



Laboratorio 6: Subprogramas

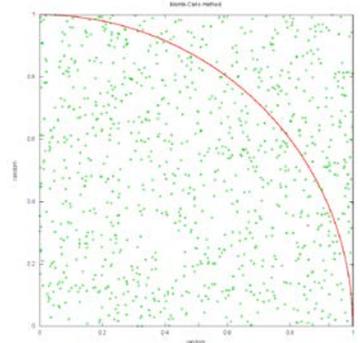
Ejercicio 1.- Aproximación de π

El método de Montecarlo consiste en realizar un experimento aleatorio un determinado número de veces. Como sabemos que la frecuencia con que ocurre un suceso se acerca a su probabilidad, a medida que aumenta el número de ensayos nos iremos acercando más y más al valor buscado. Puede utilizarse para estimar probabilidades que serían muy difíciles de calcular de forma teórica, o para corroborar que ocurrirá lo que nosotros esperamos de un determinado experimento.

Consideramos un cuadrado de lado 1 y el cuadrante de círculo de la siguiente figura:

Supongamos que lanzamos un dardo al azar sabiendo con seguridad que va a dar en el cuadrado, pero no necesariamente en el cuadrante del círculo. La probabilidad de que el dardo caiga en el cuadrante es la proporción que hay entre las áreas de las dos figuras. Más concretamente:

$$P = \frac{\text{Area del cuadrante}}{\text{Area del cuadrado}} = \frac{\pi/4}{1} = \frac{\pi}{4}$$



Suponemos ahora que lanzamos N dardos sobre el cuadrado y que M de ellos dan en el cuadrante. La *frecuencia relativa* del suceso "dar en el cuadrante" es:

$$f = \frac{M}{N}$$

Si el número de lanzamientos es muy grande, la frecuencia relativa se aproximará a la probabilidad P , luego para N grande tenemos:

$$f = \frac{M}{N} = \frac{\pi}{4}$$

De aquí podemos despejar π y nos queda:

$$\pi = \frac{4M}{N}$$

El valor aproximado de π es igual al cuádruplo del número de dardos que han caído en el cuadrante de círculo, dividido por el número de lanzamientos. Para ello obtenemos dos números aleatorios x e y , comprendidos entre 0 y 1 (para que el dardo caiga en el cuadrado con seguridad), y suponemos que el dardo ha ido a parar al punto (x,y) . El punto está dentro del cuadrante si:

$$x^2 + y^2 < 1$$

Escribe un subprograma para hallar el valor aproximado de π lanzando el dardo un número de veces pedido por el input.

Ejercicio 2.- Lanzar dados

Se trata de implementar un subprograma que sea capaz de simular el lanzamiento de dos dados y mida la frecuencia relativa de aparición de un cierto valor a lo largo de un determinado número de tiradas. El valor cuya frecuencia se desea medir y el número de tiradas a realizar se introducen como parámetros. Implementar un programa principal que permita al usuario decidir después de cada experimento (grupo de tiradas) si desea o no realizar otro.

Ejercicio 3.- Piedra, papel y tijeras

Este es un juego muy antiguo en el que intervienen dos personas. Cada jugador hace su elección entre las tres alternativas existentes (piedra, papel o tijeras) y el ganador se determina atendiendo a las siguientes reglas:

1. La piedra gana a las tijeras.
2. Las tijeras ganan al papel
3. El papel gana a la piedra.
4. Si las dos elecciones son iguales se produce un empate

Escribe un programa que permita a la máquina jugar contra un humano. Máquina y humano harán su elección y la máquina será la que decida quién gana. Implementa todos los subprogramas necesarios.

Ejercicio 4.- Aproximación del coseno

Escribir un programa que realice una aproximación del coseno, con la siguiente fórmula:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

El programa se realizará hasta que se haya acumulado un término menor que una diezmilésima.

En este programa se realizarán las siguientes funciones:

- Función **gradosRadianes**, función que dado un entero que corresponde a los grados, nos devuelve los radianes correspondientes.
- Función **potencia**, que dado un real correspondiente a la base y un entero que corresponde al exponente, devuelva la potencia
- Función **factorial**, dado un entero positivo nos devuelva el factorial del número
- Función **coseno**, dado un x en radianes nos devuelva el coseno de ese número

NO SE PUEDE UTILIZAR LA BIBLIOTECA CMATH

Ejercicio 5.- Reloj digital

8:59:59

Expresar un tiempo del día en la forma de horas, minutos y segundos, teniendo en cuenta que las horas van de 0 a 23, y los minutos y segundos de 0 a 59, siendo todos enteros.

Se pide :

- Un subprograma **LeerTiempo** que lea del input un tiempo del día, compruebe que es correcto y lo devuelva a quien lo llamó. Si el tiempo no es correcto volverá a pedirlo.

- Un subprograma **SegundoSiguiente** que reciba un tiempo del día y devuelva el tiempo obtenido al transcurrir un segundo. Recuerde los límites de los tiempos, por ejemplo, el tiempo que sigue a 8:59:59 es 9:00:00
 - Un programa **RelojDigital** que pida el tiempo del día actual, con **LeerTiempo** y muestre por pantalla el tiempo actualizado al segundo, utilizando **SegundoSiguiente**, y que se detenga transcurrido un día, es decir 86400 segundos. No se pretende que el reloj sea preciso, pero sí que su salida recuerde la de un reloj digital, completando con un cero cuando los minutos y segundos tengan un solo dígito, como por ejemplo en 2:03:00 que corresponde a las dos y tres minutos de la madrugada.
- Escribe un programa que dada una fecha de nacimiento nos devuelva el número de la suerte.