

Semestre Único – 2013

Semana 25/11 al 29/11 de 2013

Práctica Nro. 7: Inserción y eliminación de elementos en arreglos

Objetivo: Desarrollar una aplicación tipo consola en VB2010, en donde se utilicen datos tipo arreglos unidimensionales y/o bidimensionales, los cuales comparten los datos de un registro, con el propósito de insertar y/o eliminar registros dentro de él.

Introducción

Una de las actividades más comunes a emplear con el uso de arreglos es la versatilidad con que se puede manejar los registros, con el fin de realizar actividades como el insertar o eliminar registros en una posición cualquiera dentro de ellos. Para lograr realizar el proceso de eliminación o inserción de datos en un arreglo, se logra subiendo los datos o bajando los datos que se encuentran en el arreglo a partir de la posición a trabajar según sea el caso, además hay que recordar que luego de realizada la actividad, se debe actualizar el tamaño del arreglo ya que este aumenta o disminuye en uno.

En caso particular de insertar datos, hay que tener en cuenta que en los arreglos del tipo bidimensional, se puede insertar una fila o una columna, lo que significa que la cantidad de datos a insertar debe ser igual al número de elementos de la fila o columna a insertar y estos datos generalmente se encuentran almacenados en un vector.

A continuación se muestra los subprogramas acordes a realizar estos procesos:

*Inserción de un elemento en un vector de cualquier tipo en la posición **k** del Arreglo:*

```
Sub InsertarDato(ByRef V() As Tipo, ByRef Nelementos As Integer, ByVal k As Integer, ByVal Dato As Tipo)
    ' Desplazamiento de los elementos del vector
    For I = Nelementos To k Step -1
        V( I + 1 ) = V( I )
    Next
    ' Inserción del dato en la posición k
    V( k ) = Dato
    ' Actualización del tamaño del arreglo
    Nelementos += 1
End Sub
```

*Inserción en la fila **k** del Arreglo bidimensional **Matriz** de **Nfilas** x **Ncolumnas** elementos de cualquier tipo, de los elementos almacenados en el vector **V**:*

```
Sub InsertarFila(ByRef Matriz( , ) As Tipo, ByRef Nfilas As Integer, ByVal Ncolumnas As Integer,
    ByVal k As Integer, ByVal V() As Tipo)
    ' Desplazamiento de los elementos de la matriz
    For I = Nfilas To k Step -1
        For J = 1 To Ncolumnas
            Matriz( I + 1 , J ) = Matriz( I , J )
        Next
    Next
    ' Inserción del dato a la fila k
    For I = 1 To Ncolumnas
        Matriz( k , I ) = V( I )
    Next
    ' Actualización del tamaño del arreglo
    Nfilas += 1
End Sub
```

NOTA: Cuando la inserción se realiza en arreglos en paralelos, lo que ocurre es que los recorridos se solapan para funcionar en paralelo.

*Eliminación del elemento **k** de un vector de cualquier tipo:*

```
Sub EliminarDato(ByRef V() As Tipo, ByRef Nelementos As Integer, ByVal k As Integer)
    ' Desplazamiento de los elementos del vector
    For I = k To Nelementos -1
        V( I ) = V( I + 1 )
    Next
    ' Actualización del tamaño del arreglo
    Nelementos -= 1
End Sub
```

*Eliminación de la columna **k** del Arreglo bidimensional **Matriz** de **Nfilas** x **Ncolumnas** elementos de cualquier tipo:*

```
Sub EliminarColumna(ByRef Matriz( , ) As Tipo, ByRef Nfilas As Integer, ByVal Ncolumnas As Integer,
    ByVal k As Integer, ByVal V() As Tipo)
    ' Desplazamiento de los elementos de la matriz
    For I = 1 To Nfilas
        For J = k To Ncolumnas - 1
            Matriz( I , J ) = Matriz( I , J + 1 )
        Next
    Next
```

```
Next
Next
' Actualizacion del tamaño del arreglo
Ncolumnas -= 1
End Sub
```

NOTA: Cuando la eliminación se realiza en arreglos en paralelos, lo que ocurre es que los recorridos se solapan para funcionar en paralelo.

Ejercicio Nro 1: Probabilidad con los dados

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 16,67% de ser el número ganador.

Si lanzamos el dado 10 veces, no siempre van a cumplirse estas probabilidades. Saldrá, por ejemplo, 3 veces el Cinco, 2 veces el seis, 2 veces el dos, 1 vez el uno, 1 vez el cuatro y otra vez el tres. 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 16,67% que mencionamos).

