**1º Capicúa.**

Realiza un **programa modular** que pida un número por teclado (el número no puede ser superior a 5 cifras, se debe controlar esto).

Después, dicho programa, determinará si el número es capicúa, es decir, se lee igual de

realizar algún **módulo que invierta** el número.

**Ejemplos de ejecución:**

Dame un número: 2345

No es capicúa.

-------------------------------

Dame un número: 19891

Sí es capicúa.

-------------------------------

Dame un número: 234678

Número incorrecto.

Dame un número: 1221

Sí es capicúa

***(3,25 puntos)***

**2º Dados**

Vamos a crear un **juego de dados.** El juego consiste en lo siguiente:

El **jugador** parte con una cantidad inicial que se introduce por teclado (en euros), la **cantidad mínima** será de 20 €.

El **ordenador** lanzará **tres dados** y el **jugador** antes debe tratar de adivinar cuántos 1, 2, 3, 4, 5 y 6 saldrán. Hará una **apuesta** (que no podrá ser superior al dinero que le queda, lógicamente) y si acierta uno, se lleva esa misma cantidad, si acierta 2, pues 2 veces la cantidad y si acierta 3, 3 veces la cantidad. Si no acierta ninguno pierde la apuesta.

**Por ejemplo**, tengo 20 € y apuesto 10 € a que salen: 2 1’s; 0 2’s; 1 3; 0 4’s, 0 5’s y 0 6’s (1 1 3). Si los dados salieran:

* 4 3 2 habré ganado 10 € por el 3.
* 6 2 5 habré perdido 10 €.
* 1 4 3 habré ganado 10 € por el 1 y otros 10 € por el 3.
* 1 3 1 habré ganado 30 € (los 2 1’s y el 3 que he apostado => 3 veces la apuesta).

***(3,5 puntos)***

**3º F-1000.(EL GAAAAAANCHO)**

Trabajamos para una empresa de tratamiento profesional de productos generados naturalmente; ricos en vitaminas, minerales, antioxidantes y fibra y que aportan pocas calorías y un alto porcentaje de agua… vamos, en una frutería.

Esta frutería se ha modernizado y tiene un brazo robótico, el F-1000, que coge las cajas de fruta de una estantería y las apila en un palé.

Los módulos que me permiten manejar el brazo son los siguientes:

Procedimiento: **Recibir\_pedido((ref) cantidad)** 🡪 nos devolverá, por referencia, la cantidad de cajas que tenemos que coger de la estantería y apilar en el palé. En *cantidad* se devuelve un 0 si no hay ningún pedido.

Función: **Detectar\_caja(fila, columna)** 🡪 mueve el brazo a esa posición y devuelve ***verdadero*** si en esa posición existe una caja y ***falso*** en caso contrario.

Procedimiento: **Coger(fila, columna)** 🡪 Coge la caja que haya en esa posición de la estantería entre sus pinzas.

Procedimiento: **Apilar(posición)** 🡪 Mueve el brazo y apila la caja que tiene en las pinzas en el palé en la posición que le indiquemos.

Procedimiento: **Posicion\_Inicial()** 🡪 Coloca el brazo robótico en su posición de reposo inicial.

En la estantería existen 150 cajas de frutas distribuidas en 10 filas por 15 columnas.

Cuando **iniciamos el sistema** la estantería está llena de cajas, por lo que el brazo podrá acceder a la caja que está en la fila 1, columna 1 (que será la que está la primera de la primera columna); la forma de ir cogiendo cajas será: **agotar las cajas de una columna antes de pasar a la siguiente columna**. Antes de coger la caja se debe comprobar si la caja existe en esa posición. En caso de no existir el programa acabará, se mostrará un mensaje indicando que el brazo pasa a estar inoperativo y que se ha producido un error.

Si **no hay errores**, y una vez servido el pedido que hayamos recibido, el robot volverá a la **posición inicial** y procederá a comprobar si tiene más pedidos. Cuando reciba otro pedido se procede a repetir la operación de aplicarlo cómo se ha explicado anteriormente, con las mismas condiciones en caso de error.

El **programa acabará** cuando no queden cajas en la estantería. Antes de acabar, el programa colocará el brazo en su posición inicial, se mostrará un mensaje indicando que el brazo pasa a estar inoperativo y se mostrará, también, si el último pedido se pudo satisfacer o no. ***(3,25 puntos)***