



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Alejandro Pimentel

Profesor:

Asignatura:

Fundamentos de la Programación

Grupo:

3

No de Práctica(s):

11

Integrante(s):

Valeria Patricia Padilla Arellano (2438)

*No. de Equipo de
cómputo empleado:*

34

No. de Lista o Brigada:

39

Semestre:

2020-1

Fecha de entrega:

28 de Octubre del 2019

Observaciones:

CALIFICACIÓN: _____

OBJETIVO

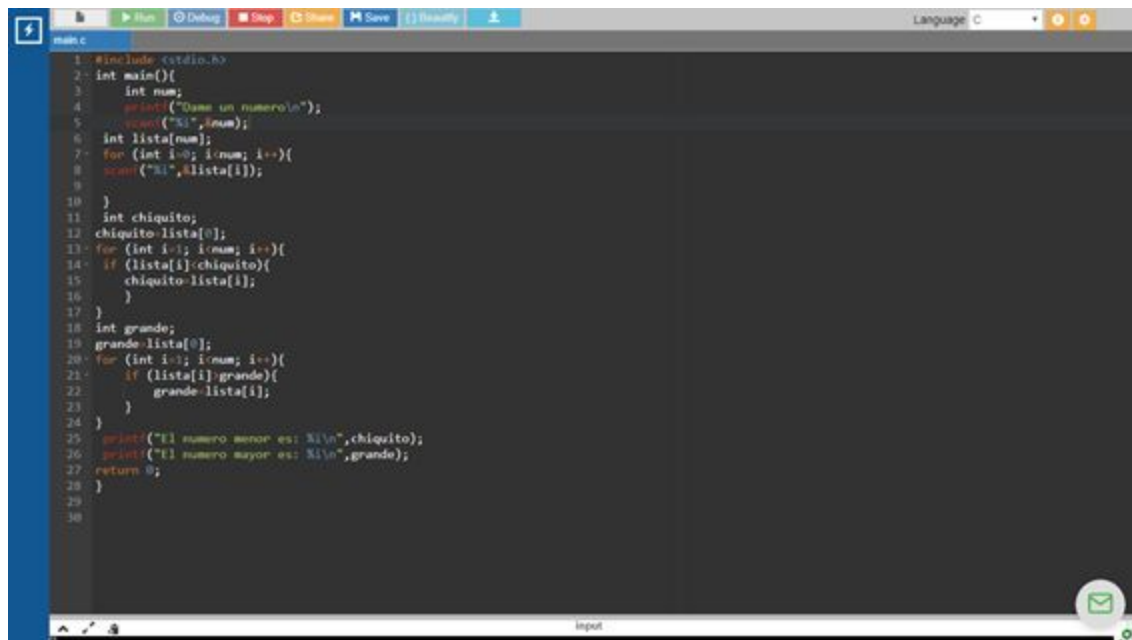
Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

INTRODUCCION

Un arreglo unidimensional es un tipo de datos estructurado que está formado por una colección finita y ordenada de datos del mismo tipo. Es la estructura natural para modelar listas de elementos iguales. Los datos que se guarden en los arreglos todos deben ser del mismo tipo.

El tipo de acceso a los arreglos unidimensionales es el acceso directo, esto mediante el uso de un índice para cada elemento del arreglo que nos da su posición relativa.

ACTIVIDAD 1



```
1 #include <stdio.h>
2 int main(){
3     int num;
4     printf("Dame un numero\n");
5     scanf("%i",&num);
6     int lista[num];
7     for (int i=0; i<num; i++){
8         scanf("%i",&lista[i]);
9     }
10 }
11 int chiquito;
12 chiquito=lista[0];
13 for (int i=1; i<num; i++){
14     if (lista[i]<chiquito){
15         chiquito=lista[i];
16     }
17 }
18 int grande;
19 grande=lista[0];
20 for (int i=1; i<num; i++){
21     if (lista[i]>grande){
22         grande=lista[i];
23     }
24 }
25 printf("El numero menor es: %i\n",chiquito);
26 printf("El numero mayor es: %i\n",grande);
27 return 0;
28 }
29
30
```

```
dame un numero
7
lista[0]=
-125
lista[1]=
345
lista[2]=
1467
lista[3]=
1234
lista[4]=
1
lista[5]=
245
lista[6]=
76
el numero menor es -125
el numero mayor es 1467
```

ACTIVIDAD 2

```
1 #include <stdio.h>
2 int main (int argc, char * argv[])
3 {
4     int n,m,x;
5     printf("Ingresa dos numeros enteros positivos ");
6     scanf("%d %d", &n, &m);
7
8     int matrix1[n][m];
9     int matrix2[n][m];
10    int matrixsuma[n][m];
11
12    printf("Ingresa %d numeros enteros positivos para  primera matriz : \n", n*m);
13
14    for (int j=0; j<m; j++){
15        for (int i=0; i<n; i++){
16            {
17                scanf("%d", &x);
18                matrix2[i][j]=x;
19            }
20        }
21        printf("Ingresa %d numeros para segunda matriz: \n",n*m);
22    for (int j=0; j<m; j++){
```

```
19    }
20    }
21    printf("Ingresa %d numeros para segunda matriz: \n",n*m);
22    for (int j=0; j<m; j++){
23        {
24            for (int i=0; i<n; i++){
25                {
26                    scanf("%d", &x);
27                    matrix2[i][j]=x;
28                }
29            }
30            printf("\nmatriz %dx%d resultante: \n", n,m);
31            for(int j=0;j<m;j++)
32            {
33                for(int i=0;i<n;i++)
34                {
35                    matrixsuma[i][j] = matrix1[i][j] + matrix2[i][j];
36                }
37                printf("\n");
38            }
39            return 0;
40        }
```

```
Ingresa dos numeros enteros positivos 1
2
Ingresa 2 numeros enteros positivos para  primera matriz :
4
5
Ingresa 2 numeros para segunda matriz:
3
9

matriz 1x2 resultante:
```

CONCLUSION

Los arreglos nos permiten hacer un conjunto de operaciones para manipular los datos guardados en ellos, estas operaciones son: ordenar, buscar, insertar, eliminar, modificar entre otras.