V Olimpiada de Informática del estado de Guanajuato
Segundo Cuasiexamen

El comité organizador te da la bienvenida al Segundo Cuasiexamen Práctico de la V Olimpiada de Informática del Estado de Guanajuato.

1) El examen tiene una duración de 4:30 horas.

2) El examen consiste en 4 problemas de programación en el ambiente “Karel”.

3) Tu carpeta de trabajo esta en “C:\OIEG\X”. X es tu nombre (nombre corto). Deberás nombrar cada programa con el nombre que se te indique respectivamente. Cada programa debe estar en una carpeta que lleve el mismo nombre del problema.

4) Debes hacer un programa para cada problema, el cual se evaluará en 10 casos de prueba. El puntaje que recibirás en cada problema, dependerá del número de casos de prueba que tu programa haya resuelto satisfactoriamente.

5) Todos los problemas valen el mismo puntaje. Cada caso de prueba tiene un valor de 1 punto.

6) EL CUASIEXAMEN NO CONTARÁ PARA LA PUNTUACIÓN DEL SELECTIVO.

7) No esta permitido el uso de libros, calculadoras, tablas o cualquier otro documento que el comité no te haya proporcionado.

8) Deberás crear un archivo de texto en tu carpeta de trabajo con el nombre de “Datos.txt”. Donde guardaras: *nombre completo*, *escuela*, *teléfono*, *correo electrónico*.

¡El comité de la OIEG te desea MUCHA SUERTE!

MCBD

Archivo: **mcbd.txt**

Historia

El famoso filósofo y matemático Karelópulos de Biperópolis no sólo aporto los fantásticos números triangulares (que ya viste en alguna ocasión). También descubrió el Máximo Común Beeper Divisor, mejor conocidos como MCBD, de dos montones de Beepers.

El MCBD de dos montones A y B de beepers, es el mayor montón de beepers C que puede dividir, tanto al montón A como al montón B, de manera exacta.

Problema

Debes realizar un algoritmo para que Karel encuentre el MCBD de dos montones A y B dados.

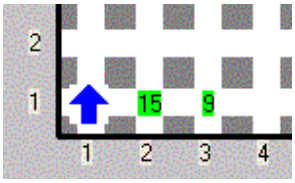
Consideraciones

1) Inicialmente Karel se encuentra en la esquina inferior izquierda de su mundo mirando al norte.

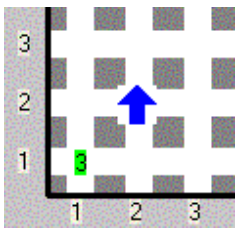
2) Los montones A y B se encuentran en la primera calle (horizontal), A esta en la segunda avenida (vertical) y B en la tercera.

3) El MCBD debe quedar en la esquina inferior izquierda.

4) No importa donde termine Karel ni su orientación sexual (y también geográfica).



Ejemplo de entrada



Salida

Ordenando

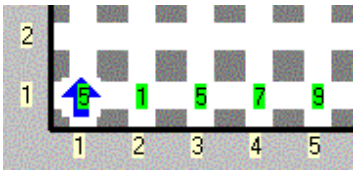
Archivo: **Ordenando.txt**

Historia

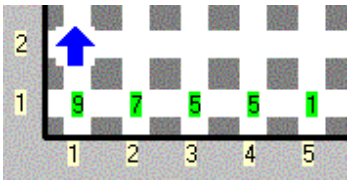
Karelópulos II (hijo del famoso Karelópulos) también tuvo diversas aportaciones a las matemáticas. Encontró un algoritmo para ordenar montones de beepers. Sin embargo, los escritos se han perdido con el paso del tiempo. Tú, como último descendiente directo de Karelópulos debes conservar su memoria.

Problema

Debes deducir de nuevo un algoritmo que ordene montones de beepers de mayor a menor que se encuentran sobre una calle (horizontal).



Ejemplo de entrada



Salida

Consideraciones

1) Inicialmente Karel se encuentra en la esquina inferior izquierda de su mundo mirando al norte.

2) Los montones de beepers se encuentran consecutivamente sobre la primera calle, iniciando desde la primera avenida.

3) El montón de mayor número de beepers debe estar en la esquina inferior izquierda del mundo de Karel

4) Todos los demás montones deben quedar sobre la primera calle de manera consecutiva y en orden descendente.

