

UNIVERSIDAD DE LA CAÑADA

Guía para el primer concurso de programación

Nivel Básico



2011

M. C. Silvana Juárez Chalini

M. C. Beatriz Adriana Sabino Moxo

M. C. José Alberto Márquez Domínguez

CONTENIDO

INTRODUCCIÓN.....	4
FASES PARA LA RESOLUCIÓN DE PROBLEMAS.....	5
Análisis del Problema	5
Diseño del Algoritmo.....	7
KAREL.....	8
El mundo de Karel	8
Instrucciones Básicas de Karel.....	8
NIVEL 1: BÁSICO	9
Ejercicios Resueltos.....	9
Ejercicios Propuestos	28
BIBLIOGRAFÍA.....	35

INTRODUCCIÓN

Karel es un lenguaje de programación muy reducido que fue inventado por el Prof. Richard Patis del Departamento de Ciencias de la Computación, en la Universidad de Carnegie Mellon, USA. A diferencia de muchos lenguajes de programación, Karel no se enfoca al cálculo de complejas expresiones, asignación de valores a variables o creación de poderosas aplicaciones. Más bien es un lenguaje orientado a la resolución de tareas en un ambiente simulado por medio de la manipulación de un robot llamado Karel.

En el ambiente simulado vive un robot de nombre Karel (el cual le da el nombre al lenguaje de programación), que recibe su nombre del escritor checo Karel Capek, el primero en la historia en usar el término robot en uno de sus libros.

Este lenguaje ha resultado ser un excelente método para introducir a los jóvenes a la programación de computadoras. Al limitar el repertorio del lenguaje del estudiante, y por medio del empleo de refuerzos visuales de las consecuencias de los comandos más comúnmente utilizados, el concepto de Karel rápidamente introduce a los estudiantes a los conceptos de procedimiento y estructuras de control.

FASES PARA LA RESOLUCIÓN DE PROBLEMAS

Las fases o etapas constituyen el ciclo de vida del software, ayudarán en el proceso de resolución de un problema, estas consisten en:

1. Análisis del problema.
2. Diseño del algoritmo.
3. Codificación (Implementación).
4. Compilación y ejecución.
5. Verificación
6. Depuración.
7. Mantenimiento.
8. Documentación.

Las dos primeras etapas conducen a un diseño detallado escrito de forma de algoritmo¹. Durante la tercera etapa (Codificación) se implementa el algoritmo en un código escrito en un lenguaje de programación reflejando las ideas desarrolladas en las fases de análisis y diseño [Joyanes, 2003].

La Compilación, Ejecución y Verificación realiza la traducción y ejecución del programa, se comprueba rigurosamente y se eliminan todos los errores que pueda tener. Si existen errores es necesario modificarlo y actualízalo de manera que cumplan todas las necesidades de cambio de sus usuarios, para ello se usan las etapas de Verificación y Depuración.

Finalmente se debe usar la fase de Documentación, es decir, es la escritura de las diferentes fases del ciclo de vida del software, esencialmente el análisis, diseño y codificación, unidos a manuales de usuario y de referencia, así como normas para el mantenimiento.

En este concurso se pondrán a prueba las cuatro primeras fases, aunque se recomienda realizar las fases faltantes para terminar con el ciclo de vida del software.

Análisis del Problema

En esta fase se requiere una clara definición del problema, para poder hacer esto es conveniente realizar las siguientes preguntas:

1. ¿Qué entradas se requieren? (tipo y cantidad)
2. ¿Cuál es la salida deseada? (tipo y cantidad)
3. ¿Qué método produce la salida deseada?

Con dichas preguntas se determina qué necesita el programa para resolver el problema. La solución puede llevarse a cabo mediante varios algoritmos [Joyanes, 2004].

Un algoritmo dado correctamente resuelve un problema definido y determinado.

El algoritmo debe cumplir diferentes propiedades:

1. *Especificación precisa de la entrada.* Se debe dejar claro el número y tipo de valores de entrada y las condiciones iniciales que deben cumplir dichos valores.

¹ Definido como un conjunto de instrucciones utilizadas para resolver un problema específico.

2. *Especificación precisa de cada instrucción.* No debe haber ambigüedad sobre las acciones que se deben ejecutar en cada momento.
3. *Exactitud, corrección.* Si debe mostrar que el algoritmo resuelva el problema.
4. *Etapas bien definidas y concretas.* Concreto quiere decir que la acción descrita por esa etapa está totalmente comprendida por la persona o máquina que debe ejecutar el algoritmo. Cada etapa debe ser ejecutable en una cantidad finita de tiempo.
5. *Número finito de pasos.* Un algoritmo se debe componer de un número finito de pasos.
6. *Un algoritmo debe terminar.* En otras palabras, no debe entrar en un ciclo infinito.
7. *Descripción del resultado o efecto.* Debe estar claro cuál es la tarea que el algoritmo debe ejecutar. La mayoría de las veces, esta condición se expresa con la producción de un valor como resultado que tenga ciertas propiedades.

Ejemplo 1

¿Es un algoritmo la siguiente instrucción?

Problema: Escribir una lista de todos los enteros positivos

Solución: Es imposible ejecutar la instrucción anterior dado que hay infinitos enteros positivos.

Ejemplo 2

Problema: Calcular la paga neta de un trabajador conociendo el número de horas trabajadas, la tarifa horaria y la tasa de impuestos.

Solución: Debemos definir el problema.

