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

Código fuente del programa en lenguaje C (textualmente)

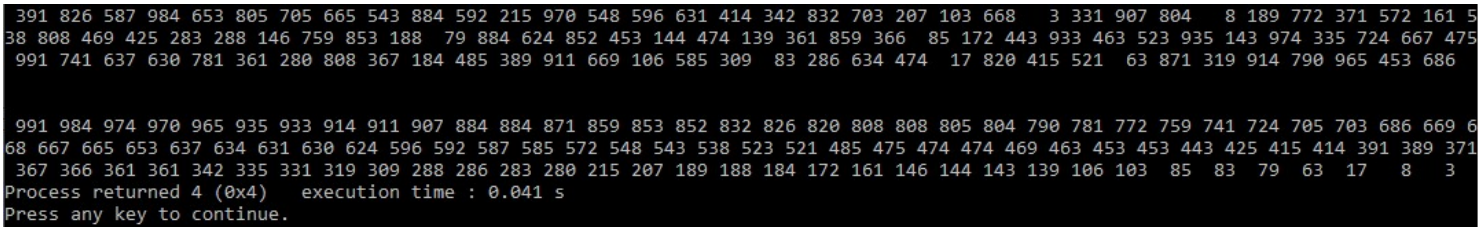
```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 100 //100

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]){
            temp=a[i];
            a[i]=a[i+1];
            a[i+1]=temp;
         }

   for(i=0;i<N;i++)
      printf("%4i",a[i]);
}
```

Resultados obtenidos:



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.