

LABORATORIO 7

TABLAS NUMÉRICAS - ARRAYS

1 OBJETIVOS

Al finalizar esta actividad, serás capaz de:

- Entender el funcionamiento de las tablas numéricas.
- Realizar las operaciones básicas: declarar una tabla , lectura y escritura de un elemento de la tabla, recorrer una tabla, ordenar una tabla etc ... sin usar funciones.

2 MOTIVACIÓN

- Supongamos que queremos resolver un programa, que dada una lista de números finalizada por 0, la muestre en orden invertido.
- Supongamos que queremos resolver un programa, que dada una lista de números finalizada por 0, muestre el número que más se repite.
- Los problemas anteriores y más no se pueden resolver sin usar estas tablas numéricas.

2.1 Las tablas numéricas en C

- Para usar la tablas numéricas hay que declararlas o definirlas teniendo que especificar el identificador de tabla, el número de elementos que la contienen y tipo de datos de los elementos que la contienen. Ejm: float notas_alum[12];
- Para leer o escribir un dato de la tabla hay que indicarle el índice del elemento a leer o escribir. Ejm para el primer elemento de una tabla:
 - Lectura->printf("%d",notas_alum[0]);
 - Escritura-> scanf("%d",¬as_alum[0]);
- Para recorrer los elemento de una tabla utilizaremos un indice que irá desde el índice 0 hasta el número de "elementos - 1". Ejm de carga y visualización de todos los elementos de una tabla:

Ejm:

```
/* Inicio del programa ppa1 */
void main()
{
    /*indice para recorrer la tabla */
    int i;

    /*número de elementos de una tabla */
    int const NELE=10;

    /*Declaración o definición de una tabla de enteros */
    int notas_alum[NELE];
```

```

/* Carga de todos los elementos de una tabla */
for (i=0;i<=NELE -1;i++)
{
    printf("Introduce la nota del alumno %d",i);
    scanf("%d",&notas_alum[i]);
}

/* Visualización de todos los elemento de la tabla */
for (i=0;i<=NELE -1;i++)
{
    printf("la nota del alumno %d es %d",i,notas_alum[i]);
}
}

```

3 EJERCICIOS

3.1 Ejercicio 1

3.1.1 Objetivo:

El objetivo del ejercicio 1 es entender como se recorre una tabla tratando sus elementos, es decir, en este caso moviéndolos a la izquierda.

3.1.2 Ayuda:

Se da resuelto el enunciado con comentarios para tomarlo de guía para el enunciado 1 y 2:

```

#include <stdio.h>
#include <stdlib.h>
void main()
{
    int t[10],ind,lon,num,aux;
    ind=0;

    do{
        printf ("Introduce un número");
        scanf("%d",&num);
        t[ind]=num;
        ind=ind+1;
    }while(ind<10);

    /* Al salir del anterior bucle ind está apuntando al 10, es decir,
       el número de elementos que tiene el array. Guardamos el primer elemento
       sobre aux antes de empezar a desplazar el resto a la izqda. */

    lon=ind;
    aux=t[0];

    printf ("aux %d\n",aux);

    /* Poner el indice a 1, cuando desplazamos hacia la izda, empezamos
       tomando una posición y asignándoselo a la que está a su izquierda, por eso
       empezamos en la segunda posición para asignárselo a la primera*/

```

```
ind=1;

/*Bucle para realizar el desplazamiento, CUIDADO,
darnos cuenta de dos cosas: empezamos en ind=1 y acabamos en
ind=long-1*/

while(ind < lon){
    t[ind-1]=t[ind];
    ind=ind+1;
}

/* En la última posición guardamos lo que había antes en la primera */
t[lon-1]=aux;

/*No olvidar poner el índice a 0 para implementar el bucle para recorrer e
imprimir la tabla*/

ind=0;
while(ind < lon){
    printf("%d\n",t[ind]);
    ind=ind+1;
}
system("PAUSE");
```

3.1.3 Enunciado 1:

Hacer un programa que pida al usuario los elementos de una tabla de tamaño N y a continuación desplace todos los elementos a la izquierda, ubicando el primero de ellos en la última posición.

