



Programación Procedural

Trabajo Práctico 5

Programación Procedural

Trabajo Práctico 5

Estructuras de Datos. Arreglos. Vectores. Matrices.

Nota: Para todos los ejercicios:

- Utilice programación modular y estructurada.
- Confeccione cartas estructuradas

1.- Desarrollar un programa que permita la lectura de un vector A de 'n' números enteros, usando una función, y luego lo imprima, con otra función, con el siguiente formato:

A [1] = x1 { x1 es el contenido de A [1] }

A [2] = x2

....

2.- Hacer un programa que lea un vector A de 'n' elementos enteros, que busque el elemento de mayor valor e imprima el siguiente mensaje:

El elemento de mayor valor es: ... y ocupa la posición: ... del vector.

3.- Hacer un programa que ingrese un vector de 'n' elementos y luego intercambie el primer elemento por el n-esimo; el segundo por n-esimo menos 1; y así sucesivamente .- No debe generarse un nuevo vector. Como salida debe imprimir el vector original y el vector intercambiado, con las especificaciones correspondientes.

4.- Hacer un programa que lea dos vectores A y B, y realice la suma y el producto escalar de ambos.

5.- Dado un arreglo de 'n' elementos, buscar un cierto elemento, contenido en el vector, e indicar como salida la posición que ocupa dentro del arreglo.

a) Suponer que el elemento no se halla repetido.

b) El elemento puede no estar o estar mas de una vez. En ambos casos sacar carteles informativos y en el segundo caso detallar la cantidad de veces que fue hallado y las correspondientes posiciones.

6.- Dado un vector de n elementos, realizar las siguientes operaciones:

a) Sumar el primer termino con el n-esimo, y cargarlo en un vector auxiliar como elemento uno.

b) Sumar el segundo termino con el n-esimo menos 1 y cargarlo en la segunda posición del auxiliar. Y así sucesivamente hasta agotar las posibilidades.

- Considerar n par (dimensión del vector).
- Considerar n impar.

El programa debe de imprimir el arreglo original y el auxiliar.

7.- Dado un número entero, descomponerlo y almacenar sus dígitos en un vector.

Ej. Nro. 347 ---> genera el vector A= (3, 4, 7)

8.- Dados los dígitos de un número, en un vector A, calcular el número, almacenarlo en una variable simple, y mostrarla.

9.- Dado el siguiente programa:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[4] ;
    int b[4];
    int c,i,aux;
    int bandera;
    clrscr();
    bandera = 1;
    c= 1;
    for (i= 0;i<=4;i=i+1)    /* Ingreso de los vectores */
    {
        scanf("%d",&a[i]);
        scanf("%d",&b[i]);
    }
    for (i= 0;i<=4;i=i+1)
        if (a[i] != b[i])
        {
            bandera = 0;
            aux = a[i];
            a[i] = b[i];
            b[i] = aux;
        }
        else
            c = c + 1;

    if (bandera)
    {
        printf(" a = (");
```

```
        for (i= 0;i<=4;i=i+1)
            printf("%d ",a[i]);
        printf("\n");
    }
    else
    {
        for (i= 0;i<=4;i=i+1)
            printf("%d %d",b[i], a[i]);
    }
}
```

a) Hacer una traza completa del programa, con los siguientes vectores:

a = (1,2,3,4,5) y b=(1,2,3,4,5)

b) Indicar que imprime el programa.

c) Cual es la funcion de la variable 'c' y 'bandera'

d) Como se puede mejorar este programa. (usando funciones)

10. Realizar un programa que permita concatenar en un vector un conjunto de números enteros positivos dados. Utilizar el dígito -1 en el vector resultante como separador de los números almacenados.

Ejemplo:

Dados los números 1456, 5632, 9876

El vector resultante es:

1 4 5 6 -1 5 6 3 2 -1 9 8 7 6 -1

11. Realizar un programa que permita almacenar en un vector Número los números de la lotería y en un vector Posicion la posición que ocupa. Se deben almacenar los 10 primeros números de la lotería.

Además el programa deberá imprimir la siguiente salida:

Posición Número

1	45
2	33
3	29
4	02
5	14
6	19
7	98
8	56
9	44
10	11

12. Escribe un programa en C que permita sumar dos matrices de tamaño $N \times M$. Las dimensiones de las matrices serán ingresadas por el usuario. Implementar las siguientes funciones:

- `void leerMatriz(int matriz[][M], int filas, int columnas):` Lee los elementos de una matriz desde el teclado.
- `void sumarMatrices(int matrizA[][M], int matrizB[][M], int resultado[][M], int filas, int columnas):` Suma dos matrices y almacena el resultado en una tercera matriz.
- `void imprimirMatriz(int matriz[][M], int filas, int columnas):` Imprime los elementos de una matriz por pantalla.

Nota:

- Las matrices se pasarán por referencia a las funciones para modificarlas directamente.
- La función `sumarMatrices` se encargará de realizar la suma elemento a elemento de las matrices.

13. Escribe un programa en C que calcule la transpuesta de una matriz cuadrada. La dimensión de la matriz será ingresada por el usuario. Implementa las siguientes funciones:

- `void leerMatriz(int matriz[][N], int dimension):` Lee los elementos de una matriz cuadrada desde el teclado.
- `void transponerMatriz(int matriz[][N], int dimension):` Calcula la transpuesta de una matriz y modifica la matriz original.
- `void imprimirMatriz(int matriz[][N], int dimension):` Imprime los elementos de una matriz cuadrada por pantalla.

Nota:

- La matriz transpuesta se obtiene intercambiando filas por columnas.
- La función `transponerMatriz` modificará directamente la matriz original para obtener su transpuesta.

14. Escribe un programa en C que busque un valor específico en una matriz de enteros. Las dimensiones de la matriz y el valor a buscar serán ingresados por el usuario. Implementa las siguientes funciones:

- `void leerMatriz(int matriz[][M], int filas, int columnas):` Lee los elementos de una matriz desde el teclado.
- `bool buscarValor(int matriz[][M], int filas, int columnas, int valor):` Busca un valor en una matriz y devuelve `true` si lo encuentra, `false` en caso contrario.
- `void imprimirPosiciones(int matriz[][M], int filas, int columnas, int valor):` Imprime las posiciones (índices de fila y columna) en las que se encuentra el valor buscado.

Nota:

- La función `buscarValor` recorrerá la matriz completa en busca del valor y retornará un valor booleano indicando si fue encontrado.

- Programación Procedural

- La función `imprimirPosiciones` mostrará las coordenadas de todas las celdas donde se encontró el valor.