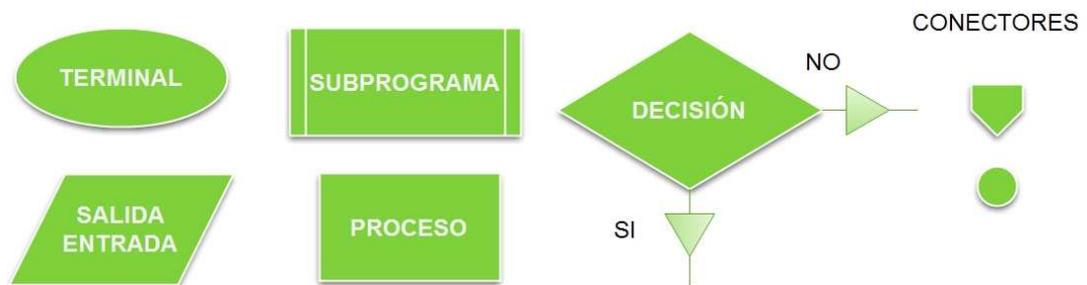
1. ¿Qué datos de entrada se requieren?
 Número de horas trabajadas
 Tarifa
 Impuestos
2. ¿Cuál es la salida deseada?
 Paga Neta
3. ¿Cuál es el método a usar? (Algoritmo)
 Inicio
 Leer Número de horas trabajadas
 Leer Tarifa
 Leer Impuestos
 $\text{Calcular Paga Bruta} = \text{Número de horas trabajadas} * \text{Tarifa}$
 $\text{Calcular Impuestos} = \text{Paga Bruta} * \text{Tasa}$
 $\text{Calcular Pago Neta} = \text{Paga Bruta} - \text{Impuestos}$
 Visualizar Paga Bruta
 Visualizar Impuestos
 Visualizar Pago Neta
 Fin

Diseño del Algoritmo

En esta fase, como se ha mencionado anteriormente, se determina cómo hace el programa la tarea solicitada. Los métodos más eficaces para el proceso de diseño se basan en el conocido divide y vencerás, esto es dividiendo el problema en subproblemas y a continuación dividir estos subproblemas en otros de nivel más bajo hasta que pueda ser implementada la solución.

Existen diferentes herramientas de programación, las más utilizadas para diseñar algoritmos son:

1. Diagramas de flujo: Es una representación gráfica de un algoritmo. Los símbolos normalizados por el Instituto Norteamericano de Normalización (ANSI) y los más frecuentes empleados se muestran a continuación.



2. Pseudocódigo: Es una herramienta de programación en la que las instrucciones se escriben en palabras similares en inglés o español, que facilitan tanto la escritura como la lectura de programas.

KAREL

El mundo de Karel

Karel es un robot que podemos controlar por medio de un programa para que realice cierto trabajo. El mundo de Karel consta de los siguientes elementos:

- *Calles* (horizontales) y *avenidas* (verticales) que se cruzan en esquinas.
- *Paredes* impenetrables colocadas entre dos esquinas.
- *Zumbadores* removibles colocados en las esquinas que emiten un sonido (su grosor es irrelevante).
- *Bolsa de zumbadores* que Karel lleva consigo.

Karel siempre está en una esquina y mirando al norte, sur, este u oeste. A través de tres cámaras puede ver si se encuentra una pared entre él y las esquinas más cercanas (enfrente, a su derecha y a su izquierda). Su oído le permite detectar el sonido de beepers en la esquina donde se encuentra.

La manera de comunicarse con Karel es por medio de un programa. El problema principal es que lo único que Karel puede hacer es seguir lo que le indiquemos "al pié de la letra". Karel no piensa y no puede darse cuenta de lo que queremos que haga si no sabemos cómo decírselo, para eso es necesario aprender su lenguaje.

Instrucciones Básicas de Karel

Realizar un trabajo o tarea específica con Karel consiste en llevarlo de una situación original a una final a través de la ejecución de instrucciones

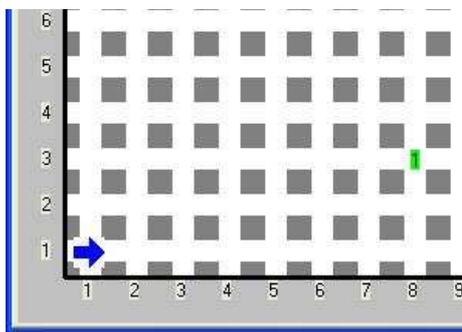
- *avanza*. Hace un paso en la dirección que está apuntando (puede causar error si hay una pared enfrente).
- *gira-izquierda*. Gira a su izquierda 90° (siempre se puede).
- *coge-zumbador*. Recoge un zumbador de la esquina donde está parado (puede causar error si no hay ningún zumbador en la esquina).
- *deja-zumbador*: Deposita un zumbador en la esquina (puede causar error si la bolsa de zumbadores de Karel está vacía).
- *Apágate*. Es el comando que finaliza y apaga a Karel.

Nota: Las nuevas funciones deben tener diferentes nombres y NO llamarse igual a una de las funciones básicas.

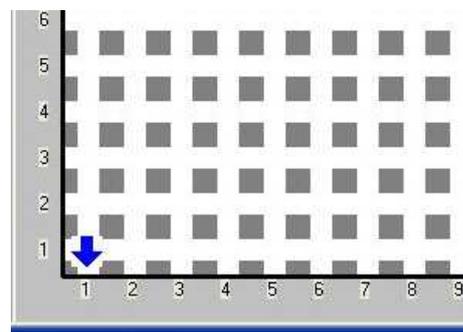
NIVEL 1: BÁSICO

Ejercicios Resueltos

Problema 1. Karel se encuentra en su casa (posición 1,1 viendo hacia el este), se ha dado cuenta que es hora de ir a recoger a su hermana a su escuela (posición 8,3), su misión es ir por ella y regresar juntos a casa.



