

PRACTICA 9: VECTOR ORDENADO

ESTUDIANTE: EFRAIN ROBLES PULIDO

CODIGO: 221350095

NOMBRE DE LA MATERIA: SEMINARIO DE SOLUCION DE PROBLEMAS DE PROGRAMACION

SECCIÓN: D67

Descripción:

Objetivo de la practica

Hacer un programa que ordene los elementos de un arreglo unidimensional de 100 elementos de tipo entero y llenado con números aleatorios entre 0 y 999 y muestre tanto el vector original desordenado y el vector ordenado en pantalla.

Fundamentación teórica

Un arreglo (vector, array, matriz) es un conjunto de datos o una estructura de datos homogéneos que se encuentran ubicados en forma consecutiva en la memoria RAM (sirve para almacenar datos en forma temporal).

Una de las formas más comunes de ordenación de un array es el famoso método de la burbuja. Se basa en el hecho de ir recorriendo todo el array a ordenar, comparando dos elementos al mismo tiempo e intercambiándolos si están en el orden incorrecto. Al terminar de recorrer los elementos, se determina si hubo algún cambio, y de haberlo, se repite el método hasta que ya no haya cambio alguno.

Análisis del problema

Se deberá utilizar los ciclos for para realizar la práctica, uno para introducir los números de manera aleatoria de 0 a 1000 en el arreglo e imprimirlo, de manera desordenada, otras de manera anidada para ordenar el arreglo y otro para mostrar el mismo arreglo, pero ya ordenado, en el que se usará la condición de una sola línea para realizarlo correctamente.

Datos de entrada y precondiciones

Ninguno, pero el programa generara los números aleatoriamente, con la función random, para introducirlos en el arreglo.

Datos o elementos de salida

- El arreglo desordenado.
- El arreglo ordenado.

Desarrollo:

Procedimiento en lenguaje natural

Se utilizará un ciclo for para introducir los números aleatorios, con la ayuda de la función rand con un rango de 0 hasta 1000, y se mostrara el arreglo sin ordenar, luego se usara dos ciclos for de manera anidada, pero en el segundo ciclo terminara un valor antes debido que estamos utilizando el valor de la posición siguiente del arreglo, luego utilizaremos una if para tener como condición si el valor de la posición actual del arreglo es menor que el valor de la posición siguiente del arreglo, para que se reescriba los valores del arreglo, en donde se guardará el valor actual en el valor temp, luego reescribirá el valor de la posición actual con el valor de la posición siguiente del arreglo para después, escribir el valor de temp en la posición siguiente del arreglo. Para finalmente usar otro ciclo for para mostrar el arreglo ya ordenado.

Algoritmo

Pseudocódigo:

```
Principal
                                                                     desde (j \leftarrow 0; j < N; inc j)
Inicio
                                                                          desde (i \leftarrow 0; i<N-1; inc i)
    constante N ← 100
                                                                                if(a[i] < a[i+1])
    entero i, j, a[N], temp
                                                                                  inicio
    srand (time (NULL)) //inicializador de
                                                                                       temp ← a[i]
numeros aleatorios
                                                                                       a[i] \leftarrow a[i+1]
    desde (i \leftarrow 0; I < N; inc i)
                                                                                        a[i+1] ← temp
    inicio
                                                                                  fin
           a[i] \leftarrow rand()\%1000
                                                                     desde (i \leftarrow 0; i<N; inc i)
           imprimir (a[i])
                                                                             imprimir (a[i])
    fin
                                                                Fin
    imprimir ("\n\n")
```

Código fuente del programa en lenguaje C (textualmente)

```
//Efrain Robles Pulido
                                                                    }
#include <stdio.h>
                                                                    printf("\n\n\n");
#include <stdlib.h>
                                                                    for(j=0;j<N;j++)
#include <time.h>
                                                                       for(i=0;i< N-1;i++)
#define N 100 //100
                                                                          if(a[i] < a[i+1]){
                                                                            temp=a[i];
void main(){
                                                                            a[i]=a[i+1];
  int i, j, a[N], temp;
                                                                            a[i+1]=temp;
  srand(time(NULL)); //inicializador de
                                                                          }
numeros aleatorios
                                                                    for(i=0;i<N;i++)
  for(i=0;i<N;i++){}
                                                                       printf("%4i",a[i]);
     a[i]=rand()%1000;
                                                                 }
     printf("%4i",a[i]);
```

Resultados obtenidos:

```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 100 //100
void main(){
    int i, j, a[N], temp;
    srand(time(NULL)); //inicializador de numeros aleatorios
    for(i=0;i<N;i++){
        a[i]=rand()%1000;
        printf("%4i",a[i]);
    printf("\n\n\n");
    for (j=0; j<N; j++)
        for(i=0;i<N-1;i++)
            if(a[i] < a[i+1]) {</pre>
                 temp=a[i];
                 a[i]=a[i+1];
                 a[i+1]=temp;
    for (i=0; i<N; i++)
        printf("%4i",a[i]);
}
```

391 826 587 984 653 805 705 665 543 884 592 215 970 548 596 631 414 342 832 703 207 103 668 3 331 907 804 8 189 772 371 572 161 98 808 469 425 283 288 146 759 853 188 79 884 624 852 453 144 474 139 361 859 366 85 172 443 933 463 523 935 143 974 335 724 667 479 991 741 637 630 781 361 280 808 367 184 485 389 911 669 106 585 309 83 286 634 474 17 820 415 521 63 871 319 914 790 965 453 686

991 984 974 970 965 935 933 914 911 907 884 884 871 859 853 852 832 826 820 808 808 805 804 790 781 772 759 741 724 705 703 686 669 6 68 667 665 653 637 634 631 630 624 596 592 587 585 572 548 543 538 523 521 485 475 474 474 469 463 453 453 443 425 415 414 391 389 371 367 366 361 361 342 335 331 319 309 288 286 283 280 215 207 189 188 184 172 161 146 144 143 139 106 103 85 83 79 63 17 8 3 Process returned 4 (0x4) execution time : 0.041 s

Conclusión

En esta práctica aprendí como puedo usar los ciclos for para poder interactuar con los arreglos unidimensionales, así como utilizar la función random para llenar estos arreglos con números aleatorios con cierto rango. También tuve que hacer una estrategia para poder reescribir los valores del arreglo según nuestra condición. Finalmente, para esta práctica tuve que usar la analogía de hacer ciclos tanto para llenar e imprimir el arreglo y para ordenar los valores asegurándome de que se haya recorrido todos los valores del arreglo.