2º Cuatrimestre 2013 **Ejercicio Nº 2 - Detalle de recorrido de matrices**	**Taller de Programación I (75.42)** **Facultad de Ingeniería** **Universidad de Buenos Aires**

Detalle de recorrido de matrices

Ejercicio N°2

Introducción

Dada una matriz de caracteres (cargada a partir de un archivo de texto) y una serie de movimientos (up,down,left,right), se deberá recorrer la matriz y mostrar por salida estándar los valores encontrados en cada posición y la cantidad de movimientos válidos realizados.
El desarrollo del trabajo permitirá aplicar y afianzar conceptos y conocimientos del lenguaje de programación C, en particular la utilización de memoria dinámica, TDAs[1], estructuras de datos y manejo de archivos.

Descripción

Se deberá implementar un programa de consola que reciba un path a un archivo de texto plano y un segundo parámetro opcional con el path a un archivo con datos del recorrido y tamaño de la matriz.

Formato de Línea de Comandos

./tp archivo_texto_plano [arch_opc]

Si no se especifica el segundo archivo se obtendrá la información por entrada estándar: se ingresará un tamaño de matriz y una serie de movimientos a realizar.
Formato: numero_filasXnumero_columnas movimientos
Los movimientos posibles son U:up, D: down, L:left y R:right; no hay movimientos en diagonal y tampoco hay movimientos circulares (si se encuentra en la primera fila y el siguiente movimiento es U, no se mueve a la última fila, sino que se considera un movimiento inválido.

Carga de matriz:

El archivo que se recibe como primer parámetro contiene caracteres que se deberán utilizar para cargar una matriz (usar memoria dinámica).
Ejemplo de ejecucion:

./tp holamundo.txt

holamundo.txt contiene: "hola mundo"

por entrada standard se ingresa: 2X2 RLDU

La matriz se carga con:

```
ho
la
```

Una vez cargada la matriz se procederá a recorrerla usando los movimientos recibidos. El contenido de cada posición deberá ser volcado a una lista (implementada por el alumno). Se comienza por la posición (1,1), la cual no se contabiliza.
Luego, por salida estándar se deberá indicar la cantidad de movimientos válidos realizados, seguido del listado con el camino recorrido en la matriz.
Por salida estándar se imprimirá el resultado: 4 ohlh
casos especiales:

*Movimiento inválido:
En caso de no poder realizar un movimiento, este paso no deberá contabilizarse y en la lista se agregará el valor de movimiento inválido [E]
2X2 RLUD

la salida del programa seria: 3 oh[E]l

*Caracter no válido para realizar un movimiento:
Se ignora el caracter y se prosigue con el resto.

*Archivo de texto de tamaño menor que la matriz:

En este caso se deberá completar la matriz realizando un round robin[2]

Ejemplo:
4X4 RLDU

```
hola
mun
doho
la m
```

Códigos de Retorno

El programa retorna 0 en todos los casos.

Ejemplos de Ejecución

Ejemplo con 2 parámetros:

./tp simple in_simple

simple contiene:
“The Three Laws are: A robot may not injure a human being or, through inaction, allow a human being to come to harm. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.”

in_simple contiene:
2X8 RLDDLRUUURL
4X4 RLDDLRUUURL

para la 1era opción la matriz quedaría:

```
The Thre
e Laws a
```

para la 2da opción la matriz quedaría:

```
The
Thre
e La
ws a
```

salida:
7 hTe[E][E] h[E][E]eh
9 hTTe[E] hh[E]eh

Restricciones

La siguiente es una lista de restricciones técnicas exigidas:

1. El sistema debe desarrollarse en ISO C (C99).
2. Se puede asumir que el archivo de entrada existe y es siempre correcto.
3. Está prohibido el uso de variables globales
4. La resolución del trabajo debe estar orientada al uso de TDAs (tipos de datos abstractos), generando

estructuras tales como listas, colas, etc. que provean una interfaz que oculte la implementación; está terminantemente prohibido violar el concepto de abstracción.
5. Debe utilizarse la cantidad necesaria de memoria, apelando a la reserva en forma dinámica cuando sea necesaria; no se considerará correcto utilizar memoria en exceso.
6. La implementación del algoritmo para la resolución del problema tiene que ser de autoría del alumno que resuelve el ejercicio.

Referencias

[1] http://es.wikipedia.org/wiki/Tipo_de_dato_abstracto
[2] http://es.wikipedia.org/wiki/Planificaci%C3%B3n_Round-robin