


| | | |
|---|---|--|
|  | Carátula para entrega de prácticas | |
| Facultad de Ingeniería | Laboratorio de docencia | |

Laboratorios de computación salas A y B

| | |
|---|--|
| <i>Profesor:</i> | M.C. Alejandro Esteban Pimentel Alarcón |
| <i>Asignatura:</i> | Fundamentos de programación |
| <i>Grupo:</i> | 3 |
| <i>No de Práctica(s):</i> | 11 |
| <i>Integrante(s):</i> | Aguilar Lara Patricia Alexa Vázquez Espinosa Ximena Itzel |
| <i>No. de Equipo de cómputo empleado:</i> | 40 |
| <i>No. de Lista o Brigada:</i> | 5515 8015 |
| <i>Semestre:</i> | Primer semestre |
| <i>Fecha de entrega:</i> | 28 de Octubre 2019 |
| <i>Observaciones:</i> | |
| | |

CALIFICACIÓN: _____

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.

Actividad 1:

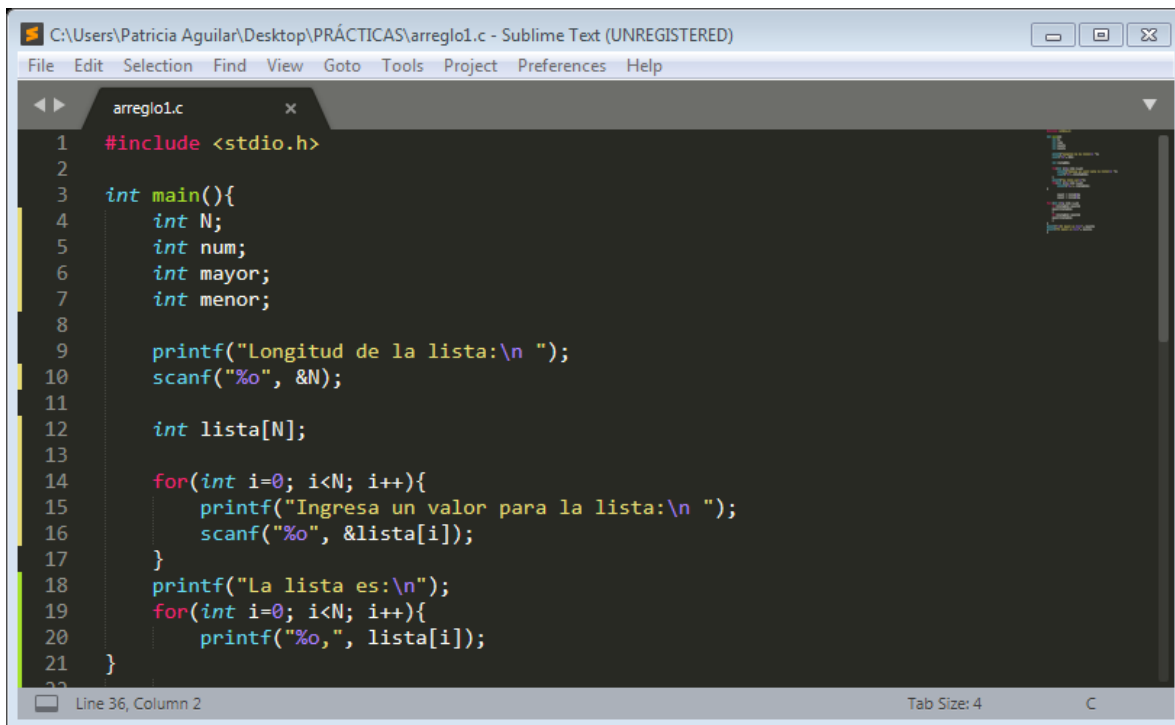
Hacer un programa que:

Pida al usuario un número.

Genere un arreglo de esa longitud.

Pida al usuario números suficientes para llenar el arreglo.

