



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Alejandro Esteban Pimentel Alarcon

Profesor:

Fundamentos de programación

Asignatura:

3

Grupo:

11

No de Práctica(s):

Oscar García García
Ana Cristina Romero Rojas

Integrante(s):

*No. de Equipo de
cómputo empleado:*

2712
1147

No. de Lista o Brigada:

2020-1

Semestre:

28/10/19

Fecha de entrega:

Observaciones:

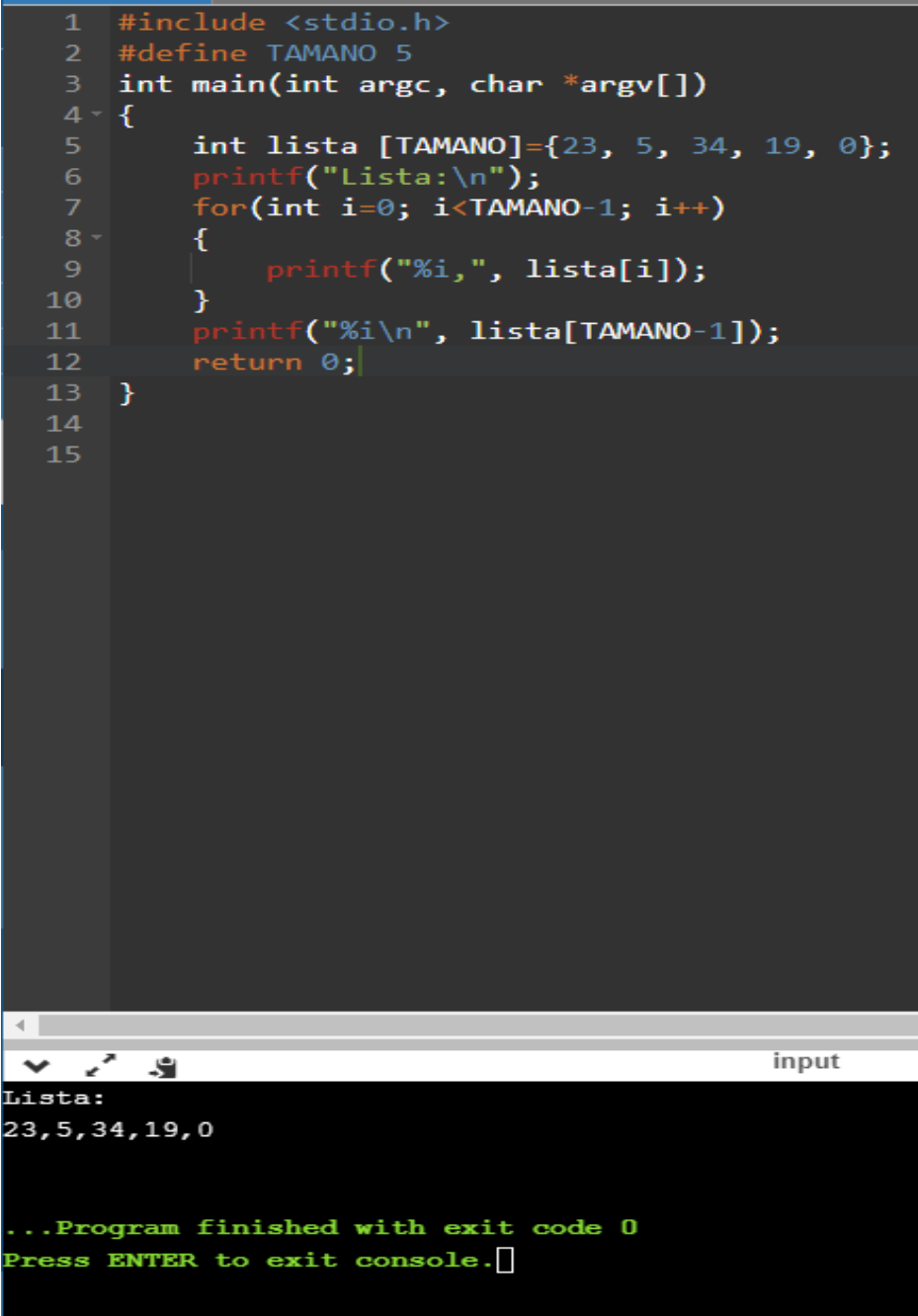
CALIFICACIÓN: _____

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.

