

SEMESTRE 1 – 2010

21 AL 23 JULIO DE 2010

PRÁCTICA DE ELIMINACIÓN**ESCENARIO : Venezuela bajo cero.**

Para las próximas Olimpiadas de Invierno, Venezuela piensa cumplir el mismo sueño que una vez tuvo Jamaica al clasificar en la competencia de trineo. Según las normas establecidas por el Comité Olímpico para que un país clasifique en la categoría de trineo en las próximas olimpiadas, debe realizar el recorrido de la pista principal de Alemania en un tiempo inferior a 1' 57" 000.



Para determinar qué países clasificarán, cada equipo tiene dos intentos en realizar dicho recorrido en un tiempo inferior al establecido, escogiendo como tiempo de clasificación el menor realizado por el mismo. En ese sentido, se almacenó en el archivo de datos "COMPETENCIA.DAT", la siguiente información en la primera línea el tiempo mínimo de clasificación y en las líneas siguientes para cada país que intenta clasificar en dicha competencia: nombre del país y los dos tiempos realizados por el equipo..

Enunciado

Elabore un programa o aplicación que dada la información contenida en el archivo "COMPETENCIA.DAT", genere el archivo de datos "CLASIFICADOS.DAT", el cual debe contener, el nombre de todos los países clasificados y el tiempo con el cual clasificó (Menor tiempo realizado).

COMPETENCIA.DAT

1	57	000					
Venezuela	1	59	032	1	56	987	
EEUU	1	58	125	1	56	879	
Canada	1	57	156	1	57	158	
Rusia	1	56	894	1	59	486	
Jamaica	1	59	458	1	57	456	
Japon	1	58	000	1	56	587	
Alemania	1	57	145	1	56	798	
Suiza	1	56	987	1	58	058	
Suecia	1	57	156	1	57	456	
Inglaterra	1	58	415	1	56	869	

CLASIFICADOS.DAT

Venezuela	1	56	987
EEUU	1	56	879
Rusia	1	56	894
Japon	1	56	587
Alemania	1	56	798
Suiza	1	56	987
Inglaterra	1	56	869

REQUERIMIENTOS

1. Elabore un tipo de dato registro de nombre **Tiempo**, el cual debe contener un tiempo expresado en minutos, segundos y centésimas de segundo.
2. Defina un dato tipo arreglo (**vectorTiempo**), el cual puede contener hasta 20 elementos tipo **Tiempo**.
3. Defina un dato tipo arreglo (**vectorEquipo**), el cual puede contener hasta 20 elementos tipo **cadena**.
4. Elabore un subprograma que convierta un **Tiempo** expresado en minutos, segundos y centésimas de segundos, y transforme el tiempo en centésimas de segundos totales.
5. Elabore un subprograma que dado dos datos tipo tiempos expresados en minutos, segundos y centésimas de segundos, devuelva el tiempo menor.
6. Elabore un subprograma que dado dos vectores (tipo **vectorTiempo**) de **N** elementos cada uno, genere un tercer vector (tipo **vectorTiempo**), que contendrá en cada elemento el tiempo menor que existe entre los dos vectores (tipo **vectorTiempo**) en la misma posición.
7. Elabore un subprograma que dado dos vectores en paralelo (uno tipo **vectorEquipo** y otro tipo **VectorTiempo**) y una posición **K**, elimine de los arreglos el dato que se encuentra en la posición **K**.
8. Elabore un subprograma que dado dos arreglos en paralelo (uno del tipo **vectorEquipo** y el otro tipo **vectorTiempo**) de **N** elementos, y un dato tipo **Tiempo T**, elimine todos los equipos cuyo tiempo de clasificación sea mayor o igual al **Tiempo T**.

CONSIDERACIONES

1 minuto equivale a 60 segundos
1 segundo equivale a 1000 centésimas de segundo

SEMESTRE 1 – 2010

14 AL 16 FEBRERO DE 2011

PRÁCTICA DE INSERCIÓN**ESCENARIO : Estadística de lanzamiento.**

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 una aplicación en VB2008 que simule **N** lanzamientos de un dado. El programa debe almacenar cada lanzamiento en un vector **v[N]** el cual a medida que vayan saliendo debe quedar en forma ascendente y además tiene que ejecutarse para valores de **N** entre 10 a 1000 e imprima en el archivo de dato Probabilidad.Txt cada uno de los **N** números simulados, finalmente debe determinar 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 programación cuentan con procedimientos o funciones que realizan esta actividad donde:

Recuerde:**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 número aleatorio sea un dato entero se debe usar la función **Int**

Int(Valor) ‘ Retorna la parte entera de un número, valor debe ser un ‘ dato numérico

Para generar un número entero aleatorio en un rango (A,B) se puede utilizar la expresión:

$$\text{Valor} = \text{Int}(\text{Rnd}()) * (B - A + 1) + A$$

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 $A \leq N \leq B$; 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 $10 \leq N \leq 1000$. Si el valor de N leído no está 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 **Entero**, donde cada elemento sea un número aleatorio en el rango [1, 6].
5. Un subprograma que imprima un arreglo v[N] tipo **Entero** 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:

```
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
```