5) No hay paredes en el rectángulo delimitado por el origen y el punto: calle 2, avenida N. Donde N es el número de montones de beepers

Karel y Gretel Reload

Archivo: **GretelReload.txt**

Historia

Como era costumbre Gretel salió desde temprano (como a la 6 a.m.) al bosque en busca de biperleñas. Un poco más tarde (por eso de las 10 a.m.) salio Karel con el mismo propósito. Conociendo a Karel, Gretel tiene la sana costumbre de dejar beepers por donde ha pasado. Para la mala suerte de Karel, una biperave (un pájaro que come beepers) ha comido muchos beepers del camino aunque dejó los suficientes para que Karel pueda reconstruir el camino y encontrar a Gretel (era buena onda la biperave).

El camino que dejo Gretel (antes que pasara la biperave) esta hecho de manera que cada beeper está a sólo un paso del anterior y del siguiente (más no se sabe en que dirección) y no hay caminos que se cruzan. Además, no hay paredes sobre el camino y si se da un paso en la dirección equivocada, nunca habrá un beeper ahí.

Problema

Debes encontrar un algoritmo para que Karel reconstruya el camino de beepers que dejo Gretel, es decir, debes rellenar de nuevo el camino donde falten beepers.

Consideraciones

1) Inicialmente Karel se encuentra sobre el primer beeper.

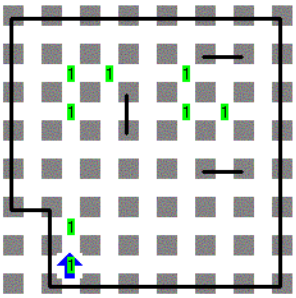
2) Las discontinuidades, en el camino de beepers, siempre están entre **un par de beepers alineados** del camino y no se sabe el largo de la discontinuidad.

3) Karel topará con Pared si equivoca la dirección.

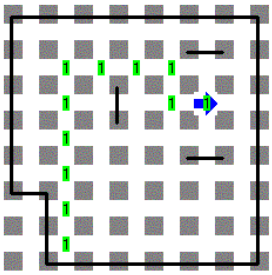
4) Karel termina si se topa con pared antes de un beeper, en cualesquiera de las direcciones disponibles (**sin contar de donde viene**) que tiene.

5) Karel debe terminar en el último beeper del camino con cualquier orientación.

6) Karel tiene suficientes beepers para colocar. Al menos, los que requiere el camino.



Ejemplo de camino inicial



Camino resuelto

Karel y Gretel Revolution

Archivo: **GretelRevolution.txt**

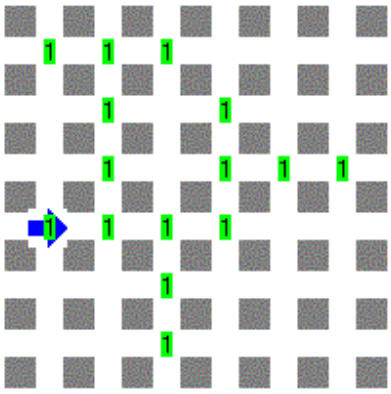
Historia

De nueva cuenta Gretel ha salido en busca de biperleñas. Ahora ha tomado varios caminos, pero ha tenido la misma precaución de siempre: “dejar beepers para que Karel la pueda encontrar”.

El camino que dejo Gretel esta hecho de manera que cada beeper está a sólo un paso del anterior y del siguiente (más no se sabe en que dirección), No hay caminos con ciclos. Es decir, no es posible salir de un beeper X, recorrer algún camino y regresar al beeper X sin repetir algún beeper del camino. Por último, no hay paredes sobre el camino.

Problema

Debes encontrar un algoritmo para que Karel recorra todo los caminos de beepers y los recoja.



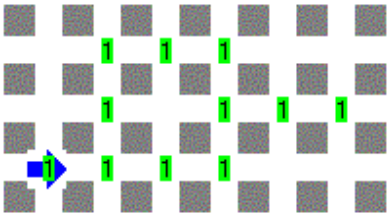
Camino válido

Consideraciones

1) Inicialmente Karel se encuentra sobre el primer beeper.

2) Karel termina cuando ha recogido todos los beepers sin importar su posición final y su orientación.

3) No importa en que orden recoge los beepers.



Camino no valido