Arreglos unidimensionales

```
1  #include <stdio.h>
2  #define TAMANO 5
3  int main(int argc, char *argv[])
4  {
5      int lista [TAMANO]={23, 5, 34, 19, 0};
6      printf("Lista:\n");
7      for(int i=0; i<TAMANO-1; i++)
8      {
9          printf("%i,", lista[i]);
10     }
11     printf("%i\n", lista[TAMANO-1]);
12     return 0;
13 }
14
15
```



input

Lista:
23,5,34,19,0

...Program finished with exit code 0
Press ENTER to exit console. □

```

1  #include <stdio.h>
2  #define DIM 3
3  int main(int argc, char *argv[])
4  {
5      int matriz[DIM][DIM]={{23, 5, 34},
6                             {8, 46, 22},
7                             {3, 9, 12}};
8      printf("matriz:\n");
9      for(int i=0; i<DIM; i++)
10     {
11         for(int j=0; j<DIM; j++)
12         {
13             printf("%i\t", matriz[i][j]);
14         }
15         printf("\n");
16     }
17     return 0;
18 }
19
20

```

☐
☐
☐

input

```

matriz:
23      5      34
8        46     22
3        9      12

...Program finished with exit code 0
Press ENTER to exit console.

```

Actividad 1:

- 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  int main()
3  {
4      int num;
5      printf("Pon un número para generar el arreglo de esa longitud\n");
6      scanf("%i", &num);
7      int lista[num];
8      for(int i=0; i<num; i++)
9      {
10         printf("lista[%i]\n", i);
11         scanf("%i", &lista[i]);
12     }
13     int a;
14     a=lista[0];
15     for(int i=1; i < num; i ++ )
16     {
17         if (lista[i]<a)
18         {
19             a=lista[i];
20         }
21     }
22     int b;
23     b=lista[0];
24     for (int i=1; i<num; i++)
25     {
26         if(lista[i]>b)
27         {
28             b=lista[i];
29         }
30     }
31     printf("El numero menor es %i\n", a);
32     printf("El numero mayor es %i\n", b);
33     return 0;
34 }
```

```
Pon un número para generar el arreglo de esa longitud
7
lista[0]
1
lista[1]
2
lista[2]
3
lista[3]
4
lista[4]
5
lista[5]
6
lista[6]
7
El numero menor es 1
El numero mayor es 7

...Program finished with exit code 0
Press ENTER to exit console.□
```

```
Pon un número para generar el arreglo de esa longitud
10
lista[0]
22
lista[1]
56
lista[2]
8
lista[3]
5
lista[4]
7
lista[5]
99
lista[6]
100
lista[7]
66
lista[8]
23
lista[9]
30
El numero menor es 5
El numero mayor es 100

...Program finished with exit code 0
Press ENTER to exit console.□
```

Actividad 2:

Hacer un programa que:

- Pida al usuario unos dos números N y M.
- Genere dos matrices de $N \times M$.
- 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  int main()
3  {
4      int x, y;
5      printf("Ingrese columnas de la matriz\n");
6      scanf("%i", &x);
7      printf("Ingrese filas de la matriz\n");
8      scanf("%i", &y);
9      int m1[x][y];
10     int m2[x][y];
11     int m3[x][y];
12     printf("Llenar la matriz 1\n");
13     printf("\n");
14     for (int a=0; a<x; a++)
15     {
16         for (int b=0; b<y; b++)
17         {
18             printf("lugar[%i][%i]\n", a+1, b+1);
19             scanf("%i", &m1[a][b]);
20         }
21     }
22     printf("\n");
23     printf("Llenar la matriz 2\n");
24     for (int a=0; a<x; a++)
25     {
26         for(int b=0; b<y; b++)
27         {
28             printf("lugar[%i][%i]\n", a+1, b+1);
29             scanf("%i", &m2[a][b]);
30         }
31     }
32     printf("\n");
33     printf("La suma de las dos matrices es\n");
34     for (int a=0; a<x; a++)
35     {
36         for(int b=0; b<y; b++)
37         {
38             m3[a][b]=m1[a][b]+m2[a][b];
39             printf("%i\t", m3[a][b]);
40         }
41     }
42     return 0;
43 }
44
```

Ingrese columnas de la matriz

3

Ingrese filas de la matriz

3

Llenar la matriz 1

lugar[1][1]

1

lugar[1][2]

2

lugar[1][3]

3

lugar[2][1]

4

lugar[2][2]

5

lugar[2][3]

6

lugar[3][1]

7

lugar[3][2]

8

lugar[3][3]

9

Llenar la matriz 2

lugar[1][1]

10

lugar[1][2]

11

lugar[1][3]

12

lugar[2][1]

13

lugar[2][2]

14

lugar[2][3]

15

lugar[3][1]

16

lugar[3][2]

17

lugar[3][3]

18

La suma de las dos matrices es

11 13 15 17 19 21 23 25 27

...Program finished with exit code 0

Press ENTER to exit console.

Conclusión:

Es muy importante conocer los arreglos unidimensionales, pues ayuda a organizar mejor los trabajos, al orden, para las matrices que veremos próximamente en álgebra y para programar por supuesto, gracias a esta práctica pudimos aprenderlo y solo que practicarlo más.