

Práctica 3:

Uso de estructuras de control.

Objetivos:

Creación, edición y compilación de un programa completo.

Elección y uso adecuado de las estructuras de control.

Identificación de los tipos adecuados de variables y constantes.

Aplicación de las normas de estilo en los nombres de identificadores y resto del programa.

Desarrollo:

1 Calcular los saltos del canguro

De cara a ayudar a la comunidad científica en el estudio del comportamiento de los canguros, vamos a crear un programa que simule el comportamiento del mismo ante un plato de comida.

Para ello vamos a suponer un canguro encerrado en un recinto circular de R metros de radio R.

El canguro inicialmente estará situado en una determinada posición aleatoria (x,y) dentro del recinto y desea llegar a un plato de apetitosa comida (de canguros).

La posición del plato de comida variará de una ejecución a otra y deberá indicarlo el científico antes de empezar la simulación

Además, debemos tener en cuenta que:

- Como el recinto es cerrado, el canguro nunca se podrá escapar.
- Que los saltos de un canguro son de 1 m de longitud.
- El canguro tiene la particularidad, de que cuando salta no es capaz de controlar la dirección. Esto se simulará siendo el usuario del programa el que controle la dirección de los movimientos mediante el uso de las siguientes teclas: N se dirige hacia el frente; S se dirige hacia atrás; E se dirige hacia la derecha; O se dirige hacia la izquierda.
- Cuando el canguro en uno de sus saltos se tope con la valla del recinto emitirá un ligero gruñido y volverá a la posición de la que partía.

De cara a realizar el programa ten en cuenta los siguientes detalles

- La posiciones, tanto del canguro como del plato, vendrán dadas por dos valores (x e y).
- Para comprobar si la posición del canguro está dentro del recinto o por el contrario está intentando salirse del recinto debemos utilizar el teorema de Pitágoras.
 $x^2+y^2=R^2$
- Para generar números aleatorios utilizaremos la función `int rand(void)` devuelve un número aleatorio entre 0 y `RAND_MAX` .

Es necesario utilizar la función `srand((unsigned)time(NULL))` para inicializar el proceso de generación de números aleatorios.

Para utilizar dichas funciones es necesario incluir las librerías `stdlib.h` y `time.h`.

Se usaría como sigue:

```
#include <stdlib.h>
#include <time.h>

void main (){
    srand( (unsigned)time( NULL ));
```