SEMINARIO DE ENCUENTRO NÚMERO 7

PRESEMINARIO

Cada uno de los estudiantes deberá realizar individualmente la siguiente lectura. Luego al finalizar realizará pruebas de escritorio al ejercicio planteado y socializará en el foro sus opiniones.

ARREGLOS

Con frecuencia puede suceder que se deba manejar una colección de datos que estén relacionados como un conjunto de datos, ya sean alfanuméricos, numéricos o lógicos y la forma de hacerlo, es a través de una estructura de datos, y la más sencilla corresponde a un arreglo.

Un arreglo se define como un conjunto de datos finito, homogéneo (es decir que todos son del mismo tipo) los cuales se relacionan entre sí, a través de un nombre (del arreglo) y un subíndice (identifica la posición del dato en el arreglo).

1.1. Clasificación de los arreglos

- Arreglos unidimensionales (vectores)
- Arreglos multidimensionales (matrices)

Arregios unidimensionales (vectores)

Corresponde a un conjunto de datos donde éstos se representan en forma lineal (ya sea fila o columna) y para referenciar un elemento del arreglo se requiere de un índice que indica la posición del elemento en el arreglo.

Representación gráfica de un arreglo unidimensional:

Como fila:

Nombre

1 2 3 4

Elementos del arreglo

Como columna

Nombre

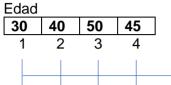
1 2
3 4

Elementos del arreglo

Donde Nombre equivale al escogido para hacer referencia a todos los elementos del arreglo y cada rectángulo representa los elementos que lo conforman y los números los índices de estos.

Cuando al nombre de un arreglo se le adiciona el índice entre corchetes ([]) se está refiriendo al dato que está en la posición indicada por el índice y éste es tratado como una variable simple, por lo tanto puede hacer parte de una asignación, una expresión, una lectura o una escritura.

Ejemplo, si se tienen las edades de 5 personas (30, 40, 50, y 45) y se quieren almacenar en memoria, se puede hacer uso de un arreglo llamado Edad



Elementos del arreglo

Por ejemplo si se hace referencia a Edad[3], estaría refiriéndose al elemento 50 Por ejemplo si se hace referencia a Edad[1], estaría refiriéndose al elemento 30

1.2. Declaración de un arreglo

Igual que las variables y constantes, los arreglos pueden ser declarados pública o localmente y se declaran de la misma forma y en el mismo lugar como se hace con las variables, solo que entre corchetes ([]) después del nombre, se especifican la cantidad de elementos que tendrá el arreglo. Es importante tener presenta que cuando se declara un arreglo usualmente se requiere de una variable tipo entero que va a manejar los índices del arreglo (para identificar los diversos elementos)

Ejemplo



1.3. Manipulación de un arreglo

Usualmente un arreglo puede ser manipulado para llenarlo (Leer), para desplegarlo (Escribir) y éste puede ser manipulado de dos formas:

- Directa
- A través de un ciclo

La manipulación directa no es muy práctica cuando se requiere manipular todo el arreglo y obedece a la forma mostrada en el ejemplo anterior.

Si se desea escribir el contenido del arreglo en un elemento específico sería así: Escribir Edad[1] y este mostraría el valor 30

Si se desea almacenar un valor específico en el arreglo sería así:

Leer Edad[4] Almacena en el elemento 4 el valor ingresado por el usuario ó Edad[4] ← 3 Asigna el valor 3 en el elemento 4

Manipulación a través de ciclos

Los arreglos pueden ser manipulados con cualquiera de los ciclos vistos en anteriormente y se sugiere crear un subprograma para cada tipo de manipulación, de tal forma que el subprograma sirva para manipular un arreglo del mismo tipo, pero de diferente tamaño.

```
    Lectura de arreglo
    Procedimiento_LLENAR (Entero N, Vec[])
    Inicio

            Variables
            Entero I
            Para (I=1, N, 1) hacer
            Leer (Vec[I])
            Fin Para

    Fin_LLENAR

            Imprimir el arreglo
            Procedimiento_MOSTRAR (Entero N, Vec[])
            Inicio
            Variables
```

Entero I Para (I=1, N, 1) hacer

Fin Para

Fin MOSTRAR

Escribir (Vec[I])

El algoritmo que invocaría estos subprogramas se puede observar a continuación y este puede servir para leer o mostrar cualquier arreglo de un tamo menor que el definido.

```
Algoritmo_Arreglos
Inicio

Variables
Entero Vector[100], Cant
Escribir ("Ingrese la cantidad de elementos a manipular (máximo 100)")
Leer (Cant)
Llamar_a_LLENAR(Cant, Vector)

Llamar_a_MOSTRAR(Cant, Vector)

Fin
```

3

1.4. Operaciones Básicas Con Arreglos

Cuando se tiene un arreglo se pueden realizar básicamente las siguientes operaciones

- Búsqueda
- Inserción
- Borrado
- Ordenación

1.4.1. Búsqueda

Consiste en encontrar un determinado valor dentro de un arreglo. Existen diversos métodos de búsqueda:

- Búsqueda secuencial
- Búsqueda Binaria
- Búsqueda HASH
- Árboles de búsqueda

A continuación, se presentan los dos primeros métodos y se invita a los estudiantes para que consulten acerca de los otros dos métodos:

Búsqueda Secuencial

Es la técnica más sencilla para buscar un elemento dentro de un arreglo y consiste en recorrer todo el arreglo elemento a elemento desde el principio hasta el fin. Si se encuentra el valor buscado el subprograma devolverá la posición dónde se encuentra dicho elemento y si no se encuentra, el subprograma devolverá un valor diferente a las posiciones.

