

# UNIVERSIDAD DE LA CAÑADA

## Primer concurso de programación

Guía



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
NIVEL 2: INTERMEDIO .....	35
Ejercicios Resueltos.....	35
Ejercicios Propuestos .....	53
BIBLIOGRAFÍA.....	60

## 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 algoritmo<sup>1</sup>. 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.

---

<sup>1</sup> 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.

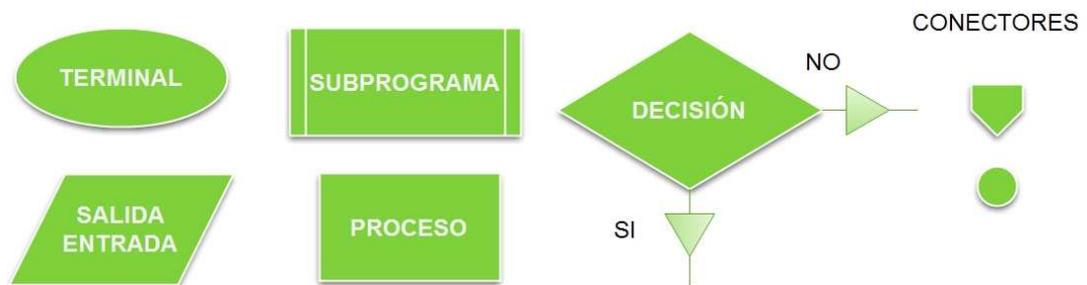
1. ¿Qué datos de entrada se requieren?  
 Número de horas trabajadas  
 Tarifa  
 Impuestos
2. ¿Cuál es la salida deseada?  
 Paga Neta
3. ¿Cuál es el método a usar? (Algoritmo)  
 Inicio  
 Leer Número de horas trabajadas  
 Leer Tarifa  
 Leer Impuestos  
 $\text{Calcular Paga Bruta} = \text{Número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

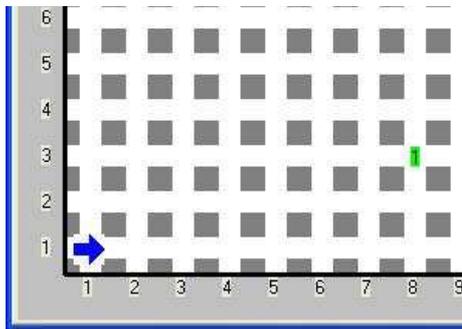
- *avanza*. Hace un paso en la dirección que está apuntando (puede causar error si hay una pared enfrente).
- *gira-izquierda*. Gira a su izquierda 90° (siempre se puede).
- *coge-zumbador*. Recoge un zumbador de la esquina donde está parado (puede causar error si no hay ningún zumbador en la esquina).
- *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.

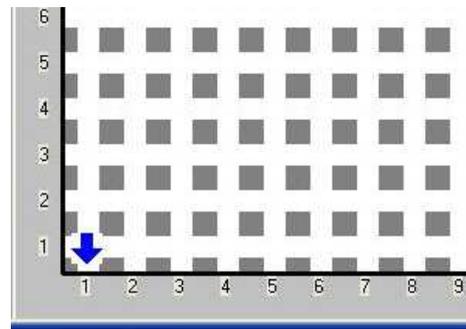
# NIVEL 1: BÁSICO

## 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;
    gira-izquierda;
    avanza;
```

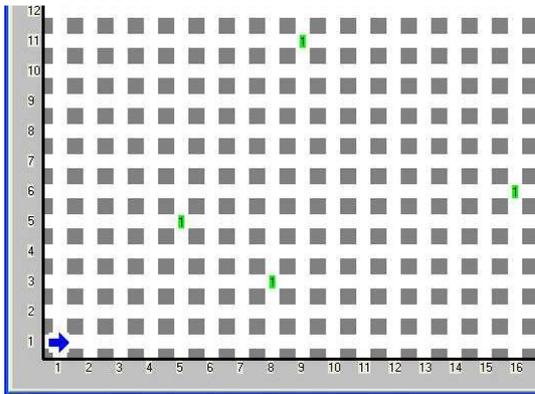
```

