

# AyED - Trabajo Práctico N° 2

Mediante un menú de opciones, resolver los siguientes 3 enunciados utilizando datos estructurados tipo array.

Menú

- 1) **Encriptar mensajes**
- 2) **Robot caminando**
- 3) **Detectar palíndromo**
- 4) **Salir**

- 1) Se necesita encriptar un mensaje ingresado por teclado, con una longitud máxima de 80 caracteres, los cuales pueden ser números, letras y/o símbolos. Para ello se tomará desde el 2º hasta el 5º espacio en blanco, para escribir dicho contenido en forma inversa.
- Si el mensaje tiene 1 espacio en blanco, no se encriptará.
- Si tiene más de 2 espacios pero menos de 5, se encriptará desde el 2º espacio hasta el final del mensaje.

Ejemplo:

hoy nos **juntamos en casa** de juan.

debería quedar

hoy nos **asac ne somatnuj** de juan.

- 2) Un pequeño robot se mueve respondiendo a las órdenes de un bastón de mando. El robot recorre una cuadrícula de F filas por C columnas y no puede salirse de esos límites. Si lo imaginás como un mapa podemos hablar de **Norte** arriba, **Sur** abajo, **Oeste** a la izquierda y **Este** a la derecha.

Las filas se numeran de abajo hacia arriba con números naturales:

1, 2, ..., F.

Las columnas se numeran de izquierda a derecha con números naturales:

1, 2, ..., C.

Las órdenes que emite el bastón de mando están codificadas con letras:

- N** avance un cuadradito en dirección Norte
- S** avance un cuadradito en dirección Sur
- E** avance un cuadradito en dirección Este
- O** avance un cuadradito en dirección Oeste.

Sin embargo hay una excepción. Cuando está en un borde y recibe una orden para exceder ese límite entonces esa orden es ignorada.

El dueño del robot quiere saber adónde irá a parar el robot, después de ejecutar una secuencia de mandos, a partir de cierta posición inicial.

Para ello se te solicita un programa que dada una posición inicial del robot y una serie de órdenes enviadas por el bastón de mando, determine la posición final del robot, mostrando el recorrido paso a paso (ejemplo poniendo X en la casilla de la matriz que ocupa el robot, y cambiándola por una x cuando el robot la abandona).

### **Restricciones:**

Sobre el número de filas  $0 < F < 20$

Sobre el número de columnas  $0 < C < 20$

Largo máximo de la secuencia de órdenes 25.

### **Datos de entrada**

- columna donde está inicialmente el robot
- fila donde está inicialmente el robot
- la secuencia de órdenes dadas por el bastón de mando, sin separación entre ellas.

Ejemplo:

Parte desde **Fila = 5, columna = 3**, con la **secuencia = NNNNNNNNOO**

Quedaría en la posición **Fila = 12, columna = 1**

A partir de esta posición, puede recibir otra secuencia de órdenes, considerando esta [fila, columna] como dato de inicio para su nuevo rumbo. Establecer un fin de datos acorde.

- 3) Una palabra o conjunto de letras (que puede no tener significado), se llama palíndromo cuando teniendo más de una letra, se lee lo mismo de izquierda a derecha ó de derecha a izquierda; por ejemplo, **ababa**.

También llamaremos i-palíndromo cuando quitando el primer carácter de la izquierda se convierte en palíndromo, por ejemplo **casa**.

Se llama d-palíndromo cuando quitando el primer carácter de la derecha se convierte en palíndromo, por ejemplo **amad**.

Llamaremos palabras distinguidas a aquellas que son palíndromos, i-palíndromo ó d-palíndromo.

El problema, cuya solución debes programar, consiste en recibir una palabra y determinar los posibles cortes que la descomponen en dos palabras distinguidas, e indicar para cada una de ellas de qué tipo son.

La palabra a cortar ingresa por teclado con una longitud máxima de 20 caracteres.

Exhibir, separado por espacios, una palabra distinguida y los tipos a los que pertenece. Si no hay ninguna descomposición posible se colocará “no se puede”.

Ejemplo 1:

Si la palabra es **azarosos** - Debería mostrarse:

**azar** d-palindromo

**osos** i-palindromo d-palindromo

Ejemplo 2:

Si la palabra es **amarrar** - Debería mostrarse:

**ama** palindromo

**rrar** i-palindromo

**amar** d-palindromo

**rar** palindromo

## NOTA

- Será muy valorado un buen diseño de las pantallas que presente el programa.
- Cada una de las opciones debe dar la posibilidad de ser ejecutada más de una vez.
- El robot puede hacer varios recorridos, donde cada uno se lo podría ver en distintos colores ó en distintos tipos de símbolos, a medida que se genera cada paso.