UNIVERSIDAD NACIONAL DEL ALTIPLANO DE PUNO

Facultad de ingeniería mecánica eléctrica, electrónica y sistemas

Escuela profesional de ingeniería de sistemas

Trabajo en clase 1 y tabulaciones

PRESENTADO POR:

- Ccallo Arpita Jhak Esnayder

CURSO:

ALGORITMOS Y ESTRUCTURAS DE DATOS

DOCENTE:

ZANABRIA GALVEZ ALDO HERNAN

SEMESTRE IV

PUNO – PERÚ

2025

Actividad en clase y tabulaciones en papel

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

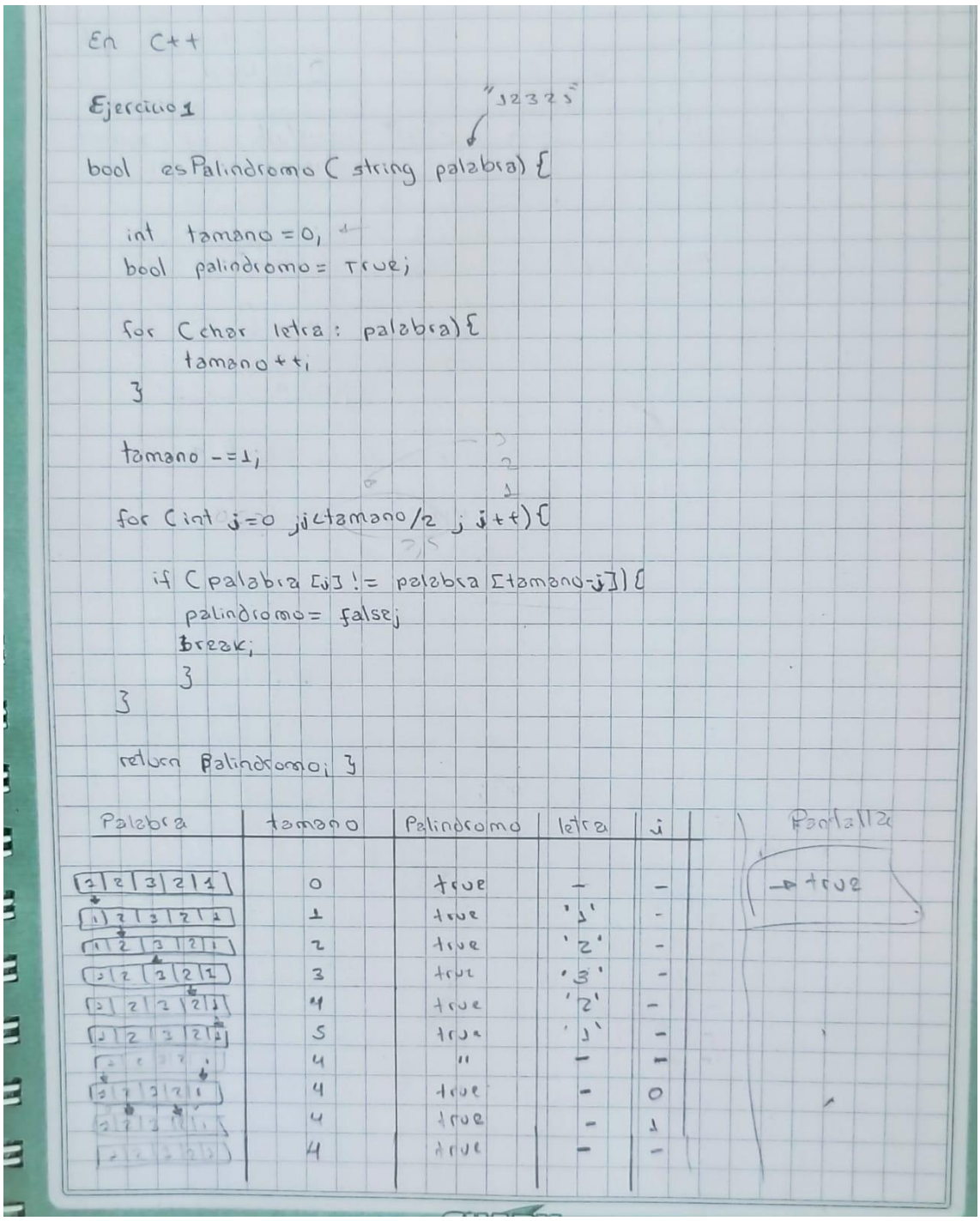
Código:

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

bool esPalindromo(string palabra){

    int tamano=0;
    bool palindromo = true;

    for(char letra : palabra){
        tamano++;
    }

    tamano-=1;

    for(int i=0; i<(tamano)/2; i++){

        if(palabra[i]!= palabra[tamano-i]){
            palindromo=false;
            break;
        }
    }

    return palindromo;
}
```

```cpp
string conMinuscula(string palabra){
    int cont=0;
    for(char letra: palabra){

        if(letra>='A' && letra<='Z'){
            palabra[cont]+=32;
        }

        cont++;
    }

    return palabra;
}
```

```cpp

int  main(){

    string palabra;
    cout<<"Ingrese una palabra: ";
    cin>>palabra;

    if(esPalindromo(conMinuscula(palabra))){
        cout<<"la palabra es un palindromo"<<endl;
    }
    else{
        cout<<"la palabra no es un palindromo"<<endl;
    }

    cout<<palabra;

    return 0;
}
```

Código y tabulación en papel



En C++

Ejercicio 1

"12321"

```
bool esPalindromo ( string palabra) {

    int tamano =0,
    bool palindromo= true;

    for (char letra : palabra){
        tamano++;
    }

    tamano -=1;

    for (int j=0 ;j<tamano/2 ; j++){

        if (palabra [j] != palabra [tamano-j]) {
            palindromo= false;
            break;
            }
    }