Ejemplo (suponiendo que N sea 10):

DAME 10 ENTEROS: 3 24 7 -25 0 3 21 19 3 895

El array desplazado hacia la IZQUIERDA queda así:
24 7 -25 0 3 21 19 3 895 3

3.2 Ejercicio 2

3.2.1 Objetivo:

El objetivo del ejercicio 2 es poner en práctica los conocimientos adquiridos en el ejercicio anterior.

3.2.2 Ayuda para el enunciado 2:

Utilizar el ejercicio 1.

3.2.3 Enunciado 2:

Igual que el ejercicio anterior, sólo que esta vez desplazando los elementos hacia la derecha y dejando el último elemento en la primera posición del array.

Ejemplo (suponiendo que N sea 10):

DAME 10 ENTEROS: 3 24 7 -25 0 3 21 19 3 895

El array desplazado hacia la DERECHA queda así:
895 3 24 7 -25 0 3 21 19 3

3.3 Ejercicio 3

3.3.1 Objetivo:

El objetivo del ejercicio 3 es el uso de contadores e índices que guardan los estados por los que pasa el programa al recorrer la tabla. Tomando como estado en este caso la cantidad máxima de elementos iguales y su posición de inicio y final.

3.3.2 Ayuda para el enunciado 3:

Al recorrer los elementos de la tabla cuando el siguiente elemento es distinto al actual es que hemos terminado con una secuencia y si procede actualizo el máximo de posiciones repetidas y los índices

3.3.3 Enunciado 3:

Dada una tabla de N elementos, calcular la cantidad máxima de elementos iguales consecutivos que hay, así como las posiciones de comienzo y final en las que se encuentran.

Ejemplo (para N igual a 10):

i:	0	1	2	3	4	5	6	7	8	9
V:	15	23	23	5	7	4	4	4	23	23

El mayor número de elementos consecutivos iguales que aparece es 3 y se encuentran entre las posiciones 5 y 7.

3.3.4 Ejercicio 4

3.3.5 Objetivo:

El objetivo del ejercicio 4 es aprender a usar condiciones a la hora de cargar tablas o recorrer las tablas, ya que no siempre tenemos un número fijo de elementos.

3.3.6 Ayuda para el enunciado 4:

Recorremos la tabla desde el principio hasta el final buscando el mínimo y guardando tanto su valor como su posición.

3.3.7 Enunciado del ejercicio 4

Pedir al usuario una serie de números y almacenarla en un array de tamaño N . El programa deberá pedir números hasta que se hayan introducido N elementos en la tabla o el usuario teclee un número negativo. Una vez se dispone del array con como mucho N números positivos, se pide intercambiar el menor de ellos con el que está en la primera posición.

Ejemplo (suponiendo que N sea 10):

DAME HASTA 10 NUMEROS (NEGATIVO PARA FINALIZAR):
45 67 13 29 3 28 32 51 -1

El array, una vez realizado el intercambio, queda así:
3 67 13 29 45 28 32 51

3.4 Ejercicio 5

3.4.1 Objetivo:

El objetivo del ejercicio 5 es utilizar el algoritmo burbuja para ordenar de mayor a menor los elementos de una tabla.

3.4.2 Enunciado 5:

Hacer un programa que ordene de mayor a menor los elementos de un vector de como mucho N números. Igual que en el ejercicio anterior, el vector puede no llenarse, dado que si el usuario teclea un número negativo no se seguirán pidiendo más valores.

Ejemplo (suponiendo que N sea 10):

DAME HASTA 10 NUMEROS (NEGATIVO PARA FINALIZAR)
4.2 1.3 4.75 4.2 0.55 -1.3

El array, una vez ordenado, queda así: 0
.55 1.3 3.25 4.2 4.2 4.75