avanza;
apagate;
termina-ejecucion
finalizar-programa

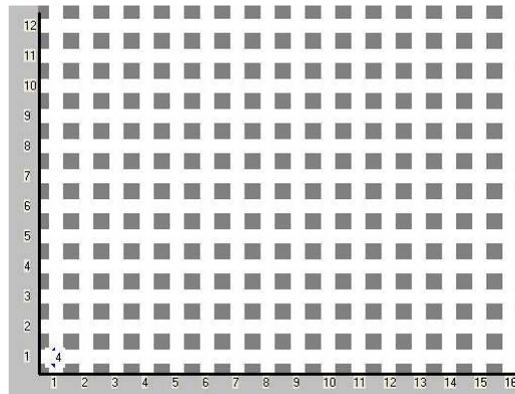
```

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

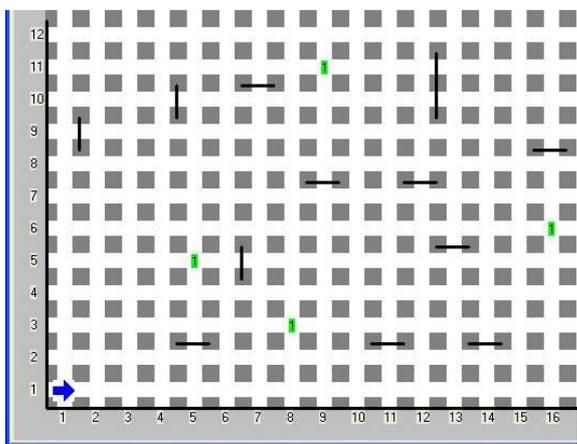
```

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

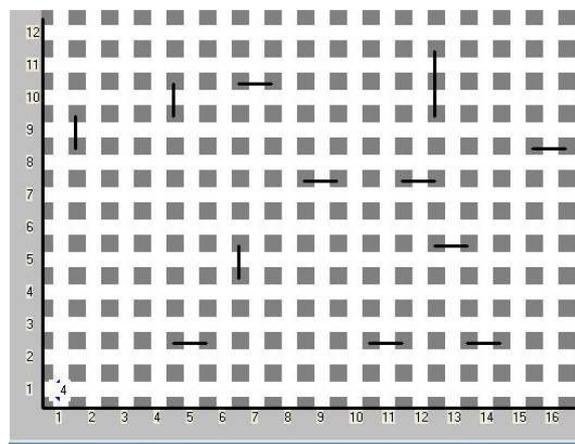
```

**Problema 3.** Es el día de compras, Karel tiene que ir por los productos de siempre, sin embargo algunas calles están cerradas, por lo que tiene que tomar vías alternas.

- Los establecimientos están en las mismas posiciones.



Estado inicial, Karel tiene 0 zumbadores



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

### Programa

```

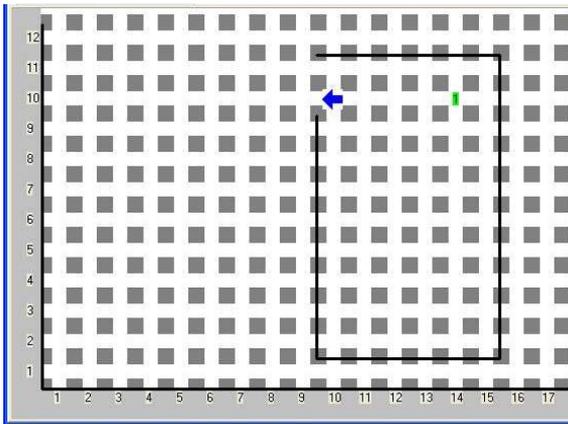
iniciar-programa
inicia-ejecucion
avanza;
avanza;
avanza;
gira-izquierda;

```

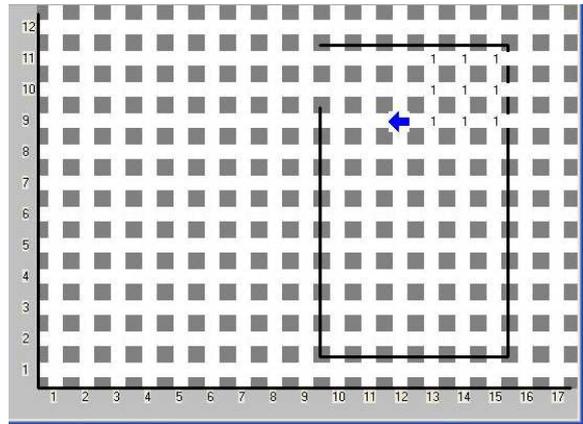
avanza;  
avanza;  
avanza;  
avanza;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
coge-zumbador;  
avanza;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
avanza;  
gira-izquierda;  
avanza;  
avanza;  
coge-zumbador;  
gira-izquierda;  
avanza;  
avanza;  
avanza;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

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

**Problema 4.** 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.



Estado inicial



Estado final

### Programa

