

25-1-2022



Tarea: 1 moda y ámbito de una función

Materia: Seminario de estructura de datos 1

Sección: D13.

Código: 216584703

Carrera: Ingeniería en computación.

Nombre alumno: Padilla Pérez Jorge Daray

Nombre profesor: Julio Esteban Valdes Lopez

Pantallazos

```
1  #include <iostream>
2  #include <string.h>
3  #define MaxData 100
4
5  using namespace std;
6
7  int Arreglo [MaxData];
8  void inserta(int cantidad){
9      if (cantidad < 0 || cantidad > MaxData){
10
11         cout<<"Ingresa un elemento valido"<<endl;
12         system("pause");
13         return;
14     }
15     for(int i = 0; i < cantidad; i ++){
16         cout <<" introduce el elemento :"<<endl;
17         cin >> Arreglo[i];
18     }
19 }
20
21 void imprimeLista(int cantidad){
22     int i;
23     for(i = 0; i < cantidad; i ++){
24         cout<<" posicion :"<<i<<endl;
25         cout<<" dato: "<<Arreglo[i]<<endl;
26     }
27 }
```

Utilizando una macro se define el tamaño máximo del arreglo en 100 y se declara debajo de este, se crea la función insertar donde no permite que sean números negativos ni que se pasen del arreglo de 100 elementos, además se muestra como se imprime la lista.

```

29 void Promedio(int cantidad){
30     int i,s = 0,prom;
31     for(i = 0; i < cantidad; i++){
32         s = s+Arreglo[i];
33     }
34     prom = s/cantidad;
35     cout << "\n suma:"<<s<<"\n promedio:"<<prom<<endl;
36 }
37
38 void moda(int cantidad){
39     int i = 0,maxre = 0;
40     int m, repeticiones = 0;
41     while( i < cantidad )
42     {
43         int j = 0;
44         while (j < cantidad)
45         {
46             if (Arreglo[i] == Arreglo[j] && i != j)
47             {
48                 repeticiones++;
49             }
50             j++;
51         }
52         if (repeticiones > maxre)
53         {
54             m = Arreglo[i];
55             maxre = repeticiones;
56         }
57         i++;
58         repeticiones = 0;
59     }
60     cout <<"La moda es: "<<m<<"\n que aparece: "<<maxre+1<<"veces"<<endl;

```

Aquí se muestra como se calcula el promedio que no tiene mayor lógica que un ciclo que lo recorra y que en la variable s se vaya sumando cada numero recorrido, para que al terminar la suma se divida entre la cantidad de números ingresados y ya sale el promedio.

La moda es mas interesante y me hizo pensar mas, pero la lógica esta simple después de entenderle, yo inicialice un while para recorrer la lista una vez, pero inicialice otro que recorra la lista por cada elemento que hay, entonces empieza con el primer elemento y va avanzando de uno en uno y comparándolos con el if si el arreglo de i en este caso el primer elemento (depende donde vaya el ciclo primero) y lo compara con el que sigue y así, además de que diferencia si es el mismo espacio del arreglo [n] o si es diferente para tomarlo en cuenta, al final solo se pone un contador cada que el numero se repite, y se compara cada que acaba una vuelta si ese numero logro tener el máximo de repeticiones, si es así se guarda en la variable moda, al final se actualiza el numero máximo de repeticiones del numero, y se inicializa desde 0 el contador de las repeticiones.

```

62     int modasre[50];
63     int indice = 0, modas;
64     modas = maxre;
65     repeticiones = 0, maxre = 0;
66     modasre[0] = m, i = 0;
67     bool bandera = true;
68
69     while( i < cantidad )
70     {
71         int j = 0;
72         while (j < cantidad)
73         {
74             if (Arreglo[i] == Arreglo[j] && i != j)
75             {
76                 repeticiones++;
77             }
78             j++;
79         }
80         if (repeticiones > maxre)
81         {
82             m = Arreglo[i];
83             maxre = repeticiones;
84         }

```

```

86     if (repeticiones == modas)
87     {
88         int r = 0;
89         while (r <= indice)
90         {
91             if (modasre[r] == Arreglo[i])
92             {
93                 bandera = false;
94             }
95             r++;
96         }
97
98         if(bandera == true)
99         {
100             indice++;
101             modasre[indice] = Arreglo[i];
102             cout <<"\n La moda es: "<<modasre[indice]<<"\n que aparece: "<<maxre+1<<"veces"<<endl;
103         }
104     }
105
106     bandera = true;
107     repeticiones = 0;
108     i++;
109 }
110