Estado inicial, Karel tiene 0 zumbadores



Estado final, Karel tiene 1 zumbador y en la posición inicial

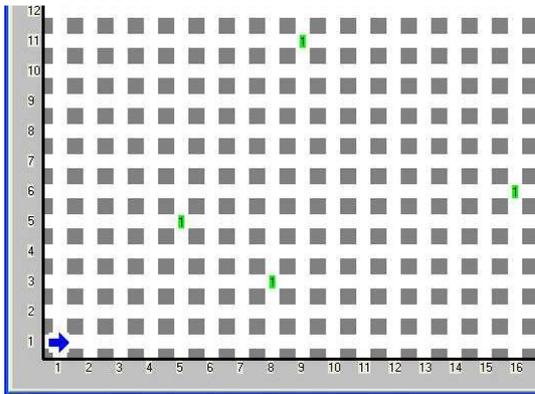
Programa

```
iniciar-programa
  inicia-ejecucion
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    gira-izquierda;
    avanza;
    avanza;
    coge-zumbador;
    gira-izquierda;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    gira-izquierda;
    avanza;
```

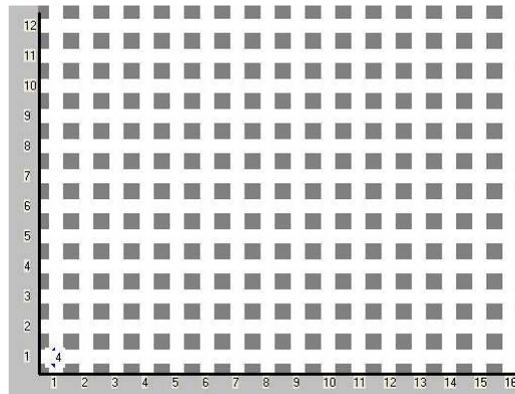
```
avanza;  
apagate;  
termina-ejecucion  
finalizar-programa
```

Problema 2. En un día de compras, Karel sale de su casa (posición 1,1) y se dirige a obtener los siguientes productos: leche, pan, huevo y carne, cada uno en diferentes establecimientos, al final regresa a casa con todas sus compras

- En el mundo de Karel los establecimientos donde venden esos productos están en las siguientes posiciones:
 - Leche: posición (5, 5).
 - Pan: posición (9, 11).
 - Huevo: posición (16, 6).
 - Carne: posición (8, 3).



Estado inicial, Karel tiene 0 zumbadores



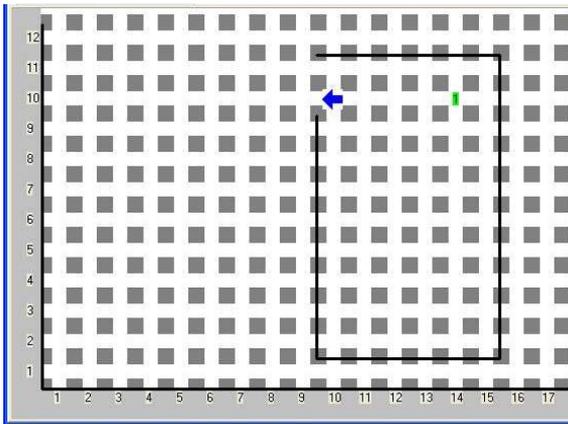
Estado final, Karel tiene 4 zumbadores y en la posición inicial

Programa

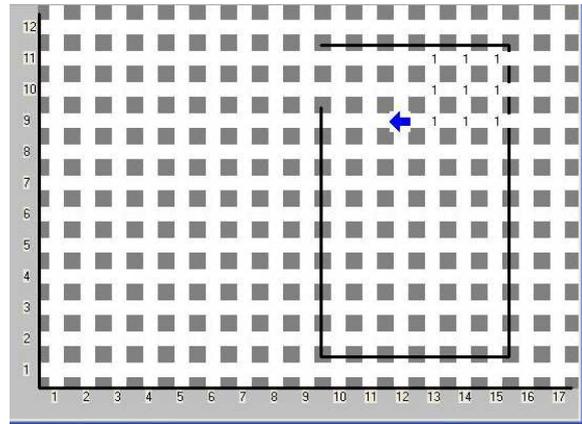
```
iniciar-programa  
inicia-ejecucion  
avanza;  
avanza;  
avanza;  
avanza;  
gira-izquierda;  
avanza;  
avanza;  
avanza;  
avanza;  
coge-zumbador;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
avanza;
```

avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
coge-zumbador;
gira-izquierda;
avanza;
gira-izquierda;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
coge-zumbador;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
avanza;
avanza;
avanza;
coge-zumbador;
avanza;
avanza;
avanza;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;

avanza;
avanza;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
coge-zumbador;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
avanza;
avanza;
coge-zumbador;
gira-izquierda;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
coge-zumbador;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
avanza;
avanza;
avanza;
gira-izquierda;
avanza;
avanza;
avanza;
avanza;



