

PRÁCTICAS PARA LA ASIGNATURA

FUNDAMENTOS DE INFORMÁTICA

PRÁCTICA 6:
Vectores y Punteros.

INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Pedro María Alcover Garau
Dolores Cano Gil
Carlos Fernández Andrés
Pedro Javier Navarro Lorente



PRÁCTICA SEXTA

1. Un uso común para un vector inicializado es la creación de una tabla de búsqueda. Por ejemplo, en este programa se inicializa una matriz bidimensional de 5x2 de modo que el primer elemento de cada fila es el nombre de una CPU, y el segundo elemento contiene su velocidad media de procesamiento. El programa permite que un usuario introduzca el número de un procesador y después examina la tabla e informa de su velocidad media.

```
/* Programa TABLA de BUSQUEDA */
#include <stdio.h>
    long lCpu[5][2]={
                                {8088, 4},
                                {8086,4},
                                {80286, 10},
                                {80386, 20},
                                {80486, 40}
                                };

void main(void)
{
    short int i;
    long lProcesador;

    printf("\nIntroduzca el numero del procesador: ");
    scanf("%ld", &lProcesador);
/* Examinar en la tabla */
    for(i = 0 ; i < 5 ; i++)
        if(lProcesador==lCpu[i][0])
        {
            printf("La velocidad media es ");
            printf("%d Mhz. \n", lCpu[i][1]);
            break;
        }
/* comunicar error si no lo encuentra */
    if(i == 5) printf("Procesador no encontrado \n");
} /* fin de main() */
```

2. Los vectores resultan especialmente útiles cuando se quiere ordenar información. Este programa permite que el usuario introduzca hasta 100 números y después los ordena de menor a mayor. El algoritmo de ordenación es el de la burbuja. El algoritmo de ordenación de la burbuja es pobre en eficiencia, pero fácil de entender y sencillo de codificar.



Práctica nº 6: Vectores y Punteros. **3 de 5**

```
/* programa ORDENACION */
# include <stdio.h>
#include <stdlib.h>
# define DIM_MAX 100

void main(void)
{
    long int V[DIM_MAX], aux;
    short int i, j;
    short int N;          // Números introducidos

    printf("Cuántos números desea ordenar? \n");
    scanf("%hd", &N);

    for(i = 0 ; i < N ; i++)
    {
        printf("Entrada (%hd) --> ", i + 1);
        scanf("%ld", &V[i]);
    }

    for(i = 0 ; i < N ; i++)
        for(j = i + 1 ; j < N ; j++)
            if(V[i] > V[j]) // Intercambiar...
            {
                aux = V[i];
                V[i] = V[j];
                V[j] = aux;
            }

    for(i = 0 ; i < N ; i++)
        printf("%10ld", Vi[]);
}
```

- 3. Este programa permite al operador introducir diez cadenas, después le permite mostrarlas, una a una, en cualquier orden que elija. Para detener el programa se introducirá un número negativo.**

```
/* Programa CADENAS */
# include <stdio.h>
# define LINEAS 10
# define CARACTERES 80

void main(void)
{
    char cTexto[LINEAS][CARACTERES];
    short int i;

    for(i = 0 ; i < LINEAS ; i++)
    {
        printf("%d: ", i+1);
```



Práctica nº 6: Vectores y Punteros. 4 de 5

```
        gets(cTexto[i]);
    } /* fin for() */

do
{
    printf("\n¿Numero de linea (1-¿hd)? :", LINEAS);
    scanf("%hd", &i);
    i--; /* ajustar valor para indice de vector */
    if(i >= 0) printf("%s", cTexto[i]);
} while (i >= 0); /* fin do-while() */
} /* fin de main() */
```

- 4. Un uso común de los vectores de caracteres es para implementar una interfaz basada en órdenes. A diferencia de un menú, que permite que el usuario haga una elección, una interfaz basada en órdenes muestra un mensaje indicativo, espera a que el usuario introduzca una orden y después hace lo que pide la orden. Este programa permite que el usuario sume, reste, multiplique o divida dos datos, sin usar menú, sino interfaz de órdenes.**

```
/* programa COMANDOS */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void main(void)
{
    long int i, j;
    char cComando[80], cTemporal[80];

    for( ; ; )
    {
        printf("Operacion?: ");
        gets(cComando);

        /* comprobar si el usuario desea abandonar la aplicacion */
        if(!strcmp(cComando, "salir")) break;

        printf("Introduzca el primer número:");
        gets(cTemporal);
        i=atoi(cTemporal);

        printf("Introduzca el segundo número:");
        gets(cTemporal);
        j=atoi(cTemporal);

        /* ahora, realiza la operación */
        if(!strcmp(cComando, "sumar"))
            printf("La suma es %d\n", i + j);
```



Práctica nº 6: Vectores y Punteros. **5 de 5**

```
else if(!strcmp(cComando, "restar"))
    printf("La resta es %d\n", i - j);

else if (!strcmp(cComando, "dividir"))
    if(j) printf ("la división es: %d\n", i / j);

else if (!strcmp(cComando, "multiplicar"))
    printf ("la multiplicación es: %d\n", i * j);

    else printf ("Comando desconocido \n");
} /* fin for vacio */
} /* fin de main() */
```

- 5. Escribir un programa que inicialice una matriz de 10x3 de modo que el primer elemento de cada fila contenga un número, el segundo elemento contenga su cuadrado y el tercer elemento contenga su cubo.**
- 6. Escriba un programa que defina una matriz de tres dimensiones de 3x3x3, y almacene los números del 1 al 127. Haga que el programa muestre la suma de los datos.**