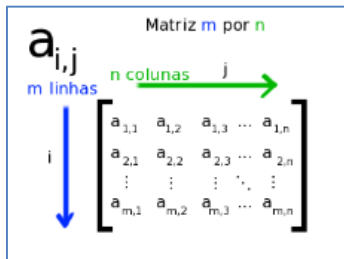


Practica N°5 – Arreglos (Vectores)

Semana del 25 de Mayo al 30 de Mayo del 2015

Objetivo de la Práctica: Desarrollar una aplicación tipo consola en VB2010, en donde se utilice datos tipo arreglos unidimensionales y bidimensionales, buscando consolidar la forma de recorrerlos con el propósito de determinar cálculos del tipo estadísticos (promedios, porcentajes, máximos y mínimos).

INTRODUCCIÓN



Los arreglos es un tipo de dato compuesto capaz de almacenar un conjunto de datos finitos en un solo identificador, siendo esta cantidad conocida como el tamaño del arreglo. Todos estos datos almacenados en el arreglo son del mismo tipo ya que es una estructura homogénea y estos datos representa un mismo concepto a manejar, los cuales se almacenan en la maquina en espacios de memoria continuos, estos espacios de memoria son reservados al momento de definir el arreglo, siendo para su definición para el caso de:

Vector (Unidimensional) => **Dim** Vector(LimiteMaximo) **As** TipoDato

Matriz (Bidimensional) => **Dim** Matriz(LimiteFila, LimiteColumna) **As** TipoDato

Entendiéndose que el identificador del arreglo almacena es la ubicación en memoria del primer elemento del arreglo, y el tipo de dato que puede este almacenar, puede ser cualquier tipo de dato que pueda emplearse en el programa desarrollado, es decir, un dato tipo simple, estructura u objeto de ser necesario.

Ahora bien, ya que el identificador del arreglo como tal no almacena el valor de un dato almacenado, al momento de necesitar manejar un dato o varios que este contenga, se tiene que hacer acceso al espacio de memoria del dato a manejar, es decir, para almacenar, imprimir o utilizar un dato del arreglo sea para un cálculo o asignación, se debe hacer referencia a la posición o índices del elemento a usar ej.

- a) Lectura por teclado del elemento i del vector: **Vector(i) = Console.ReadLine().**
- b) Asignación en la posición fila 2 columna 4 de una matriz el valor 45: **Matriz(2, 4) = 45**

La posición o índice es un dato entero que puede ser constante o variable, generalmente se trabaja es con variables para hacer referencia a las posiciones que se quieren accesar, y estas deben ser menor o igual al límite máximo establecido al momento de definir el arreglo. Los índices que se pueden hacer acceso en el arreglo son desde la posición cero hasta el límite establecido al momento de su definición.

Para no hacer acceso a una posición que no contiene un dato o una posición no reservada, siempre se tendrán variables que guarden el tamaño del arreglo, es decir por ejemplo, cuando se emplea un vector en un programa existirán dos variables en uso, la variable tipo arreglo (Vector) y otra variable tipo entera que guarde el tamaño del vector (tv).

Finalmente, los arreglos pueden ser empleados como parámetros de un subprograma, la diferenciación es que cuando se define por valor o referencia hay que entender que es la posición en memoria la que se está estableciendo de esa manera, no los datos que contiene el arreglo, lo que significa que sin importar como se pasó el arreglo (ByVal o ByRef), si se modifica un dato contenido en él, este cambio es efectivo permanentemente. Por otro lado, cuando se establece el parámetro no debe tener tamaño definido el arreglo. Ej.

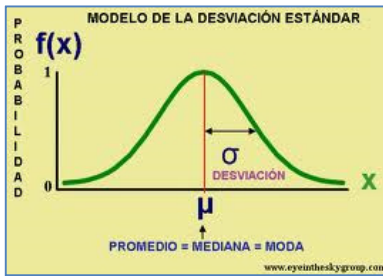
Vector pasado por valor: **Function** Promedio(**ByVal** V() **As** TipoDato, **ByVal** N **As** Integer) **As** Single

Matriz pasada por referencia:

Sub Leer(**ByRef** M(,) **As** TipoDato, **Byref** nFil **As** Integer, **ByRef** nCol **As** Integer)

Y al momento de invocar el subprograma, si este emplea arreglos como parámetros se coloca solamente el identificador del mismo sin incluir algún paréntesis. Ej. **Leer(Matriz, Fila, Columnas)**

Escenario 1: Calculo Estadístico



Uno de los procesos más empleados en la industria es calcular la media aritmética de un conjunto de valores finito, así como su desviación estándar. Sabiendo que, siendo X el conjunto de N elementos numéricos:

$$\text{Media aritmética: } \bar{X} = \frac{\sum_{i=1}^N x_i}{N}$$

$$\text{Desviación estándar: } \sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{X})^2}{N - 1}}$$

ENUNCIADO

Elabore un programa en VB2010 que dado el archivo de datos “**Datos.Txt**” en el cual se almacenó un conjunto de valores, determine e imprima por pantalla, cada uno de los elementos leídos como un vector fila y luego el resultado de su media aritmética como su desviación estándar.

Ejemplo de entrada y salida del programa planteado

Datos.txt

16.49
14.21
11.46
14.29
17.29
14.28

Salida por pantalla

X=	16.49	14.21	11.46	14.29	17.29	14.28
Media =	14.67					
Desviación Estándar=	2.049					

Requerimientos:

Para la solución del problema debe definir y utilizar:

1. Un subprograma que lea el archivo de datos “**Datos.Txt**” y almacena toda la información que en él se encuentre en el vector Q de elementos tipo **Single**.
2. Un subprograma que dado el vector **V** de **Tv** elementos tipo **Single**, retorne el valor de su Media Aritmética.
3. Un subprograma que dado el vector **Z** de **Tz** elementos tipo **Single**, retorne el valor de su Desviación estándar.
4. Un subprograma que imprima por pantalla el vector (**A**) de **Ta** elementos tipo **Single**, como un vector fila.

Escenario 2: Cálculo de probabilidades en un dado

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]** en forma consecutiva como fueron generados. 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, 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 ser empleada 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 número, valor debe ser un dato numérico

Artificio matemático 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 se cumple que $A \leq N \leq B$; caso contrario retorne **FALSE**
2. Un subprograma que lea un valor entero **N** que esté en el rango [10, 1000] y utilizando el subprograma anterior para validar la entrada. En caso que el valor de N no se valide, el subprograma debe seguir leyendo N hasta que se cumpla que $10 \leq N \leq 1000$. Si no se cumple la condición se debe imprimir un mensaje que indique el error por pantalla.
3. 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].
4. Un subprograma que imprima un arreglo v[N] tipo **Entero** hacia el archivo **Narch** (Número del archivo), en forma de vector fila.

N= 15	Probabilidades.Txt
Valores generados	4, 2, 3, 2, 6, 1, 1, 5, 6, 2, 3, 4, 5, 1, 2
6	Porcentaje de veces que sale 1: 20.00%
6	Porcentaje de veces que sale 2: 6.67%
3	Porcentaje de veces que sale 3: 6.67%
5	Porcentaje de veces que sale 4: 26.67%
6	Porcentaje de veces que sale 5: 13.33%
6	Porcentaje de veces que sale 6: 26.67%
4	
5	
1	
4	
1	
1	
2	
4	
4	