```

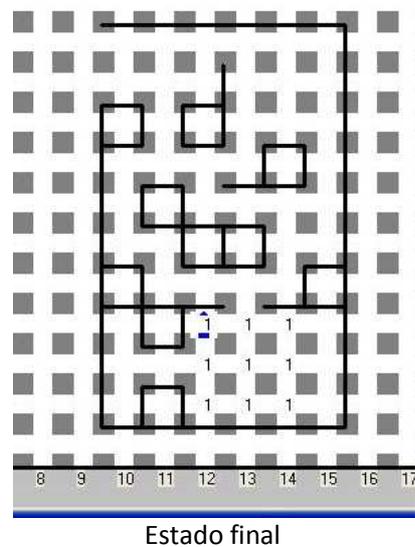
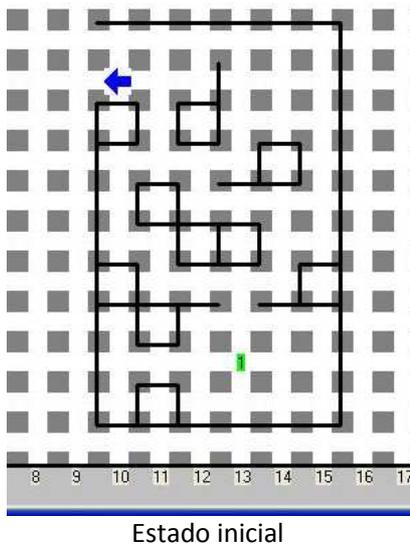
iniciar-programa
  inicia-ejecucion
    gira-izquierda;
    gira-izquierda;
    avanza;
    avanza;
    avanza;
    deja-zumbador;
    gira-izquierda;
    avanza;
    deja-zumbador;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    deja-zumbador;
    avanza;
    deja-zumbador;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

```

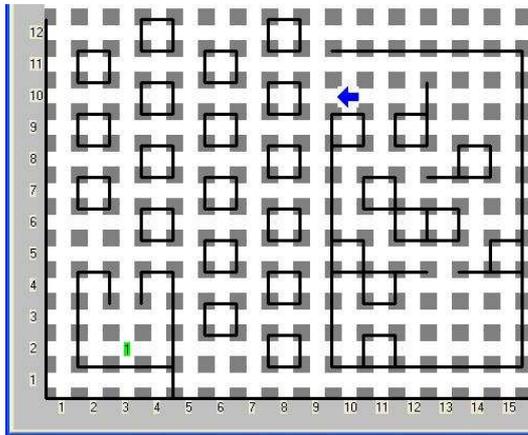
iniciar-programa
inicia-ejecucion
  gira-izquierda;
  gira-izquierda;
  avanza;
  gira-izquierda;
  gira-izquierda;
  gira-izquierda;
  avanza;
  avanza;
  gira-izquierda;
  gira-izquierda;
  gira-izquierda;
  avanza;
  gira-izquierda;
  avanza;
  avanza;
  gira-izquierda;
  avanza;
  gira-izquierda;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