Estado inicial



Estado final

Programa

```

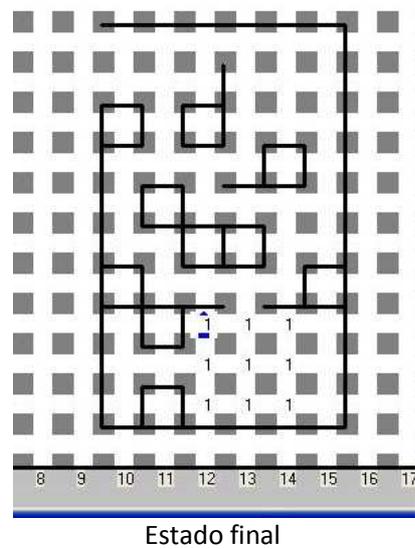
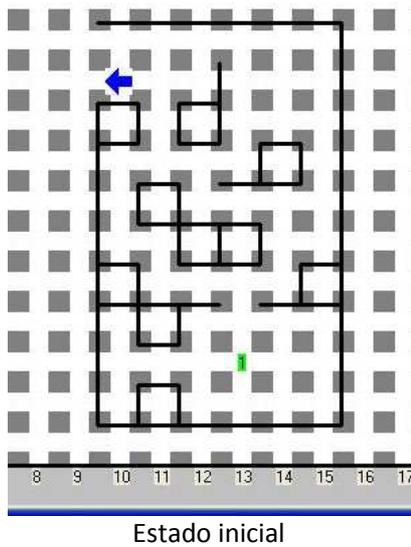
iniciar-programa
  inicia-ejecucion
    gira-izquierda;
    gira-izquierda;
    avanza;
    avanza;
    avanza;
    deja-zumbador;
    gira-izquierda;
    avanza;
    deja-zumbador;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    deja-zumbador;
    avanza;
    deja-zumbador;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    deja-zumbador;
    avanza;
    deja-zumbador;
    avanza;
  
```

```

    apagate;
    termina-ejecucion
finalizar-programa

```

Problema 5. Karel tiene una mascota que se le ha escapado en varias ocasiones, preocupado porque la última vez le costó mucho trabajo encontrarlo, decidió encerrarlo (utilizando 8 zumbadores) para que no se volviera a escapar, esta vez el perro estará en el jardín trasero de su casa y para llegar hasta allá tiene que pasar por varios obstáculos.



Programa

```

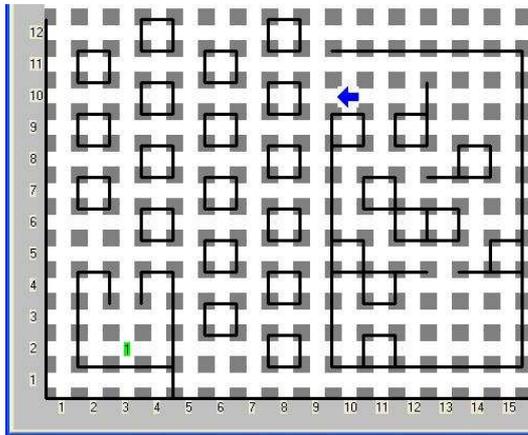
iniciar-programa
  inicia-ejecucion
    gira-izquierda;
    gira-izquierda;
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    gira-izquierda;
    avanza;
    avanza;
    gira-izquierda;
    avanza;
    gira-izquierda;
    gira-izquierda;

```

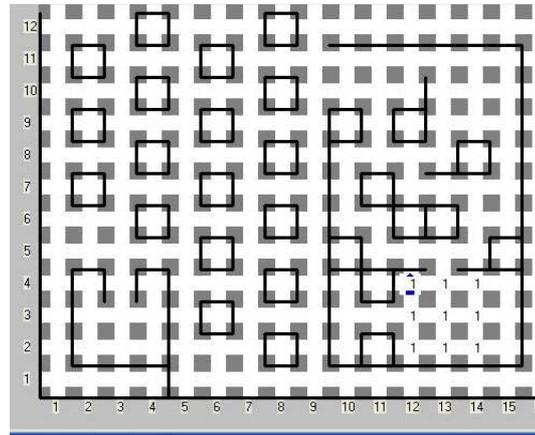
gira-izquierda;
avanza;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
gira-izquierda;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
avanza;
deja-zumbador;
apagate;
termina-ejecucion
finalizar-programa

Problema 6. Karel tiene una mascota que se le ha escapado en varias ocasiones, esta vez lo volvió a hacer, un vecino le avisó que su mascota estaba en el parque de la ciudad, la misión de Karel es ir por él , regresarlo a su casa y encerrarlo para que no se vuelva a escapar.

- La casa de Karel es rectangular delimitado por paredes.
- Karel inicia apuntando hacia el este en la esquina inferior izquierda de su casa.



Estado inicial



Estado final

Programa

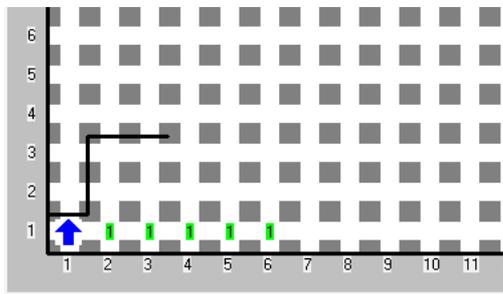
```

iniciar-programa
  inicia-ejecucion
    avanza;
    gira-izquierda;
    avanza;
    avanza;
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    avanza;
    gira-izquierda;
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    avanza;
    gira-izquierda;
    avanza;
    avanza;
    avanza;
    avanza;
    coge-zumbador;
    gira-izquierda;
  
```

gira-izquierda;
avanza;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
avanza;
avanza;
gira-izquierda;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
gira-izquierda;
avanza;
avanza;

gira-izquierda;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
avanza;
deja-zumbador;
apagate;
termina-ejecucion
finalizar-programa

Problema 7. Karel debe recoger los 5 zumbadores que se encuentran en la calle 1 del mundo.



