



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Alejandro Pimentel

Profesor:

Fundamentos de programación

Asignatura:

3

Grupo:

11

No de Práctica(s):

Integrante(s):

Colonia Montero Sonia
Godínez Juárez Alondra Itati

*No. de Equipo de
cómputo empleado:*

7455
6153

No. de Lista o Brigada:

2020-1

Semestre:

28/10/2019

Fecha de entrega:

Observaciones:

Bien

CALIFICACIÓN: 10

Introducción

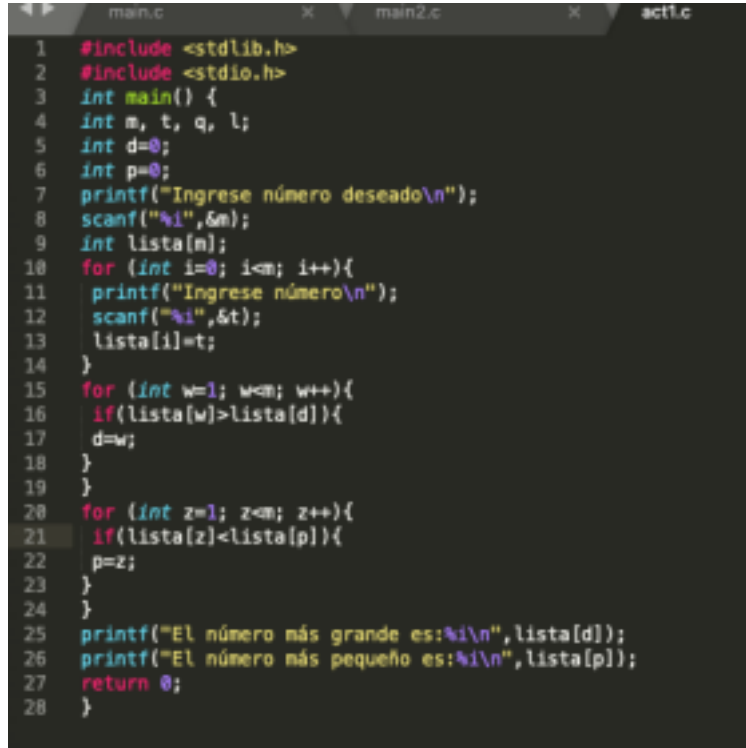
Existe una manera de agrupar datos de un mismo tipo dentro de un solo arreglos o un arreglo de arreglos, también conocido como matriz. En la elaboración de códigos es bastante común trabajar con este tipo de arreglos ya que tienden a facilitar el trabajo, por lo tanto es importante conocerlos, entenderlos y saber cómo programarlos. Es por ello que a lo largo del presente documento se busca conocer la importancia de saber utilizar los arreglos unidimensionales y multidimensionales para la elaboración de códigos computacionales.

Objetivo

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

Desarrollo

En primer lugar, se hizo un programa que pida al usuario un número, genere un arreglo de esa longitud, pida al usuario números suficientes para llenar el arreglo y que muestre al usuario el número menor y el mayor de dicho arreglo.



```
1  #include <stdlib.h>
2  #include <stdio.h>
3  int main() {
4  int m, t, q, l;
5  int d=0;
6  int p=0;
7  printf("Ingrese número deseado\n");
8  scanf("%i",&m);
9  int lista[m];
10 for (int i=0; i<m; i++){
11     printf("Ingrese número\n");
12     scanf("%i",&t);
13     lista[i]=t;
14 }
15 for (int w=1; w<m; w++){
16     if(lista[w]>lista[d]){
17         d=w;
18     }
19 }
20 for (int z=1; z<m; z++){
21     if(lista[z]<lista[p]){
22         p=z;
23     }
24 }
25 printf("El número más grande es:%i\n",lista[d]);
26 printf("El número más pequeño es:%i\n",lista[p]);
27 return 0;
28 }
```

