

1 - Librero (librero.txt)

por: Fernando Guzmán

Primaria :: Secundaria :: OMI :: Preuniver

Karelman es muy trabajador así como Tú Olímpico!!!, pero a veces el Face y su nueva compu le quitan mucho tiempo y ha cometido errores que antes no realizaba, en esta ocasión le ha sido asignada la tarea de simplemente levantar un librero pero lleva toda la mañana sin poder lograrlo. Tú te has dado cuenta de esto y decides ser su “San Karel” y ayudarlo en momento crucial.

Problema:

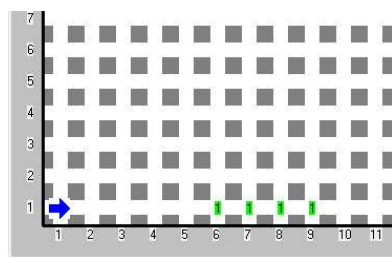
La tarea de Karelman consiste en levantar un librero del piso, así de simple.

Condiciones:

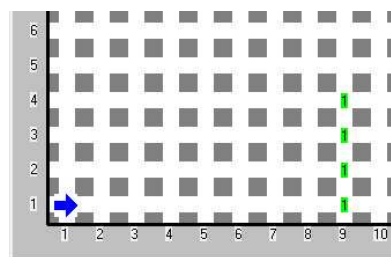
- Karelman inicia en su casa orientado al este (posición 1,1) y no tiene zumbadores en su mochila.
- Gira-Izquierda; Avanza;
- El librero esta representado por zumbadores.
- No importa donde termine Karelman.
- Karelman deberá levantar el librero en el mismo lugar donde esta la base del mismo.
- El librero siempre esta en forma horizontal sobre el piso.
- Siempre habrá un librero.

Ejemplo 1:

ENTRADA



SALIDA



2 - Piramide (piramide.txt)

por: Javier Padilla

Primaria :: Secundaria :: OMI :: Preuniver

HISTORIA

Karel y su familia han salido de vacaciones rumbo a la zona arqueológica de Karel Itzá y en su estancia han decidido escalar todas las pirámides; pero Karel se ha quedado atrás y ahora no sabe como llegar a la cima de las diferentes pirámides.

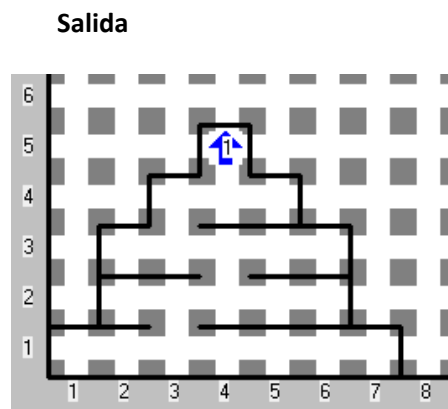
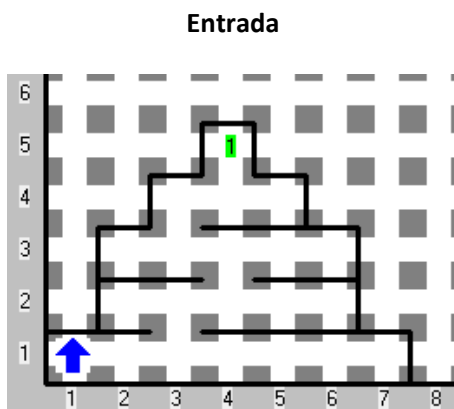
PROBLEMA

Ayuda a Karel a subir a lo más alto de cada pirámide que se encuentre en su camino, la cima estará indicada con un zumbador.

CONSIDERACIONES

- ❖ Karel inicia sin zumbadores en la mochila.
- ❖ Karel siempre inicia en la parte interna inferior de la pirámide, pegado en el lado izquierdo y orientado al norte.
- ❖ Cada pirámide tiene diferente altura.
- ❖ Karel debe terminar orientado al norte en la posición donde está el zumbador que indica la cima de la pirámide.
- ❖ Karel Itzá es una nueva maravilla del mundo.
- ❖ Solo existe una puerta para subir al siguiente nivel.
- ❖ Pueden existir niveles del mismo largo que el nivel anterior.
- ❖ Se desconoce la posición de las puertas en cada pirámide.

Ejemplo:



3 - Kambiandose (kambian.cpp)

por: José Luis Roa

Universitarios

HISTORIA

Karel esta súper emocionado ya que Grettel le dijo que si iba a salir al cine con el, por lo que ahora su problema es escoger la ropa que debe ponerse, más sin embargo no sabe cuantas combinaciones de ropa posible puede realizar.

