SEMESTRE 1 – 2010 21 AL 23 JULIO DE 2010

PRÁCTICA DE ELIMINACIÓN

ESCENARIO: Venezuela bajo cero.

Si lanzamos un dado, sabemos que cada cara tiene una probabilidad de 1 en 6 de salir (son 6 números y va a salir uno de ellos: 1, 2, 3, 4, 5 ó 6). Si lanzamos el dado una sola vez, la probabilidad es 1/6 o 16,7% de ser el número ganador.

Si lanzamos el dado 10 veces, no siempre van a cumplirse estas probabilidades. Saldrá, por ejemplo, 3 veces el 5, 2 veces el 6, 2 veces el 2, 1 vez el 1, 1 vez el 4 y otra vez el 3. Ahora bien, si tenemos la paciencia para lanzar el dado 100 veces, es más probable que se vaya cumpliendo el pronóstico teórico de que cada cara del dado va a aparecer cerca de 17 veces en 100 lanzamientos consecutivos (aquella probabilidad de 16,7% que mencionamos).

Si hacemos 1000 lanzamientos, cada vez más cerca del valor teórico estaremos. Y así por delante. En la medida que el número de lanzamientos aumente, estaremos más próximos a la probabilidad teórica y los eventos que escapen a ella serán los menos frecuentes. Diremos que cada lanzamiento independiente tiende a seguir una "distribución estadística normal".

Enunciado

Dado un valor de **N**, leído desde teclado, desarrolle un programa o aplicación que simule **N** lanzamientos de un dado. El programa debe almacenar cada lanzamiento en un vector **v[N]** en forma ascendente y debe poderse ejecutar para valores de **N** entre 10 a 1000 e imprima en el archivo de dato Probabilidad.dat cada uno de los **N** números simulados, además debe calcular el porcentaje de veces que sale cada una de las seis caras del dado.

Para Simular el lanzamiento de un dado se debe generar números aleatorios, para lograrlo los lenguajes de programacion cuentan con procedimientos o funciones que realizan esta actividad donde:

Pascal	Visual Basic
Procedimiento de generación de la semilla y debe colocarse dentro del programa principal. **Randomize;** {cambia entre ejecución el valor aleatorio} *Generador del número aleatorio:* **Random();** {genera un número aleatorio entre 0 y 1} **Random(Valor);** {Genera un número aleatorio entre 0 y Valor-1 y valor debe ser un dato entero} **Para generar un número entero aleatorio en un rango (A,B) se puede utilizar la expresión:** Valor := **random (B – A + 1) + A;**	*Procedimiento de generación de la semilla* **Randomize()** ' Cambia el valor aleatorio entre ejecuciones *Generador del número aleatorio:* **Rnd()** ' Genera un número aleatorio entre 0 y 1 ' como un dato tipo single **Si se quiere que el numero aleatorio sea un dato entero se debe usar la función Int** **Int(Valor)** ' Retorna la parte entera de un numero, valor debe ser un ' dato numérico **Para generar un número entero aleatorio en un rango (A,B) se puede utilizar la expresión:** Valor = ***Int(Rnd () * (B – A + 1))*** **+ A**

CONSIDERACIONES

Para la solución del problema debe definir y utilizar:

1. Un subprograma que reciba tres valores enteros positivos A, B y N, retorne **TRUE** si **A ≤ N ≤ B**; caso contrario retorne **FALSE**

2. Un subprograma que lea un valor entero **N** que esté en el rango [10, 1000], utilizando el subprograma anterior. El subprograma debe seguir leyendo N hasta que **10 ≤ N ≤ 1000**. Si el valor de N leído no esta en el rango [10,1000], imprima un mensaje que indique el error por teclado.
3. Un subprograma que dado un vector de elementos enteros inserte en la posición K un valor.
4. Un subprograma que dado un valor **N**, retorne un arreglo v[N] tipo **Vector**, donde cada elemento sea un número aleatorio en el rango [1, 6].
5. Un subprograma que imprima un arreglo v[N] tipo **Vector** hacia un archivo, en forma de vector fila.

NOTA: para ubicar la posición donde se tiene que insertar el valor generado se dispone del subprograma ***Posicion*** el cual se muestra a continuación:

Pascal

```
Function Posicion( V : vector;
            N, Valor : Integer) : Integer;
Var
   I : Integer;
Begin
   I:=1;
   While (I<=N) And (Valor>V[I]) do
      I:=I+1;
   Posicion:=I;
End;
```

Visual Basic

```
Function Posicion(ByVal V() As Integer, _
        ByVal N As Integer, _
        ByVal Valor As Integer) As Integer

   Dim I As Integer = 1
   While I<=N And Valor>V(I)
      I += 1
   End While
   Posicion = I

End Function
```