Posteriormente, se hizo un programa que Pida al usuario un dos números N y M, Genere dos matrices de $N \times M$, Pida al usuario números suficientes para llenar ambas matrices y que Muestre al usuario la matriz resultado de sumar las dos de entrada.

```

#include <stdio.h>
#include <stdlib.h>
int main() {
    int N, M, t;
    printf("Ingrse numero de filas N\n");
    scanf("%i",&N);
    printf("Ingrse numero de columnas M\n");
    scanf("%i",&M);
    int matriz1[N][M];
    int matriz2[N][M];
    int matrizR[N][M];

    for(int m=0; m<M; m++){
        for(int n=0; n<N; n++){
            printf("Ingrese numero para matriz 1\n");
            scanf("%i",&t);
            matriz1[n][m]=t;
        }
    }

    for(int m=0; m<M; m++){
        for(int n=0; n<N; n++){
            printf("Ingrese numero para matriz 2\n");
            scanf("%i",&t);
            matriz2[n][m]=t;
        }
    }

    for(int m=0; m<M; m++){
        for(int n=0; n<N; n++){
            matrizR[n][m]=matriz1[n][m]+matriz2[n][m];
        }
    }

    printf("Matriz resultado:\n");
    for(int m=0; m<M; m++){
        for(int n=0; n<N; n++){
            printf("%i\t",matrizR[n][m]);
        }
        printf("\n");
    }
    return 0;
}

```

Resultados

Todos los programas se compilaron y se leyeron.

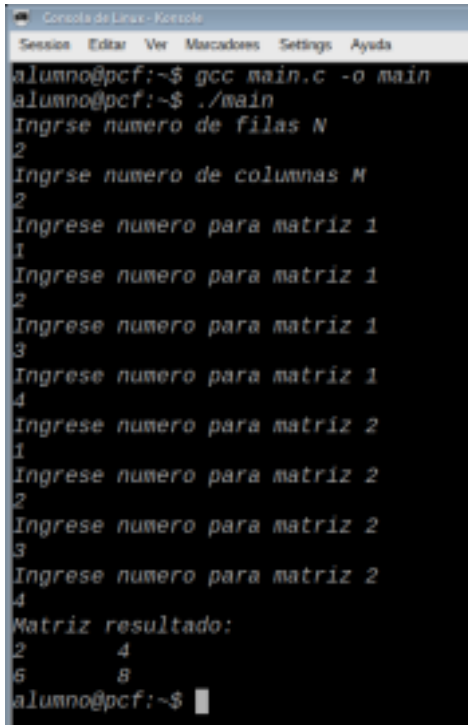
Arreglos unidimensionales

```

Escritorio — -bash — 112x28
[Lituania09:desktop fp03alu08$ gcc act1.c -o act1
[Lituania09:desktop fp03alu08$ ./act1
Ingrese número deseado
3
Ingrese número
1
Ingrese número
2
Ingrese número
3
El número más grande es:3
El número más pequeño es:1
[Lituania09:desktop fp03alu08$ ./act1
Ingrese número deseado
5
Ingrese número
4
Ingrese número
7
Ingrese número
1
Ingrese número
9
Ingrese número
0
El número más grande es:9
El número más pequeño es:0
Lituania09:desktop fp03alu08$

```

Arreglos multidimensionales



```
alumno@pcf:~$ gcc main.c -o main
alumno@pcf:~$ ./main
Ingrese numero de filas N
2
Ingrese numero de columnas M
2
Ingrese numero para matriz 1
1
Ingrese numero para matriz 1
2
Ingrese numero para matriz 1
3
Ingrese numero para matriz 1
4
Ingrese numero para matriz 2
1
Ingrese numero para matriz 2
2
Ingrese numero para matriz 2
3
Ingrese numero para matriz 2
4
Matriz resultado:
2      4
6      8
alumno@pcf:~$
```

Conclusión

El agrupar datos en manera de arreglos unidimensionales o multidimensionales tiene múltiples ventajas, entre ellas que puede ser más fácil trabajar con un solo arreglo que contenga a todos los datos a utilizar en lugar de estar trabajando con cada dato individualmente. Así mismo, el usar arreglos de datos permite elaborar códigos más organizados y por lo tanto más fáciles de entender y trabajar con ellos.