Estado inicial, Karel tiene 0 zumbadores



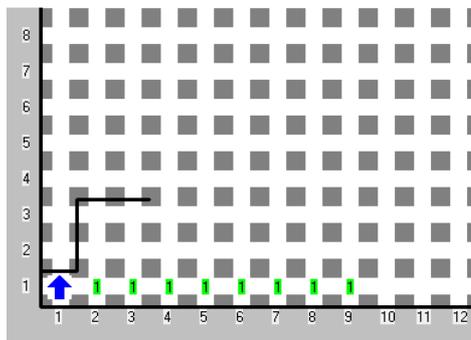
Estado final, Karel tiene 5 zumbadores

Programa

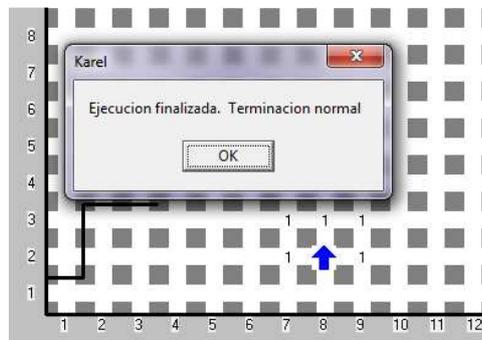
```

iniciar-programa
  inicia-ejecucion
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    coge-zumbador;
    avanza;
    coge-zumbador;
    avanza;
    coge-zumbador;
    avanza;
    coge-zumbador;
    avanza;
    coge-zumbador;
    apagate;
  termina-ejecucion
finalizar-programa
  
```

Problema 8. Karel se encuentra en alguna esquina de su mundo, con 8 o más zumbadores en su bolsa, se necesita que deje las chicharras en las esquinas de su alrededor.



Estado inicial, Karel tiene 0 zumbadores

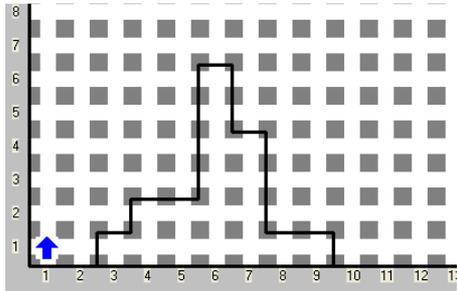


Estado final, Karel tiene 0 zumbadores, todas ellas puestas alrededor en la posición (8,1)

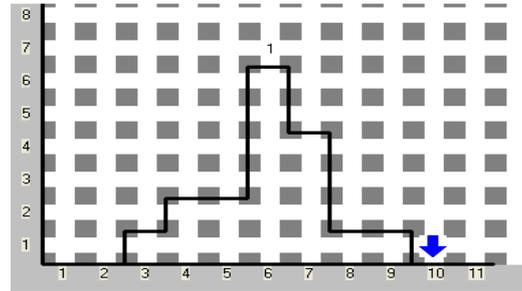
Programa

iniciar-programa
inicia-ejecucion
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
coge-zumbador;
avanza;
deja-zumbador;
avanza;
deja-zumbador;
gira-izquierda;
avanza;
deja-zumbador;
avanza;
deja-zumbador;
gira-izquierda;
avanza;
deja-zumbador;
avanza;
gira-izquierda;
avanza;
gira-izquierda;
avanza;
apagate;
termina-ejecucion
finalizar-programa

Problema 9. Karel se encuentra frente a una montaña, debe conquistar el punto más alto y dejar una bandera. La bandera estará representada por un zumbador.



Estado inicial, Karel debe dejar un zumbador



Estado final, Karel ha dejado su bandera y se encuentra en la posición (10,1)

Programa

```

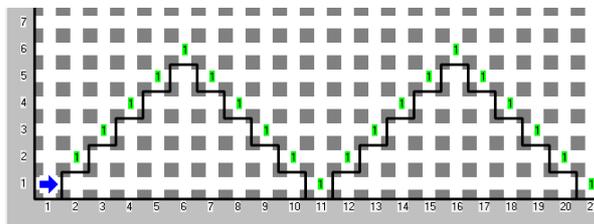
iniciar-programa
  inicia-ejecucion
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    gira-izquierda;
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    gira-izquierda;
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    avanza;
    avanza;
    avanza;
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    deja-zumbador;
    avanza;
    gira-izquierda;
  
```

```

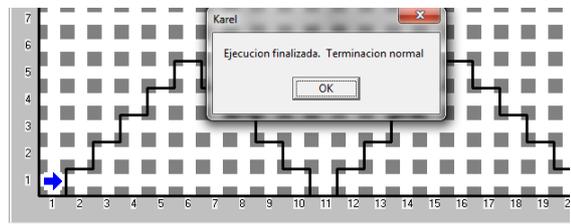
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
avanza;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
apagate;
termina-ejecucion
finalizar-programa

```

Problema 10. Karel se encuentra en el origen mirando hacia el este, en el mundo hay montañas escalonadas que parten del límite horizontal del mundo, y alcanzan una altura de 6 calles, las dos montañas están separadas por una calle, debe recoger los zumbadores que se encuentran en su mundo.



Estado inicial, Karel tiene 0 zumbadores



Estado final, Karel tiene 20 zumbadores y regreso a su posición inicial

Programa

```

iniciar-programa
inicia-ejecucion
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
coge-zumbador;
gira-izquierda;
avanza;

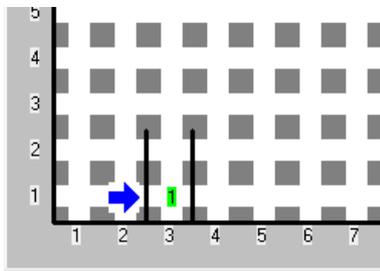
```

```

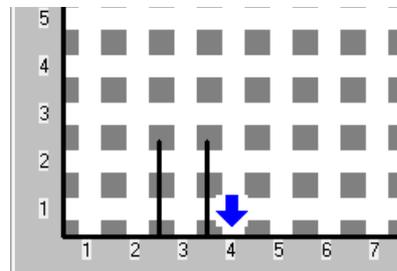
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
  coge-zumbador;
  gira-izquierda;
avanza;
  gira-izquierda;
  gira-izquierda;
  gira-izquierda;
avanza;
  coge-zumbador;
avanza;
  gira-izquierda;
  gira-izquierda;
  gira-izquierda;
avanza;
  coge-zumbador;
  apagate;
termina-ejecucion
finalizar-programa

```

Problema 11. Karel se encuentra en alguna calle de la avenida 2. Al frente de él hay un muro de dos calles de altura y una calle más adelante hay otro muro igual. A la izquierda del segundo muro, hay un zumbador que Karel debe recoger.