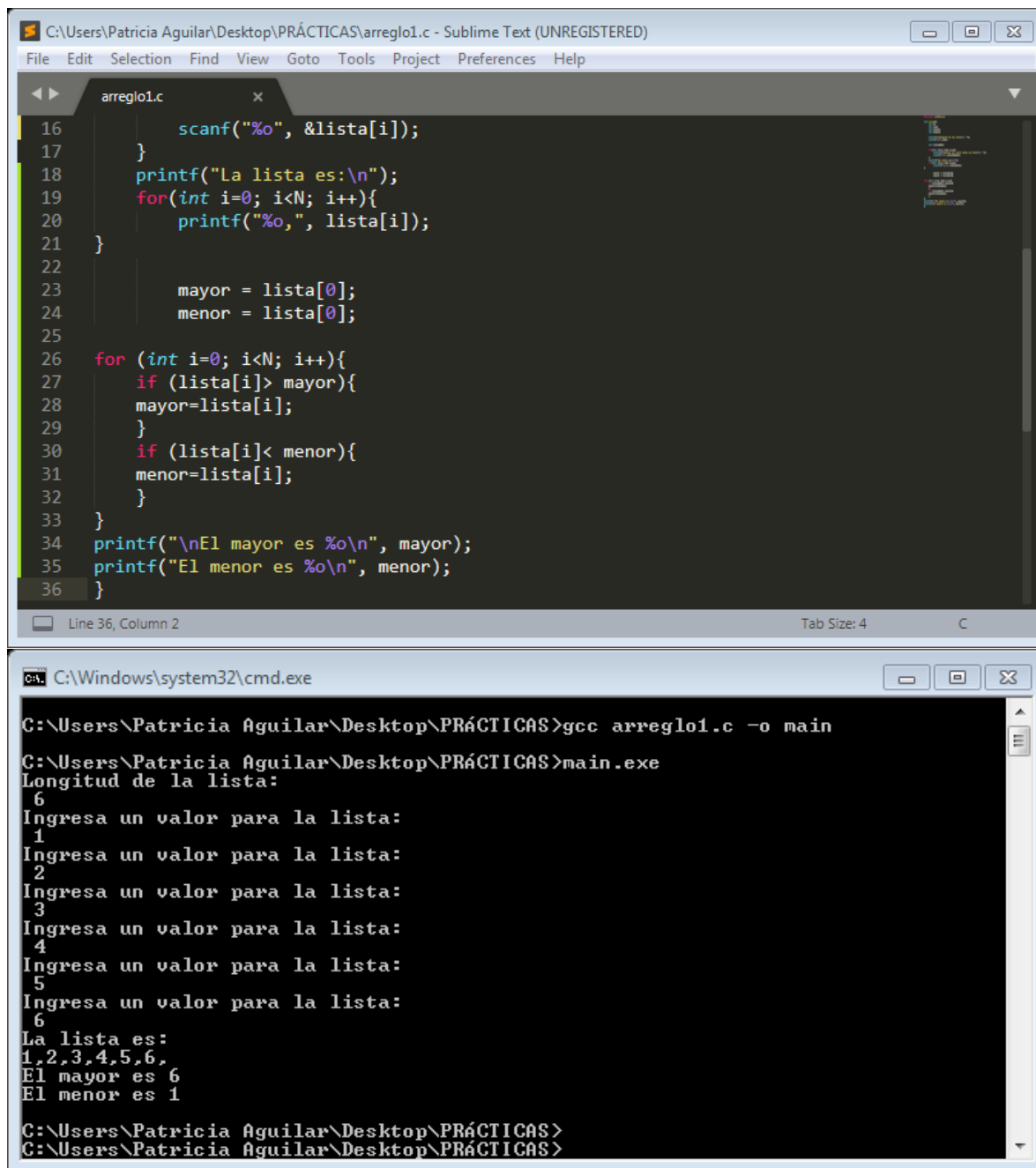
Muestre al usuario el número menor y el mayor de dicho arreglo.



```
C:\Users\Patricia Aguilar\Desktop\PRÁCTICAS\arreglo1.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

arreglo1.c
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 }
```

Line 36, Column 2 Tab Size: 4 C



```
C:\Users\Patricia Aguilar\Desktop\PRÁCTICAS\arreglo1.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

arreglo1.c
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 }
```

