

Curso: SIS210 – Algoritmos y Estructuras de Datos

Docente: Ing. Zanabria Galvez Aldo Hernan

Alumno: Flores Macedo Anderson Leonardo

Código: 236177

Pseudocódigo

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

Definir cadena texto\_az, texto\_z;

Imprimir ( "Ingrese el texto a ser evaluado: " );

Ingresar ( texto\_az );

Para n = 0 hasta n < tamaño( texto\_az ), avanzar en 1{

    texto\_z = texto\_z + texto\_az[n];

}

Si ( texto\_az == texto\_z ){

    Imprimir ( "La palabra es palíndroma" );

}

Sino{

    Imprimir ( "La palabra no es palíndroma" );

}

Su complejidad temporal es  $O(n + 5)$

He usado un arreglo o cadena, ya que es mas manipulable y aprovecho este para invertir el orden de la palabra, resolviendo el problema.

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

Definir entero N, numA, numB;

Ingresar ( N );

Definir entero numbers[N];

numA = numbers[0];

numB = numbers[1];

Si ( numB > numA ){

    numB = numbers[0];

```

        numA = numbers[1];
    }

    Para l = 0 hasta l < N, avanzar en 1{
        Si numbers[N] > numA{
            numB = numA;
            numA = numbers[N];
        }
    }

    Si ( numA == numB ){
        Imprimir ( "No hay un segundo número mayor" );
    }
    Sino{
        Imprimir ( "El segundo número mayor es: " + numB );
    }

```

Su complejidad temporal es  $O(n + 10)$

He usado un arreglo, ya que este servirá como lista para calcular el segundo número mayor, y aprovechando que el próximo número puede ser mayor, significa que el segundo es menor.

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

```

Definir entero usuario, distancia, costedistancia;
Definir flotante costefinal;
Imprimir ( "Tipo de Usuario\n1.Escolar: 0.5\n2.Adulto: 1.0\nElección: " );

Ingresar ( usuario );

Imprimir ( "1000m – 1.00\n2000m – 1.50\n3000m – 2.00\netc.\nDistancia recorrida: " );
Ingresar ( distancia );

costedistancia = distancia / 1000;

costefinal = usuario * 0.5 + costedistancia * 0.5 + 1;
imprimir ( "El coste de este viaje es: ", costefinal );

```

Su complejidad temporal es  $O(9)$

He usado una estructura normal, ya que se tiene definido los precios de la tarifa del transporte público.