Estado inicial, Karel tiene 0 zumbadores



Estado final, Karel tiene 1 zumbador

Programa

```

iniciar-programa
inicia-ejecucion
  gira-izquierda;
  avanza;
  avanza;
  gira-izquierda;
  gira-izquierda;
  gira-izquierda;
  avanza;
  gira-izquierda;
  gira-izquierda;

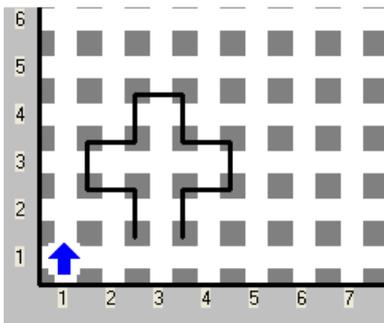
```

```

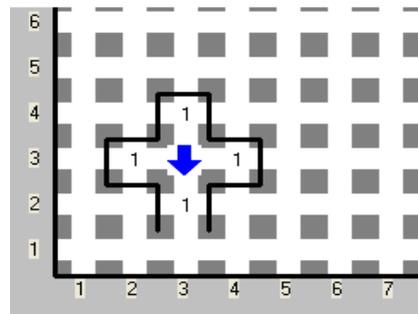
gira-izquierda;
avanza;
avanza;
coge-zumbador;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
apagate;
termina-ejecucion
finalizar-programa

```

Problema 12. La siguiente figura se conoce como escalerágonos, Karel debe poner zumbadores por toda la “orilla”, un zumbador en cada lugar.



Estado inicial, Karel tiene 4 zumbadores.



Estado Final, Karel tiene 0 zumbadores y está en la posición (3,3)

Programa

```

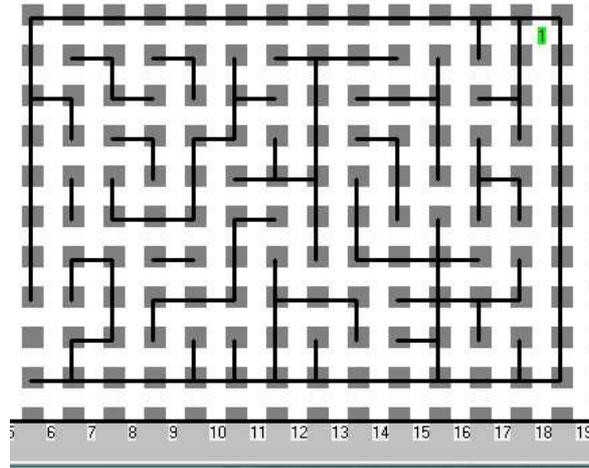
iniciar-programa
inicia-ejecucion
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
avanza;

```

deja-zumbador;
avanza;
gira-izquierda;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
deja-zumbador;
gira-izquierda;
gira-izquierda;
avanza;
apagate;
termina-ejecucion
finalizar-programa

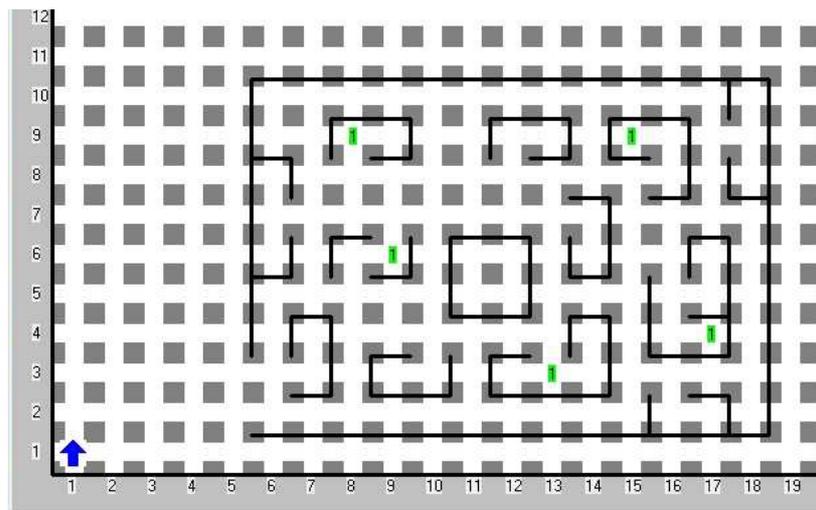
Ejercicios Propuestos

Problema 1. Una vez más la mascota de Karel escapó, pero en esta ocasión se metió a un laberinto, la misión de Karel es sacarlo y regresarlo a su casa (posición 1, 1).