La búsqueda supone que en el algoritmo o subprograma que invoca la búsqueda, se haya leído el valor a buscar. A continuación, se muestra el subprograma:



```
Entero Función_BUSQUEDA(entero : Vec[], N, Vr_B)
Inicio
```

```
Variables
Entero: I

I ← 1

Mientras (I <= N AND Vr_B <> Vec[I]) Hacer
I ← I + 1

Fin Mientras
Si(I <= N) Entonces
Devolver (I)
Si_No
Devolver (0)
Fin_Si

Fin_BUSQUEDA
```

El algoritmo que llama el subprograma se muestra a continuación

```
Algoritmo_XYZ
Inicio

Var

Entero ; Vector [10], T, Bus, En
Escribir (Ingrese el número de elementos a llenar máximo 10"

Leer (T)

Llamar_a_LLENAR( T, Vector)
Escribir ("Ingrese el valor a buscar en el arreglo ")
Leer (Bus)

En ← (BUSQUEDA(Vector, T, B)
Si (En = 0) Entonces
Escribir ("El valor buscado no fue encontrado")

Si_no

Escribir ("El valor fue encontrado en la posición", En)
Fin-SI
```

Búsqueda Binario

Fin

En la búsqueda secuencial cuando no se encuentra el valor buscado, hay que recorrer todo el arreglo, lo que implica poca eficiencia del método (en especial cuando el arreglo es muy grande).

La búsqueda binaria se realiza en un arreglo cuyos elementos se encuentren ordenados y consiste en dividir el arreglo en dos partes y comparar el valor buscado con el elemento de la mitad, si no hay coincidencia entonces se determina si el valor buscado está en la primera mitad del arreglo o en la segunda mitad, eliminando así, la búsqueda en la mitad que no se pueda encontrar. Posteriormente se toma la mitad donde posiblemente puede encontrarse el valor buscado y se repite el procedimiento. A continuación, se muestra el subprograma que implementa la búsqueda binaria:

```
Entero Función_BUSQUEDA_BINARIA(Entero : Vec[], N, Vr_B) Inicio

Variables

Entero P, U, Sw, Mitad

P \leftarrow 1

U \leftarrow N

Sw \leftarrow 0

Mientras((P <= U) AND (Sw = 0)) Hacer

Mitad \leftarrow Entero ((P + U)/2)

Si(Vec[Mitad] = Vr_B) Entonces

Sw \leftarrow 1

Si_No

Si (Vr_B < Vec[Mitad]) Entonces

U \leftarrow Mitad - 1

Si_no
```

```
P ← Mitad + 1
Fin_SI
Fin_SI
Fin_Mientras
Si Vr_b = Vec[Mitad) Entonces
Devolver(Mitad)
Si_no_
Devolver (0)
Fin_SI
Fin_BUSQUEDA_BINARIA
```

1.4.2. Inserción

Esta operación consiste en adicionar un nuevo elemento al arreglo. Se debe tener en cuenta que solo se puede realizar siempre y cuando el tamaño del arreglo que se está manipulado (tamaño virtual) sea inferior al tamaño del arreglo definido en la declaración (Tamaño real). La inserción se puede hacer de dos formas

- En un arreglo desordenado
- En un arreglo ordenado.

Inserción en uno desordenado es bastante simple, se incrementa en 1 el tamaño virtual del arreglo y en esa posición se inserta el nuevo elemento.

```
Procedimiento_INSERTAR(Entero : Vec[ ], Tv, Vr_Ins)
Inicio

Tv ← Tv + 1

Vec[Tv] ← Vr_Ins
Fin INSERTAR
```

En el algoritmo o subprograma que invoca al procedimiento se debe controlar el tamaño del arreglo.

Ejemplo:

```
Algoritmo_XYZ
```

Inicio

Var

Entero; Vector [10], T, Vr_I

Escribir (Ingrese el número de elementos a llenar máximo 10"

```
Leer (T)

Llamar_a_LLENAR( T, Vector)

Si (T < 10) Entonces

Escribir ("Ingrese el valor a insertar en el arreglo ")
```

```
Leer (Vr_I)
Llamar_a_INSERTAT (Vector, T, Vr_I)
Si_no
Escribir ("El arreglo no tiene espacio para insertar un nuevo valor")
Fin-SI
Fin
```

Inserción en un arreglo ordenado

ACTIVIDAD DE REPASO

Realizar un programa que, a través de subprogramas realice las siguientes tareas:

- Capturare las cédulas de los N estudiantes de un grupo. Se sabe que un grupo nunca tiene más de 50 alumnos
- Mostrar las cédulas de los N estudiantes de un grupo.
- Buscar una cédula en al arreglo (Secuencial y binaria)
- Insertar una nueva cédula (ordenado y desordenado)

Para cada uno de las búsquedas e inserciones realizar las respectivas pruebas de escritorio