



## Arreglos unidimensionales y multidimensionales.

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de Programación

Grupo: 3

Alumna: Aguilar Lara Alexa Patricia y Vázquez Espinosa Ximena Itzel

No. de Equipo de cómputo empleado: Máquina 28 Níger

No. de lista o Brigada: 01 y 52

No. de cuenta: 316315515 y

317058015

Fecha de entrega: 28/10/2019

Observaciones: Falta evidencia del correcto funcionamiento de la última actividad, e insito esta carátula no es aceptable.

\_\_\_\_\_ CALIFICACIÓN: 8

# Arreglos unidimensionales y multidimensionales

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.

Se ven temas nuevos con los que no estamos acostumbrados a convivir, sin embargo, observaremos su importancia a través de la realización de programas en los que sea de ayuda la utilización de éstos términos.

## Arreglos unidimensionales



```
#include <stdio.h>
#define TAMANO 5

int main(int argc, char *argv[])
{
    int lista[TAMANO] = {23, 5, 34, 19, 0};

    printf("Lista:\n");
    for(int i=0; i< TAMANO-1; i++){
        printf("%i, ",lista[i]);
    }

    printf("%i\n",lista[TAMANO-1]);

    return 0;
}
```

```
Peppermint Terminal
usuario@host ~ $ ./matn
Lista:
23, 5, 34, 19, 0
usuario@host ~ $ _
```

## Arreglos multidimensionales

```
#include <stdio.h>
#define DIM 3

int main(int argc, char *argv[])
{
    int matriz[DIM][DIM] = {{23, 5, 34},
                             { 8, 46, 22},
                             { 3, 9, 12}};

    printf("Matriz:\n");
    for(int i=0; i < DIM; i++){
        for(int j=0; j < DIM ; j++){
            printf("%i\t",matriz[i][j]);
        }
        printf("\n");
    }

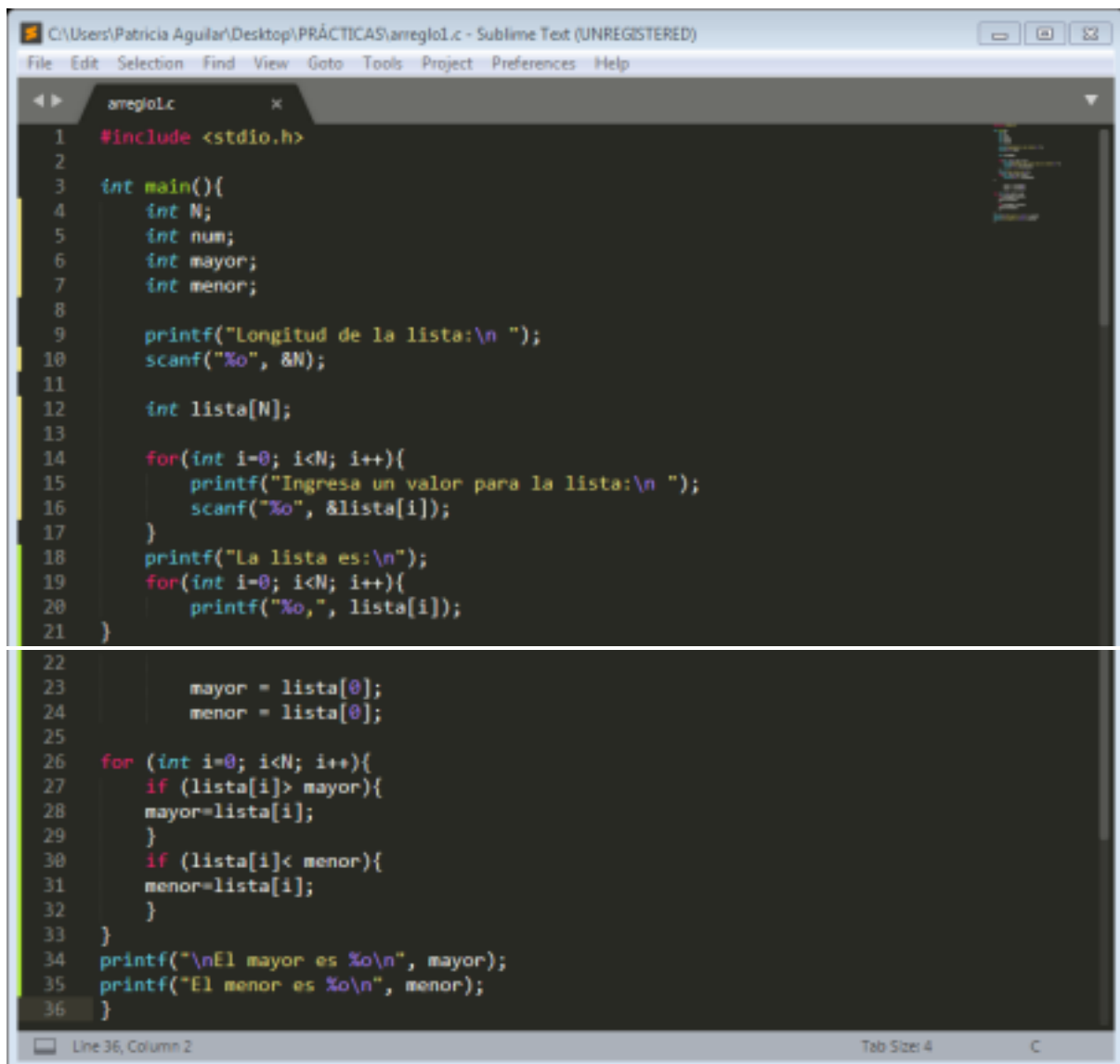
    return 0;
}
```

```
Peppermint Terminal
usuario@host ~ $ ./matn
Lista:
23, 5, 34, 19, 0
usuario@host ~ $ ./matn
Matriz:
23      5      34
8       46     22
3       9      12
usuario@host ~ $ _
```