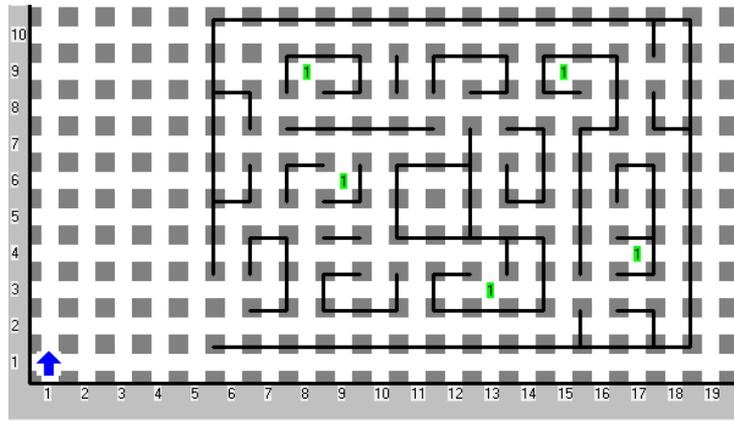
Problema 2. En un día de compras, Karel sale de su casa (posición 1,1) y se dirige a obtener los siguientes productos: leche, pan, huevo, carne y verduras, cada uno en diferentes establecimientos, al final regresa a casa con todas sus compras. Para agilizarlas Karel debe recorrer los establecimientos utilizando el camino más corto entre ellos.

Mundo Inicial



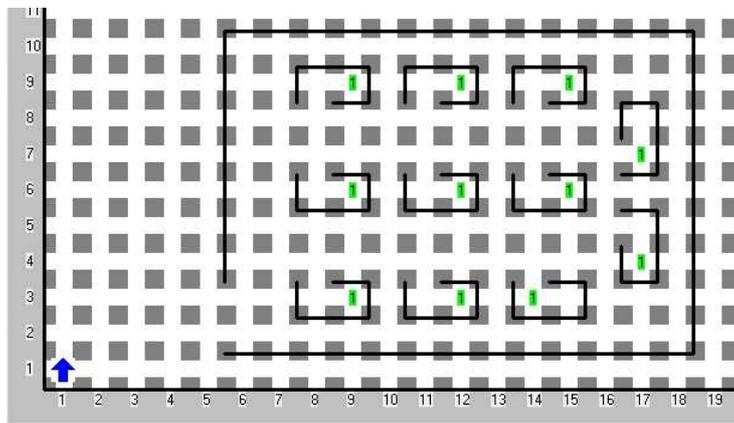
Problema 3. En un día de compras, Karel tiene que ir por los productos de siempre, sin embargo se da cuenta que algunas calles están cerradas por lo que necesita tomar vías alternas para llegar a los diferentes establecimientos y realizar sus compras.

Mundo Inicial



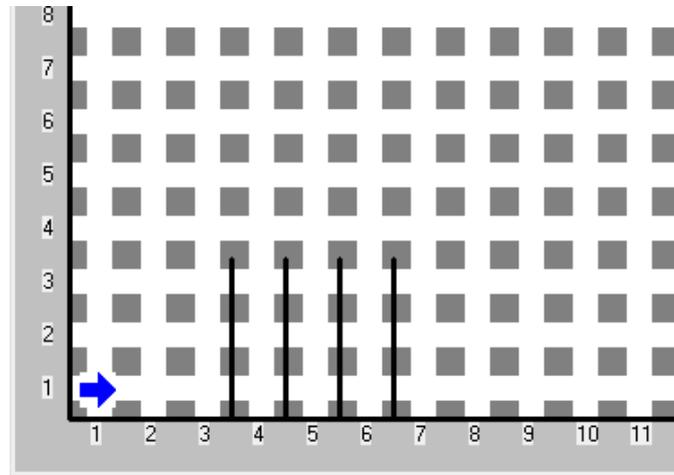
Problema 4. Karel consiguió un nuevo empleo en la tienda que renta películas, en este caso estas se entregan a domicilio y cada vez que Karel va por ellas

Mundo Inicial



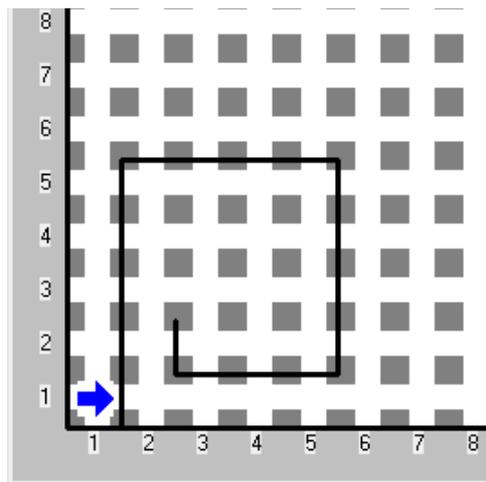
Problema 5. Se define el mundo de Karel, quien debe formar tres torres del mismo tamaño en las avenidas rodeadas por muros.

Mundo Inicial



Problema 6. Karel se encuentra en la siguiente situación, y debe llenar el cuarto de zumbadores.

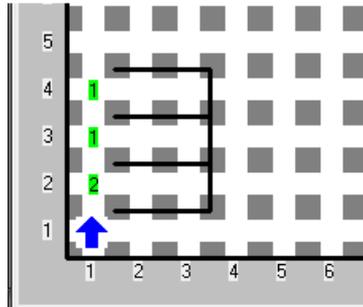
Mundo Inicio



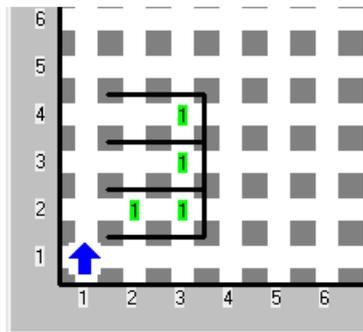
Problema 7. Karel se encuentra en la siguiente situación, y debe llenar el cuarto de zumbadores como se ilustra.

