieza y claridad en el código.
- Presencia de comentarios en el código.
- Errores descritos en el documento "*Errores a evitar.pdf*" que se encuentra en Moodle junto con el enunciado de la práctica.

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

Cotorrete (parte 1)	3
Broodals (parte 2)	3
Crazy Cap (parte 3)	4
[EXTRA y OPCIONAL] Dibujar la Mario Bros con letras	0.5
Total	10

La puntuación máxima a obtener puede ser un 10.5 si se realizan todos los apartados obligatorios y opcionales de la práctica.