

Pasos para resolver un problema de Karel

Tip: “No te aceleres, no hay prisa por empezar a escribir en la computadora tu código, primero toma tu tiempo, piensa en grande, sigue un proceso y al final tendrás un mejor resultado”

- 1) **Entender el problema:** Es importante realizar los siguientes pasos antes de pensar en una solución para el problema.
 - Leer cuidadosamente el problema que te dan.
 - Separar la “historia” del problema en sí.
 - Entender el objetivo final del problema (solución pedida).
 - Analizar las consideraciones y si tienes dudas hacerlas durante el tiempo previsto para esto durante los exámenes, o por email en caso de exámenes en línea.

- 2) **Pseudocódigo:** Estas son las herramientas con las que te puedes apoyar para hacer tu solución del problema.
 - Dibujos.
 - Esquemas.
 - Diagramas.
 - Escritos a manera de Notas.

- 3) **Codificar:** A grandes rasgos estos son los pasos para implementar tu solución.
 - Hacer el programa que de la solución al problema.
 - Capturar el programa en Karel, teniendo extremo cuidado de faltas ortográficas.
 - Hacer pruebas parciales al programa cada que terminas una sección de tu código.

- 4) **Prueba y Error:** Es muy importante que pruebes tu solución, para asegurarte que funcionará en el mayor número de casos posibles (y si se puede en todos, mejor!).
 - Prueba la solución con el caso (mundo) de ejemplo (recuerda que el caso de ejemplo NO es un caso que se evalúa).
 - Cuando tu solución resuelva el caso de ejemplo prueba con por lo menos otros 5 casos de prueba (mundos).

- 5) **Código perfecto:** Trata de pensar como evaluador e imaginar los casos de prueba. Los casos de prueba se dividen en:



- **Extremos:** Son los casos donde el mundo está lo más grande posible y basándose en las consideraciones, en pocas palabras son los casos más “difíciles” y los cuales generalmente sirven para saber quiénes serán los ganadores de los eventos
- **Tontos:** Son casos de prueba de solución trivial, por ejemplo mundos vacios donde incluso un código que solo diga APAGATE; funcionaria!!! Pero recuerda de estos casos te encontrarás a lo mucho 2 mundos por problema
- **Parecidos al ejemplo:** Son casos en los que el mundo inicial es similar al que viene en el caso de ejemplo con variantes mínimas.
- **Normales:** Son los casos (mundos) variados, que aunque no extremos si usan las consideraciones básicas y ofrecen una diversidad de opciones, por lo general el 50% de los casos de evaluación están en este punto.



Ejemplo:

Vamos a utilizar los pasos descritos para resolver problema, concretamente el problema “La Torre” No.16 del Problemando Básico 2.

1. Entender el problema.

Leemos la historia del problema que nos dice:

Karel consiguió chamba como demolidor de edificios en la ciudad y su primer trabajo es tumbar la torre más alta de la ciudad.

Separando la historia del problema, tenemos que en el mundo se encuentran unas columnas de zumbadores cuya altura desconocemos y el objetivo es buscar la más alta y poner todos sus zumbadores en la primera fila de la columna.

Las consideraciones del problema:

- *No se sabe donde inicia Karel.*
- *El mundo es rectangular o cuadrado y no tiene obstáculos adentro.*
- *Karel termina en la base de la torre encima del montón de escombros.*
- *Si hay dos torres del mismo tamaño Karel tiene que tumbar la de la izquierda.*

Las consideraciones las debemos de tener en cuenta cuando pensamos en nuestra solución, por ejemplo, si en mi solución quiero que karel recorra el mundo por filas, tengo que tener en cuenta que la primera consideración me dice que karel puede iniciar en cualquier parte, por lo tanto lo primero que hará mi programa es mandar a karel a la esquina por la cual comenzara a recorrer. Además por la cuarta consideración, concluyo en que el recorrido de karel en mi solución tiene que ser de izquierda a derecha, para que si hay dos columnas con la misma altura máxima siempre encuentre primero la de la izquierda.

3. Codificar.

Se tiene que implementar la solución para el problema, para este problema podemos dividirlo en 3 “sub-problemas”, el primero podría ser mandar a karel a la esquina superior-izquierda, el segundo sub-problema sería recorrer las filas de izquierda a derecha, de arriba hacia abajo, hasta encontrar un zumbador, el tercer sub-problema sería, una vez encontrado un zumbador, coger todos los que estén debajo de él y dejarlos todos en la primera fila.

Una vez que tienes una parte funcional del código puedes probarla para irte asegurando de que cada parte del problema funciona.



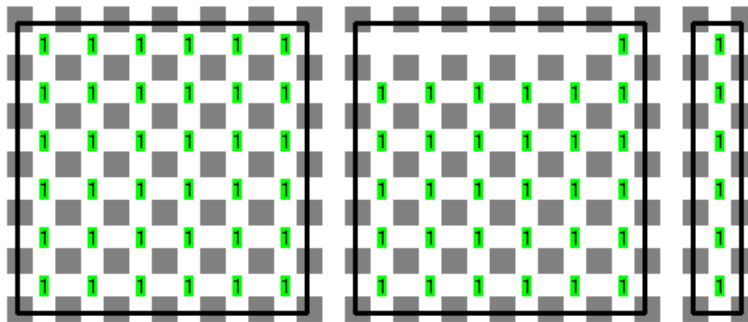
4. Prueba y Error.

Este pasó es muy importante, debes de probar primero tu código con el caso de ejemplo, si no funciona entonces debes de depurarlo, tratar de encontrar el error y corregirlo, una vez que tu programa soluciona el caso de ejemplo correctamente, debes de probar con más casos de prueba con estos podrás darte cuenta de mas errores que puedes tener en tu código.

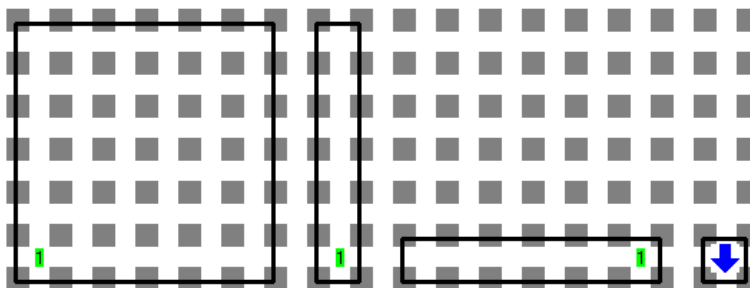
5. Código Perfecto.

Una vez que has depurado todos los errores de tu código, puedes pasar al paso final que es en donde te puedes dar cuenta si aspiras o no a un puntaje perfecto en el problema, tienes que probar tu código con casos de prueba de las 4 categorías apuntadas arriba, pero sobretodo con casos de prueba extremos y tontos.

Ejemplos de caso extremo:



Ejemplos de caso tonto:



Si tu programa logra resolver todos los casos de prueba anteriores correctamente, tienes un 99% de probabilidades de sacar puntaje perfecto.

