

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍAS DE SISTEMAS



PRACTICA 1-TABULACIÓN

PRESENTADO POR:

EVELYN YANET CURO YAGUNO

CODIGO: 227880

DOCENTE:

ALDO HERNAN ZANABRIA GALVEZ

CURSO:

ALGORITMOS Y ESTRUCTURAS DE DATOS

SEMESTRE:

IV

GRUPO: "A"

Puno – PERÚ

ABRIL 2025

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

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n, copia, reversa = 0;
6     cout << "Ingrese un numero: ";
7     cin >> n;
8     copia = n;
9
10    while (n > 0) {
11        reversa = reversa * 10 + n % 10;
12        n /= 10;
13    }
14
15    if (copia == reversa)
16        cout << "Si es palindromo";
17    else
18        cout << "No es palindromo";
19 }
```

Código:

```
1. #include <iostream>
2. using namespace std;
3. int main () {
4.     int n, copia, reversa = 0;
5.     cout << "Ingrese un numero: ";
6.     cin >> n;
7.     copia = n;
8.
9.     While (n>0) {
10.         reversa = reversa * 10 + n % 10;
11.         n /= 10;
12.     }
13.     if (copia == reversa)
14.         cout << " Si es palindromo";
15.     else
16.         cout << "No es palindromo";
17. }
```

Tabulación:

Date: ____/____/____

$n = 47874$

ITERACIÓN	copia	temp = copia	reversa antes	reversa después
1	47874	4	0	4
2	4787	7	4	47
3	478	8	47	478
4	47	7	478	4787
5	4	4	4787	47874
6	0	-	-	(termina el ciclo)

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

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

Código:

```
int n; cin >> n;
```

```
int cad[n];
```

```
int sec_mayor = 0, mayor = 0;
```

```
for (int i = 0; i < n; i++)
```

```
    cin >> cad[i];
```

```
for (int i = 0; i < n; i++) {
```

```
    if (cad[i] > mayor) {
```

```
        sec_mayor = mayor;
```

```
        mayor = cad[i];
```

```
    }
```

```
    if (cad[i] < mayor && cad[i] > sec_mayor)
```

```
        sec_mayor = cad[i];
```

```
}
```

```
cout << "El mayor es:" << mayor << endl;
```

```
cout << "El segundo mayor es:" << sec_mayor;
```

```
}
```

Tabulación:

Date: ____/____/____

$n = 6$		cad = [8, 12, 4, 6, 12, 10]	
i	cad[i]	mayor	Sec_mayor
-	-	0	0
0	8	8	0
1	12	12	8
2	4	12	8
3	6	12	8
4	12	12	8
5	10	12	10

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

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int usuario;
6      float distancia, tarifa, costo;
7
8      cout << "ingresa tipo de usuario (1-estudiante, 2-general, 3-adulto
9  mayor): ";
10     cin >> usuario;
11     cout << "ingresa distancia en km: ";
12     cin >> distancia;
13
14     if (usuario == 1) {
15         tarifa = 0.5;
16     } else if (usuario == 2) {
17         tarifa = 1.0;
18     } else if (usuario == 3) {
19         tarifa = 0.7;
20     } else {
21         cout << "tipo de usuario invalido\n";
22         return 0;
23     }
24
25     costo = tarifa * distancia;
26     cout << "el costo total es: " << costo << endl;
27 }
```


Código:

```
int usuario;  
  
float distancia, tarifa, costo;  
  
cout << "ingresa tipo de usuario (1-estudiante, 2-general,  
3-adulto mayor) :";  
  
cin >> usuario;  
cout << "ingre la distancia en km:";  
cin >> distancia;  
  
if (usuario == 1) {  
    tarifa = 0.5;  
  
} else if (usuario == 2) {  
    tarifa = 1.0;  
  
} else if (usuario == 3) {  
    tarifa = 0.7;  
  
} else {  
    cout << "tipo de usuario invalido \n";  
    return 0;  
}  
  
Costo = tarifa * distancia;  
  
cout << "el costo total es!" << costo << endl;  
  
}
```

Tabulación:

Date: ____/____/____

Tipo de usuario	Distancia (km)	tarifa (por km)	Costo total (resultado)
Estudiante (1)	10	0.5	5.0
Estudiante (1)	25	0.5	12.5
General (2)	10	1.0	10.0
General (2)	15	1.0	15.0
Adulto mayor (3)	10	0.7	7.0
Adulto mayor (3)	30	0.7	21.0
Usuario no valido	10	—	—