PROBLEMA

Ayuda a Karel a saber por medio de un programa escrito en C/C++ de cuantas formas puede vestirse de modo que cada conjunto de ropa difiera de los demás en por lo menos un elemento.

ENTRADA

- En la primera línea: C casos que Karel puede tener en el problema, donde $1 \leq C \leq 1'000,000$
- *Línea 2 hasta C+1 (es un caso diferente por línea):* N prendas de ropa que tiene Karel en su armario donde $1 \leq N \leq 100$ y separado por un único espacio el valor R que es el numero de prendas que Karel quiere usar a la vez donde $1 \leq R \leq 100$ y $R \leq N$.

SALIDA

- C líneas indicando en cada una el numero de formas en que se puede vestir Karel, una línea por cada caso expuesto en la entrada

EJEMPLOS

ENTRADA	SALIDA
2	3
3 2	1
10 10	

4 - Kaminos (kaminos.cpp)

por: José Luis Roa

Universitarios

HISTORIA

A Karel le gusta mucho los juegos del tipo sopa de letras, pero es tanta su afición que ya ninguna sopa de letras es reto para él. Por lo que decidió diseñar un nuevo juego llamado k-minos.

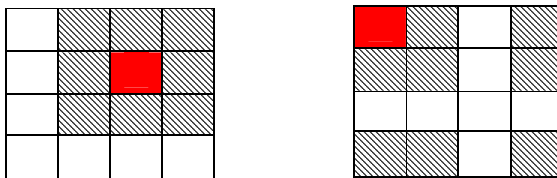
PROBLEMA

Karel ha establecido un tablero de dimensiones $M * N$ donde $1 \leq M \leq 99$ y $1 \leq N \leq 99$, tal que cada casilla contenga una letra mayúscula. La casilla que está en la fila m y la columna n la identificamos mediante (m,n) . Dos casillas diferentes (m_i, n_i) y (m_j, n_j) son adyacentes si se cumple:

Para la primera componente, $|m_i - m_j| \leq 1$ o $|m_i - m_j| = M - 1$

Para la segunda componente, $|n_i - n_j| \leq 1$ o $|n_i - n_j| = N - 1$.

Es decir, son adyacentes todas aquellas casillas que rodean a una dada, considerando que en el tablero como si la última fila estuviera unida a la primera, y lo mismo para las columnas. En el dibujo siguiente están sombreadas las casillas adyacentes a las casillas marcadas con rojo en un tablero de 4×4 :



Dada una palabra de k letras mayúsculas $A = a_1, a_2, \dots, a_k$, $k \geq 1$, decimos que A está contenida en el tablero si se cumple que:

Existe una casilla (m_1, n_1) que contiene la letra a_1

Para cada letra a_{i+1} donde $1 \leq i < k$, existe una casilla (m_{i+1}, n_{i+1}) que contiene a_{i+1} cumpliéndose que (m_i, n_i) y (m_{i+1}, n_{i+1}) son casillas adyacentes en el tablero

No existen dos casillas (m_i, n_i) y (m_j, n_j) iguales, $1 \leq i, j \leq k$.

A la secuencia de casillas $(m_1, n_1), \dots, (m_k, n_k)$ la llamamos el camino de A en el tablero.

Así, dado el tablero 4x4 de la figura siguiente, las cadenas "SOLA", "HOLA" y "ADIOS" están contenidas en él, pero no sucede lo mismo con "GOZA", "HORA" ni "HALA".

S	H	A	Z
I	O	L	G
E	Z	E	F
O	H	D	I

En el caso de "SOLA", las casillas que forman su camino son (1,1), (2,2), (2,3) y (1,3). Para "HOLA", son (1,2), (2,2), (2,3) y (1,3). Para "ADIOS", el camino es (1,3), (4,3), (4,4), (4,1) y (1,1).

Dado un tablero de las características anteriormente descritas y una palabra A compuesta por letras mayúsculas, se pide calcular el camino de A. Al construir el programa, puedes suponer que A está contenida en el tablero y que existe un único camino para ella.

ENTRADA

Línea 1: valores de M y N separados por un único blanco

Líneas de la 2 a la M+1 (la línea k representa la fila k-1 del tablero): N caracteres, representando el contenido de la línea correspondiente del tablero

Línea M+2: p caracteres, $M \cdot N \geq p \geq 1$, que representa la palabra a tratar.

SALIDA

p líneas (una para cada letra de la palabra a tratar), siendo el contenido de la línea k igual a la casilla que aparece en posición k dentro del camino de la palabra, de esta forma: NUMERO ESPACIO NUMERO

EJEMPLO

ENTRADA	SALIDA
4 4	1 1
SHAZ	2 2
IOLG	2 3
EZEF	1 3
OHDI	
SOLA	