

PRACTICA N° 1

Nombre: Fernando Manuel Menendez Copari

1. Diseñar un algoritmo que determine si una cadena es un palíndromo (sin usar funciones integradas).

<https://onlinegdb.com/YynRU4zwb>

```
#include <iostream>
using namespace std;

int main(){
    int n;
    cin>>n;
    int copia;
    copia=n;
    int reversa = 0;
    int temp = 0;

    while(copia>0)
    {
        temp = copia%10;
        copia = copia / 10;
        reversa = reversa + temp;
        if(copia > 0)
            reversa = reversa * 10;
    }
    if(n == reversa)
        cout << "si es";
    else
        cout << "no es";
}
```

n = 12321

copia	reversa	temp
1 2 3 2 1	0	0
1 2 3 2	1 0	1
1 2 3	1 2 0	2
1 2	1 2 3 0	3
1	1 2 3 2 0	2
0	1 2 3 2 1	1

• reversa = 12321 = n

=> si es

2. Elaborar un algoritmo que recorra un arreglo de N elementos y determine el segundo valor más alto sin ordenarlo.

<https://onlinegdb.com/ahG1FWIXj>

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      cin>>n;
8      int cad[n];
9      int sec_mayor=0;
10     int mayor=0;
11
12     for(int i=0;i<n;i++)
13     {
14         cin>>cad[i];
15     }
16     //ALGORITMO
17     for(int i=0;i<n;i++)
18     {
19         if(cad[i]>mayor)
20         {
21             sec_mayor=mayor;
22             mayor=cad[i];
23         }
24
25         if(cad[i]<mayor && cad[i]>sec_mayor)
26         {
27             sec_mayor=cad[i];
28         }
29     }
30
31     cout<<"el mayor es : "<<mayor<<endl;
32     cout<<" el segundo mayor es : "<< sec_mayor;
33
34 }
```

2. Elaborar un algoritmo que recorra un arreglo de N elementos y determine el segundo valor más alto sin ordenarlo.

n = 5
[7, 2, 9, 3, 5]

mayor	sec_mayor	i	cad[i]
0	0	-	-
7	0	0	7
7	2	1	2
9	7	2	9
9	7	3	3
9	7	4	5

3. Simular una calculadora de tarifas para transporte público basada en el tipo de usuario y distancia recorrida.

<https://onlinegdb.com/d0juPHVZW>

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string tipo_usuario;
6      float distancia_km;
7      float tarifa_por_km = 0.0;
8      float total;
9
10     cout << "Ingrese el tipo de usuario (Escolar, Universitario, Adulto): ";
11     cin >> tipo_usuario;
12
13     cout << "Ingrese la distancia recorrida en km: ";
14     cin >> distancia_km;
15
16     if (tipo_usuario == "Escolar" || tipo_usuario == "escolar") {
17         tarifa_por_km = 0.10;
18     } else if (tipo_usuario == "Universitario" || tipo_usuario == "universitario") {
19         tarifa_por_km = 0.14;
20     } else if (tipo_usuario == "Adulto" || tipo_usuario == "adulto") {
21         tarifa_por_km = 0.20;
22     } else {
23         cout << "Tipo de usuario no válido." << endl;
24         return 1;
25     }
26
27     total = tarifa_por_km * distancia_km;
28
29     cout << "La tarifa total a pagar es: S/ " << total << endl;
30
31     return 0;
32 }
33
```

3) Simular una calculadora de tarifas para transporte público basada en el tipo de usuario y distancia recorrida.

Tipo	tarifa por Km	distancia Km	total
Escolar	0.10	10	S/ 1.00
Universitarios	0.14	20	S/ 2.80
Adulto	0.20	15	S/ 3.00