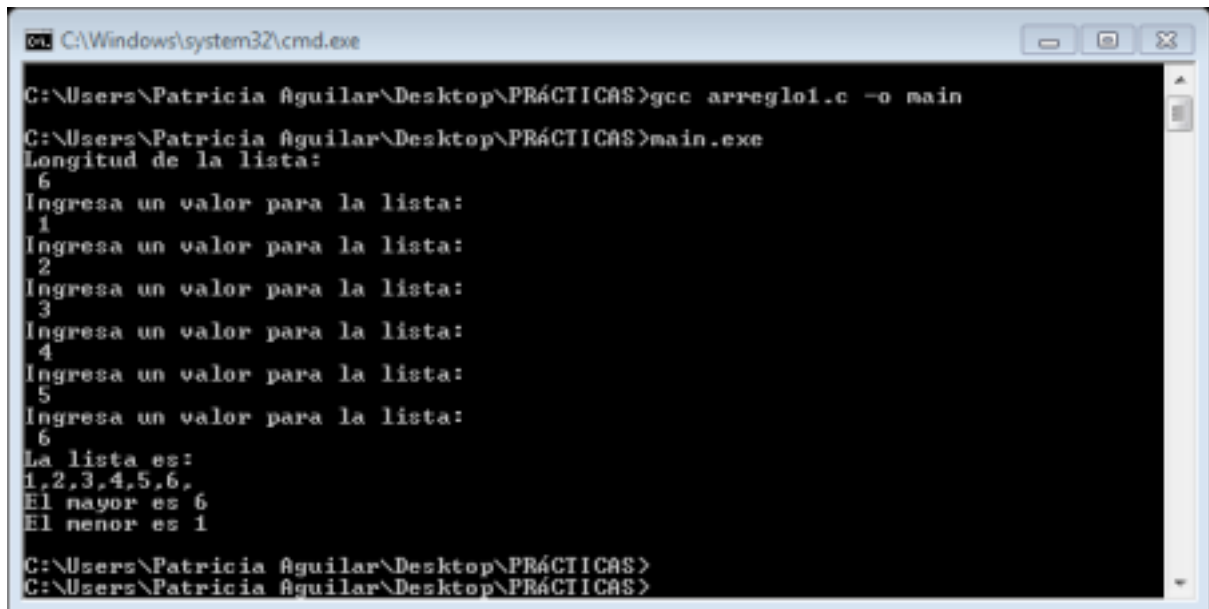
## Actividad 1

Hacer un programa que:

- Pida al usuario un número.
- Genere un arreglo de esa longitud.
- Pida al usuario números suficientes para llenar el arreglo.
- Muestre al usuario el número menor y el mayor de dicho arreglo.



```
1  #include <stdio.h>
2
3  int main(){
4      int N;
5      int num;
6      int mayor;
7      int menor;
8
9      printf("Longitud de la lista:\n ");
10     scanf("%o", &N);
11
12     int lista[N];
13
14     for(int i=0; i<N; i++){
15         printf("Ingresa un valor para la lista:\n ");
16         scanf("%o", &lista[i]);
17     }
18     printf("La lista es:\n");
19     for(int i=0; i<N; i++){
20         printf("%o,", lista[i]);
21     }
22
23     mayor = lista[0];
24     menor = lista[0];
25
26     for (int i=0; i<N; i++){
27         if (lista[i]> mayor){
28             mayor=lista[i];
29         }
30         if (lista[i]< menor){
31             menor=lista[i];
32         }
33     }
34     printf("\nEl mayor es %o\n", mayor);
35     printf("El menor es %o\n", menor);
36 }
```



```
C:\Windows\system32\cmd.exe

C:\Users\Patricia Aguilar\Desktop\PRÁCTICAS>gcc arreglo1.c -o main
C:\Users\Patricia Aguilar\Desktop\PRÁCTICAS>main.exe
Longitud de la lista:
6
Ingresa un valor para la lista:
1
Ingresa un valor para la lista:
2
Ingresa un valor para la lista:
3
Ingresa un valor para la lista:
4
Ingresa un valor para la lista:
5
Ingresa un valor para la lista:
6
La lista es:
1,2,3,4,5,6,
El mayor es 6
El menor es 1

C:\Users\Patricia Aguilar\Desktop\PRÁCTICAS>
C:\Users\Patricia Aguilar\Desktop\PRÁCTICAS>
```

## Actividad 2

Hacer un programa que:

- Pida al usuario un número N.
- Genere dos matrices de N x N.
- Pida al usuario números suficientes para llenar ambas matrices.
- Muestre al usuario la matriz resultado de sumar las dos de entrada.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void sumaMatrices(int x){
5
6     int m[x][x];
7     int n[x][x];
8     int suma[x][x];
9     int a;
10    int b;
11    int c;
12    char pausa;
13
14    for (a=0; a<x; a++){
15        for (b=0; b<x; b++){
16            m[a][b]=0;
17            n[a][b]=0;
18            suma[a][b]=0;
19        }
20    }
21
22    for(c=1; c<=2; c++){
23        for(a=0; a<x; a++){
24            for(b=0; b<x; b++){
25                if(c==1){
26                    printf("Valor de casilla [%d][%d] en matriz m: \n", a,b);
27                    fflush(stdin);
28                    scanf("%d",&m[a][b]);
29                }
30                else{
31                    printf("Valor de casilla [%d][%d] en matriz n: \n", a,b);
32                    fflush(stdin);
33                    scanf("%d",&n[a][b]);
34                }
35            }
36        }
37    }
38
39    for(a=0; a<x; a++){
40        for(b=0; b<x; b++){
41            suma[a][b]=m[a][b]+n[a][b];
42        }
43    }
44
45    system("CLS");
46
47    printf("Matriz m:\n");
48    for(a=0; a<x; a++){
49        for(b=0; b<x; b++){
50            printf("%d", m[a][b]);
51        }
52        printf("\n");
53    }
54    printf("\n");

```

```

55     printf("Matriz m:\n");
56     for (a=0; a<x; a++){
57         for(b=0; b<x; b++){
58             printf("%d", m[a][b]);
59         }
60         printf("\n");
61     }
62     printf("\n");
63     printf("Matriz suma m+n:\n");
64     for(a=0; a<x; a++){
65         for (b=0; b<x; b++){
66             printf("%d", suma[a][b]);
67         }
68         printf("\n");
69     }
70     printf("Presione una tecla para continuar");
71     fflush(stdin);
72     pausa = getch();
73 }
74
75 void menu()
76 {
77     int opc;
78     int Naux;
79
80     do{
81
82         clrscr("CLS");
83         printf("Suma de Matrices\n");
84         printf("1.- Suma de dos matrices\n q.- Salir\n");
85
86         fflush(stdin);
87         scanf("%d", &opc);
88         if(opc!=0){
89             printf("Valor para dimensionar las matrices:\n");
90             fflush(stdin);
91             scanf("%d", &Naux);
92
93             switch(opc)
94             {
95                 case '1': sumaMatrices(Naux);
96                     break;
97                 default:
98                     printf("\n Saliendo del programa\n");
99                     break;
100             }
101         }
102     }
103     }while(opc!=0);
104
105 }
106
107 void main()
108 {
109     menu();
110 }

```

Al ser un tema nuevo, fue algo complejo de comprender. La unidimensionalidad y multidimensionalidad al final fue una idea algo compleja de digerir, sin embargo logramos realizar la práctica.