Problema 10. Karel trabaja en una librería y debe colocar todos los libros en los cuatro anaqueles de la biblioteca, en cada anaquel caben hasta dos libros; Karel debe guardarlos.

Mundo inicial

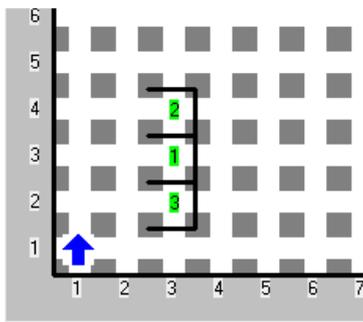


Mundo final

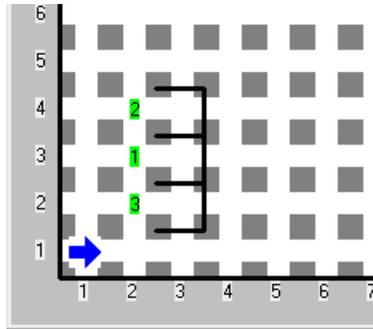


Problema 10. Karel tiene un criadero de conejos formado por cuatro corrales. Todos los días saca los conejos de su corral y los pone frente a la entrada para que coman. En cada corral puede haber 3 conejos. Karel debe sacar los conejos de cada corral y ponerlos frente a la entrada, Karel parte mirando al norte y debe finalizar mirando al este.

Mundo inicial

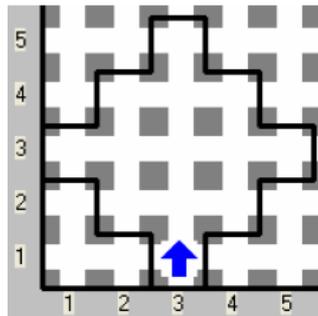


Mundo final

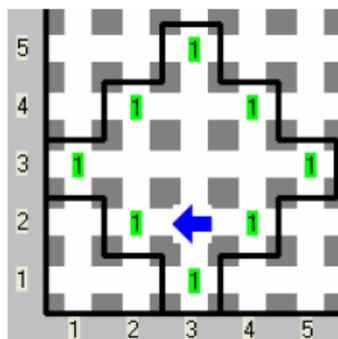


Problema 11. Karel debe poner zumbadores por toda la “orilla” del siguiente escalerágono.

Mundo inicial

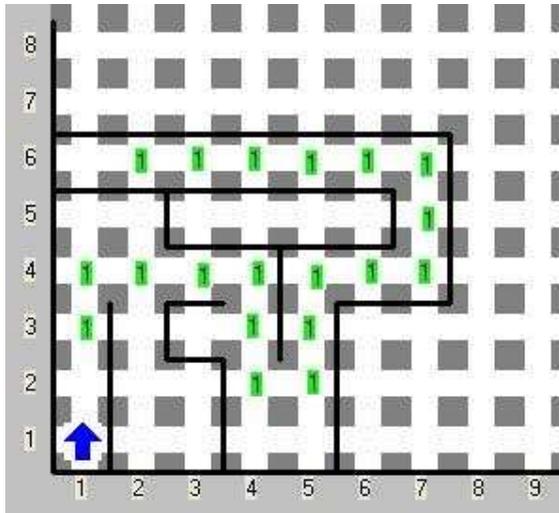


Mundo final

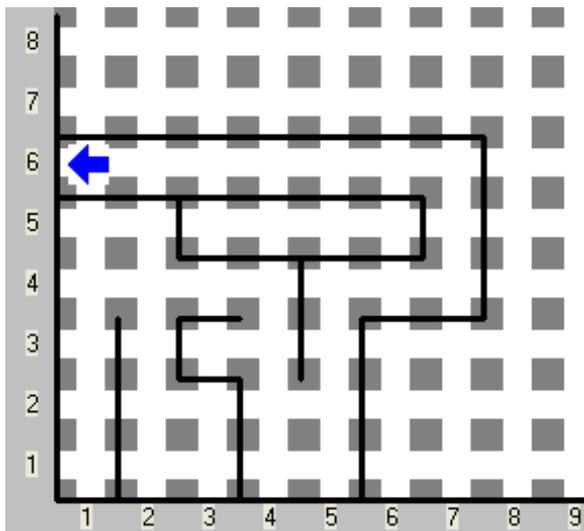


Problema 12. Karel y su hermanita Gretel fueron a la feria, y a los dos hermanos se les ocurrió entrar al juego del laberinto. Primero entró Gretel, a quien se le ocurrió dejar un zumbador a cada paso que daba al internarse en el laberinto para que Karel pudiera encontrarla. Al entrar Karel y encontrar los zumbadores, decidió buscarla siguiendo el camino de zumbadores que Gretel había dejado. Karel debe terminar en el otro extremo del laberinto y haber recogido todos los zumbadores que lo conforman.

Mundo inicial



Mundo final



BIBLIOGRAFÍA

Luis Joyanes Aguilar. “*Fundamentos de Programación, Algoritmos, estructuras de datos y objetos*”. Tercera Edición. McGrawHill. 2003.

Luis Joyanes Aguilar, Ignacio Zahonero Martínez. “*Algoritmos y Estructura de datos, una perspectiva en C*”. Primera Edición. McGrawHill. 2004.

Glenn Brookshear. “*Introducción a las Ciencias de la Computación*”. Cuarta Edición. Addison Wesley Iberoamericana. 1995.

José J. García-Badell. “*Turbo C++3 Programación y manejo de Archivos*”. Primera Edición. Addison Wesley Iberoamericana. 1994.

Jordi Batallar Mascarrell, Rafael Magdalena Benedito. “*Programación en C*”. Primera Edición. Alfa Omega. 2001.