    return palindromo; }
```

Palabra	tamano	Palindromo	letra	j
1 2 3 2 1	0	true	-	-
1 2 3 2 1	1	true	'1'	-
1 2 3 2 1	2	true	'2'	-
1 2 3 2 1	3	true	'3'	-
1 2 3 2 1	4	true	'2'	-
1 2 3 2 1	5	true	'1'	-
1 2 3 2 1	4	"	-	-
1 2 3 2 1	4	true	-	0
1 2 3 2 1	4	true	-	1
1 2 3 2 1	4	true	-	-

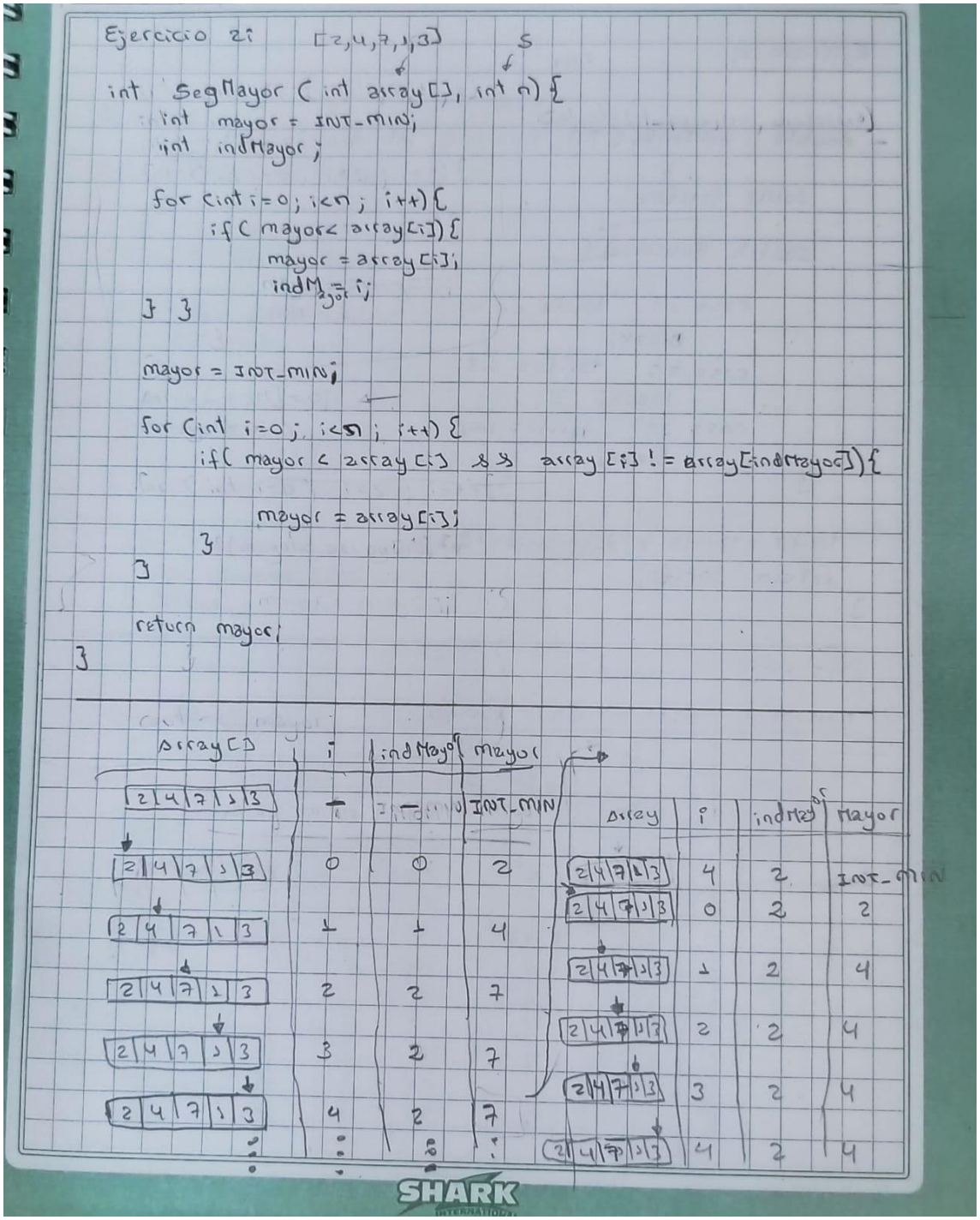
Pantalla → true

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

Código:

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

int segMayor(int array[], int n){

    int mayor=INT_MIN;//INT_MIN es una constante definida en <cl
    int indMayor;
    for(int i=0;i<n; i++){
        if(mayor<array[i]){
            mayor=array[i];
            indMayor=i;
        }
    }

    mayor=INT_MIN;

    for(int i=0;i<n; i++){
        if(mayor<array[i] && array[i]!=array[indMayor]){
            mayor=array[i];
        }
    }

    return mayor;

}
int  main(){
    int n;
    cout<<"introduzca el tamano del array: ";
    cin>>n;
    int array[n];
    cout<<"introduzca los datos del array: ";
    for(int i=0; i<n; i++){
        cin>>array[i];
    }

    cout<<"El segundo numero mayor es: "<<segMayor(array,n);

    return 0;
}
```