```

Aquí es el punto mas complicado y es que cuando ponía 2 modas o mas solo me ponía la primer moda de esta lista, por ende pensando llegue a la conclusión de este código, repetí lo mismo que el primero, con la diferencia de que le agregue un array que guarde las modas repetidas, entonces le agregue otra condición de si las repeticiones de una iteración de la lista son iguales al numero de repeticiones máximo de las modas (la anterior pagina) entra en el while que busca si la moda que se tiene es igual a la que están comparando entonces simplemente manda falso y continua, esto para evitar que se clonen los resultados, y abajo se tiene la otra bandera que en caso de que de verdadero, el índice aumenta y se guarda en el arreglo de i.

```

#include <iostream>
#include <string.h>
#define MaxData 100

using namespace std;

int Arreglo [MaxData];
void inserta(int cantidad){
    if (cantidad < 0 || cantidad > MaxData){

        cout<<"Ingresa un elemento valido"<<endl;
        system("pause");
        return;
    }

    for(int i = 0; i < cantidad; i ++){
        cout <<" introduce el elemento :"<<endl;
        cin >> Arreglo[i];
    }
}

void imprimeLista(int cantidad){
    int i;
    for(i = 0; i < cantidad; i ++){
        cout<<" posicion :"<<i<<endl;
        cout<<" dato: "<<Arreglo[i]<<endl;
    }
}

void Promedio(int cantidad){
    int i,s = 0,prom;
    for(i = 0; i < cantidad; i ++){
        s = s+Arreglo[i];
    }
    prom = s/cantidad;
    cout << "\n suma:"<<s<<"\n promedio:"<<prom<<endl;
}

void moda(int cantidad){
    int i = 0,maxre = 0;
    int m, repeticiones = 0;
    while( i < cantidad )
    {
        int j = 0;
        while (j < cantidad)
        {
            if (Arreglo[i] == Arreglo[j] && i != j)
            {
                repeticiones++;
            }
            j++;
        }
        if (repeticiones > maxre)
        {

```

```

        m = Arreglo[i];
        maxre = repeticiones;
    }
    i++;
    repeticiones = 0;
}
cout <<"La moda es: "<<m<<"\n que aparece: "<<maxre+1<<"veces"<<endl;

int modasre[50];
int indice = 0, modas;
modas = maxre;
    repeticiones = 0, maxre = 0;
    modasre[0] = m, i = 0;
    bool bandera = true;

    while( i < cantidad )
    {
        int j = 0;
        while (j < cantidad)
        {
            if (Arreglo[i] == Arreglo[j] && i != j)
            {
                repeticiones++;
            }
            j++;
        }

        if (repeticiones > maxre)
        {
            m = Arreglo[i];
            maxre = repeticiones;
        }

        if (repeticiones == modas)
        {
            int r = 0;
            while (r <= indice)
            {
                if (modasre[r] == Arreglo[i])
                {
                    bandera = false;
                }
                r++;
            }

            if(bandera == true)
            {
                indice++;
                modasre[indice] = Arreglo[i];
                cout <<"\n La moda es: "<<modasre[indice]<<"\n
que aparece: "<<maxre+1<<"veces"<<endl;
            }
        }
    }
}

```

```

        bandera = true;
        repeticiones = 0;
        i++;
    }
}

int main(){
    int opcion, cantidad;
    while(opcion!=5){
        cout<<"\n Que opcion deseas, \n5-Salir \n1-insertar elementos \n2-
imprime lista \n3-Promedio y suma \n4-Moda \nOpcion: "<<endl;
        cin>>opcion;

        switch(opcion){
            case 1: system("cls");
                    cout<<"Inserta la cantidad de elementos a insertar
:"<<endl;cin>>cantidad;
                    inserta(cantidad);
                    break;
            case 2: system("cls");
                    imprimeLista(cantidad);
                    break;
            case 3: system("cls");
                    Promedio(cantidad);
                    break;
            case 4: system("cls");
                    moda(cantidad);
                    break;
            case 5: break;
            default: printf("Esa opcion no existe...\n");
        }
    }

    return 0;
}

```