Si hacemos 1000 lanzamientos, cada vez más cerca del valor teórico estaremos, donde 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 bajo consola en VB2010 que simule **N** lanzamientos de un dado. El programa debe almacenar cada lanzamiento en un vector **v[N]** colocando los diferentes valores obtenidos en el arreglo en forma ascendente. Además, los valores de **N** posibles a emplear deben encontrarse entre 10 a 1000. Finalmente el programa debe imprimir en el archivo de dato “**Probabilidad.Txt**” cada uno de los **N** números simulados (Ordenados en forma ascendente), y posteriormente debe mostrar 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 para el caso de VB2010 son:

Procedimiento de generación de la semilla

Randomize() ‘ Cambia el valor aleatorio entre ejecuciones, debe colocarse este llamado dentro de Sub Main, una sola vez y antes de emplear la función Rnd()

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 numero, valor debe ser un ‘ dato numérico

Para generar un número entero aleatorio en el rango (A,B) se utiliza la siguiente expresión:

Valor = Int(Rnd () * (B - A + 1)) + A

CONSIDERACIONES

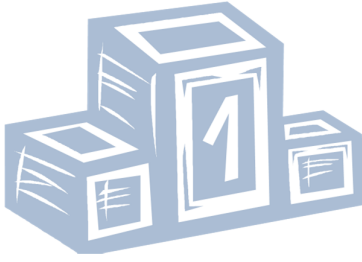
Para la solución del problema debe definir y utilizar:

1. Un subprograma que reciba tres valores enteros positivos A, B y N, y 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 se cumpla 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 pantalla.
3. Un subprograma que dado un vector de elementos enteros, inserte en la posición **K** un valor **T**.
4. Un subprograma que dado un valor **N**, retorne un arreglo v[N] tipo **Entero**, donde cada elemento contenido sea un número aleatorio en el rango [1, 6].
5. Un subprograma que imprima un arreglo v[N] tipo **Entero** hacia el archivo **Narch** (Número del archivo), en forma de vector fila, y los datos ordenados en forma ascendente.

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
    Posicion = 1
    While Posicion <= N And Valor > V( I )
        Posicion += 1
    End While
End Function
```

Ejercicio Nro 2: The Amazing Race



The amazing race es una competencia de Equipos integrados por dos a cuatro miembros que corren alrededor del mundo para lograr completar los retos establecidos más rápido que sus contrincantes. Durante la carrera se llevan indicadores de ruta que se registran cada vez que un equipo logra culminar un reto, permitiendo así estimar el avance de cada uno. El indicador puede tomar dos valores: 1 si el reto fue logrado y 0 si el reto no ha sido alcanzado, es decir no logró vencer un reto, y si esto pasa el resto de los siguientes retos se consideran no alcanzados.

PROBLEMA

Dado un el archivo de datos “**Avances.Txt**” que almacena en cada línea la identificación del Equipo, y luego un conjunto de indicadores de avance durante la carrera, desarrolle una aplicación bajo consola en VB2010 que determine para cada grupo la cantidad de retos culminados, con el fin de generar el archivo “**Resultados.Txt**” que contenga los tres equipos que van más avanzados en la competencia.

Ejemplo:

Avances.Txt		Resultados.Txt	
Los invencibles	1 1 1 1 0 0 0	Los multizona	7 retos alcanzados
Vista de aguilá	1 1 1 1 1 1 0	Vista de aguilá	6 retos alcanzados
Los multizona	1 1 1 1 1 1 1	Los speedracer	5 retos alcanzados
Fuerza aérea	1 0 0 0 0 0 0		
Los speedracer	1 1 1 1 1 0 0		
Los Bachacos	1 1 1 0 0 0 0		

REQUERIMIENTOS

Además de los subprogramas que usted necesite diseñar para dar solución al problema, su programa debe contener los siguientes subprogramas (los utilice o no en el programa principal u otros subprogramas):

1. Un subprograma que dada una matriz **m** de **fm×cm** elementos tipo integer, retorne como resultado un vector paralelo con la cantidad de unos que hay por cada fila de la matriz **m**.
2. Un subprograma que dado un vector **Q** retorne la posición de ubicación del mayor elemento del arreglo.
3. Un subprograma que elimine el elemento ubicado en la posición k de un vector **Q**.
4. Un subprograma que elimine la fila **k** de una matriz **m** de **fm×cm** elementos tipo integer.
5. Un subprograma que haciendo uso de los subprogramas anteriores obtenga un vector donde se muestre los tres equipos que van más adelantados en la competencia en forma descendente.