Código y Tabulación a mano



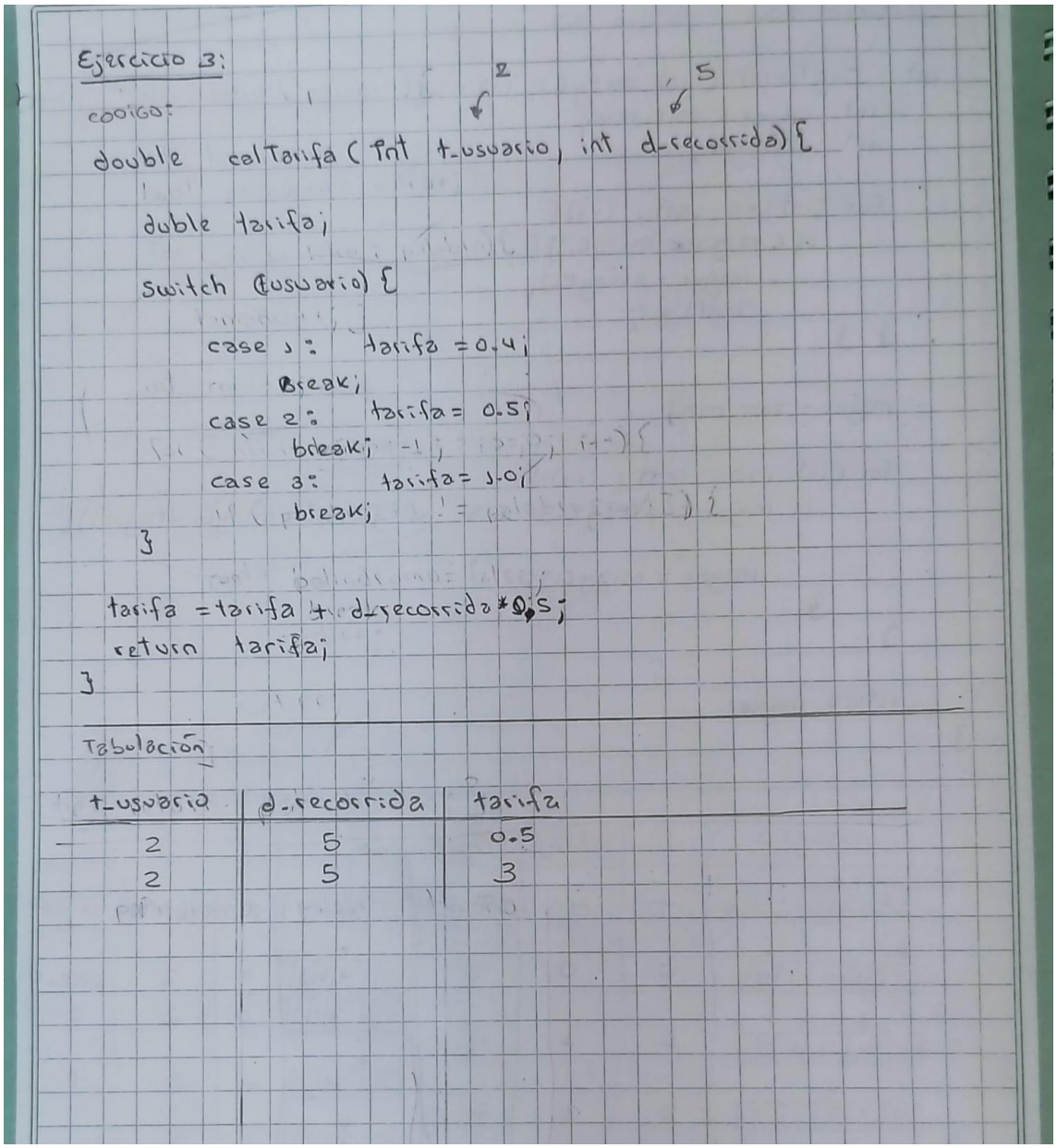
3. Simular una calculadora de tarifas para transporte público basada en el tipo de usuario y distancia recorrida.

Código:

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

//el tipo de usuario define el valor base de la tarifa


double calTarifa(int t_usario, int d_recorrida){
    double tarifa;
    //definimos la tarifa base
    switch(t_usario){
        case 1:
            tarifa=0.4;
            break;
        case 2:
            tarifa=0.5;
            break;
        case 3:
            tarifa=1.0;
            break;
    }

    tarifa = tarifa + d_recorrida*1;

    return tarifa;
}
```

```cpp
int  main(){

    int t_usario, d_recorrida;

    cout<<"Indique el tipo de usario: "<<endl;
    cout<<"1. escolar "<<endl;
    cout<<"2. universitario: "<<endl;
    cout<<"3. adulto "<<endl;
    cin>>t_usario;

    cout<<"Indique la distancia recorrida en km: "<<endl;
    cin>>d_recorrida;

    cout<<"la tarifa es de: "<<calTarifa(t_usario,d_recorrida)<<"soles ";


    return 0;
}
```

Código y tabulación a mano:



Ejercicio 3:

codigo:

```
double calTarifa (int t_usuario, int d_recorrida){
    duble tarifa;
    switch (tusuario) {
        case 1:  tarifa =0.4;
            Break;
        case 2:  tarifa= 0.5;
            break;
        case 3:  tarifa= 1.0;
            break;
    }
  tarifa =tarifa + d_recorrida*0,5;
  return tarifa;
}
```

Tabulación

t_usuario	d_recorrida	tarifa
2	5	0.5
2	5	3