Line 36, Column 2 Tab Size: 4 C

```
cmd 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>
```

Actividad2:

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.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```

void sumaMatrices(int x){

    int m[x][x];

    int n[x][x];

    int suma[x][x];

    int a;

    int b;

    int c;

    char pausa;

    for (a=0; a<x; a++){
        for (b=0; b<x; b++){
            m[a][b]=0;

            n[a][b]=0;

            suma[a][b]=0;

        }
    }

    for(c=1; c<=2; c++){
        for(a=0; a<x; a++){
            for(b=0; b<x; b++){
                if(c==1){
                    printf("Valor de casilla [%d][%d] en matriz m: \n", a,b);

                    fflush(stdin);

                    scanf("%d", &m[a][b]);

                }

                else{
                    printf("Valor de casilla[%d][%d] en matriz m: \n", a,b);

```

```
        fflush(stdin);

        scanf("%d", &m[a][b]);

    }

}

}
```

```
for(a=0; a<x; a++){

    for(b=0; b<x; b++){

        suma[a][b]=m[a][b]+n[a][b];

    }

}
```

```
system ("CLS");
```

```
printf("Matriz m:\n");

for(a=0; a<x; a++){

    for(b=0; b<x; b++){

        printf("%d", m[a][b]);

    }

    printf("\n");

}

printf("\n");
```

```
printf("Matriz n:\n");

for (a=0; a<x; a++){

    for(b=0; b<x; b++){

        printf("%d", m[a][b]);

    }

}
```

```

        printf("\n");
    }
    printf("\n");
    printf("Matriz suma m+n:\n");
    for(a=0; a<x; a++){
        for (b=0; b<x; b++){
            printf("%d", suma[a][b]);
        }
        printf("\n");
    }
    printf("Presione una tecla para continuar");
    fflush(stdin);
    pausa = getch();
}

void menu()
{
    int opc;
    int Naux;

    do{
        system("CLS");
        printf("Suma de Matrices\n");
        printf("1.- Suma de dos matrices\n q.- Salir\n");

        fflush(stdin);
        scanf("%d", &opc);
        if(opc!=0){

```

```
printf("Valor para dimensionar las matrices:\n");  
fflush(stdin);  
scanf("%d", &Naux);
```

```
switch(opc)  
{  
    case '1': sumaMatrices(Naux);  
        break;  
    default:  
        printf("\n Saliendo del programa\n");  
        break;  
}  
}
```

```
}while(opc!=0);
```

```
}
```

```
void main()
```

```
{
```

```
    menu();
```

```
}
```

```

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 m: \n", a,b);
32                     fflush(stdin);
33                     scanf("%d", &m[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
56     printf("Matriz n:\n");
57     for (a=0; a<x; a++){
58         for(b=0; b<x; b++){
59             printf("%d", m[a][b]);
60         }
61         printf("\n");
62     }
63     printf("\n");
64     printf("Matriz suma m+n:\n");
65     for(a=0; a<x; a++){
66         for (b=0; b<x; b++){
67             printf("%d", suma[a][b]);
68         }
69         printf("\n");
70     }
71     printf("Presione una tecla para continuar");
72     fflush(stdin);
73     pausa = getchar();
74
75 }
76
77 void menu()
78 {
79     int opc;
80     int Naux;
81
82     do{

```

```

81
82     do{
83         system("CLS");
84         printf("Suma de Matrices\n");
85         printf("1.- Suma de dos matrices\n q.- Salir\n");
86
87         fflush(stdin);
88         scanf("%d", &opc);
89         if(opc!=0){
90             printf("Valor para dimensionar las matrices:\n");
91             fflush(stdin);
92             scanf("%d", &Naux);
93
94             switch(opc)
95             {
96                 case '1': sumaMatrices(Naux);
97                     break;
98                 default:
99                     printf("\n Saliendo del programa\n");
100                     break;
101             }
102         }
103     }
104
105     }while(opc!=0);
106
107 }
108

```

```

104
105     }while(opc!=0);
106
107 }
108
109 void main()
110 {
111     menu();
112 }

```

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.