



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:

03

Grupo:

11

No de Práctica(s):

Nava Pamatz Oscar Gustavo

Integrante(s):

*No. de Equipo de
cómputo empleado:*

No asistí

35

No. de Lista o Brigada:

2020-1

Semestre:

28/10/2019

Fecha de entrega:

Tarde entrega

Observaciones:

CALIFICACIÓN: 7

Práctica 11

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.

Arreglo unidimensional

Un arreglo unidimensional tiene una sola dimensión.

' La dimensión se maneja por medio de un par de corchetes.

' los arreglos de una dimensión también se les llama listas

```
#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;
}
```

Arreglo multidimensionales

Arreglos multidimensionales pueden ser descritos como arreglos de arreglos. Por ejemplo, un arreglo bidimensional se puede imaginar como una tabla bidimensional de un tipo de dato concreto y uniforme.

```
#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;
}
```

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 int main(){
3     int a, num;
4     printf("Ingrese el numero de elementos que desee contenga el arreglo\n");
5     scanf("%i",&num);
6     int lista [num];
7     for(int a=0;a<num;a++){
8         printf("ingrese un número\n");
9         scanf("%i",&lista[a]);
10    }
11    int valor1=lista[0];
12    int valor2=lista[0];
13    for(int a; a<num; a++){
14        if(lista[a]>valor1){
15            valor1=lista[a];
16        }
17        if(lista[a]<valor2){
18            valor2=lista[a];
19        }
20    }
21    }
22    printf("El valor mayor es: %i\n",valor1);
23    printf("El valor menor es: %i\n",valor2);
24 }

```

Actividad 2

Hacer 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.

Muestre al usuario la matriz resultado de sumar las dos de entrada.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     int N, M, t;
5     printf("Ingrese numero de filas N\n");
6     scanf("%i",&N);
7     printf("Ingrese numero de columnas M\n");
8     scanf("%i",&M);
9     int matriz1[N][M];
10    int matriz2[N][M];
11    int matriz3[N][M];
12
13    for(int m=0; m<M; m++){
14        for(int n=0; n<N; n++){
15            printf("Ingrese numero para matriz 1\n");
16            scanf("%i",&t);
17            matriz1[n][m]=t;
18        }
19    }
20
21    for(int m=0; m<M; m++){
22        for(int n=0; n<N; n++){
23            printf("Ingrese numero para matriz 2\n");
24            scanf("%i",&t);
25            matriz2[n][m]=t;
26        }
27    }
28
29    for(int m=0; m<M; m++){
30        for(int n=0; n<N; n++){
31            matriz3[n][m]=matriz1[n][m]+matriz2[n][m];
32        }
33    }
34 }

```

```

21    for(int m=0; m<M; m++){
22        for(int n=0; n<N; n++){
23            printf("Ingrese numero para matriz 2\n");
24            scanf("%i",&t);
25            matriz2[n][m]=t;
26        }
27    }
28
29    for(int m=0; m<M; m++){
30        for(int n=0; n<N; n++){
31            matriz3[n][m]=matriz1[n][m]+matriz2[n][m];
32        }
33    }
34
35    printf("Matriz resultado:\n");
36    for(int m=0; m<M; m++){
37        for(int n=0; n<N; n++){
38            printf("%i\t",matriz3[n][m]);
39        }
40        printf("\n");
41    }
42    return 0;
43 }

```

Conclusión: En las ultimas prácticas he tenido demasiados problemas, pero espero mejorar, la actividad me pareció un poco complicada.