```

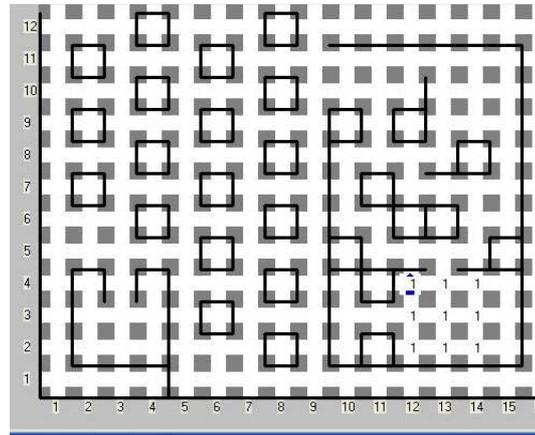
gira-izquierda;  
avanza;  
gira-izquierda;  
avanza;  
avanza;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
deja-zumbador;  
gira-izquierda;  
avanza;  
deja-zumbador;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
deja-zumbador;  
avanza;  
deja-zumbador;  
gira-izquierda;  
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;
  
```

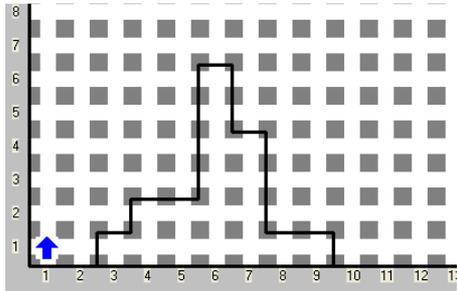
gira-izquierda;  
avanza;  
avanza;  
avanza;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
avanza;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
avanza;  
avanza;  
avanza;  
gira-izquierda;  
avanza;  
avanza;  
avanza;  
avanza;  
avanza;  
avanza;  
avanza;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

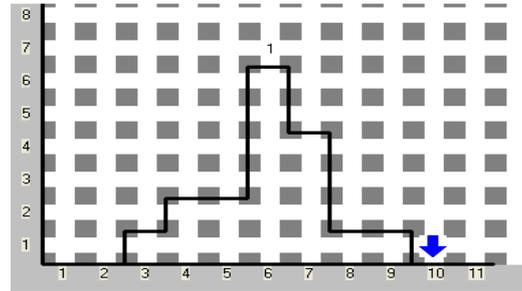




**Problema 9.** Karel se encuentra frente a una montaña, debe conquistar el punto más alto y dejar una bandera. La bandera estará representada por un zumbador.



Estado inicial, Karel debe dejar un zumbador



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

### Programa

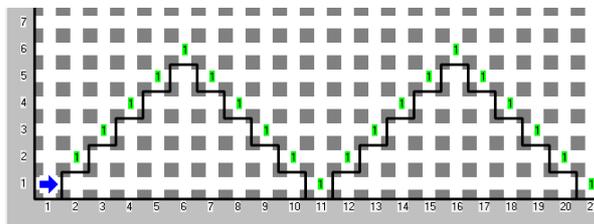
```
iniciar-programa
  inicia-ejecucion
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    gira-izquierda;
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    gira-izquierda;
    avanza;
    gira-izquierda;
    avanza;
    avanza;
    avanza;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

```

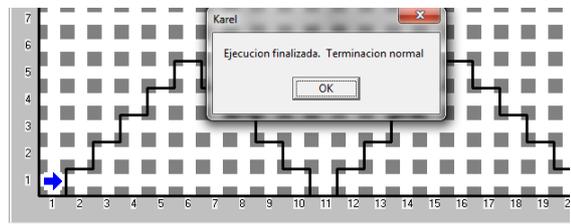
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
avanza;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
apagate;
termina-ejecucion
finalizar-programa

```

**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;

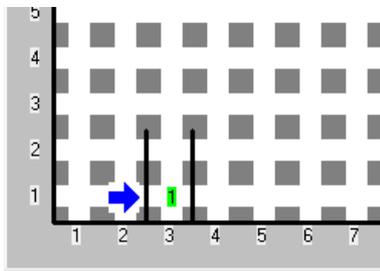
```

```

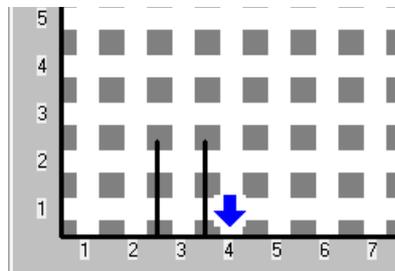
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
  coge-zumbador;
  gira-izquierda;
avanza;
  gira-izquierda;
  gira-izquierda;
  gira-izquierda;
avanza;
  coge-zumbador;
avanza;
  gira-izquierda;
  gira-izquierda;
  gira-izquierda;
avanza;
  coge-zumbador;
  apagate;
  termina-ejecucion
finalizar-programa

```

**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;

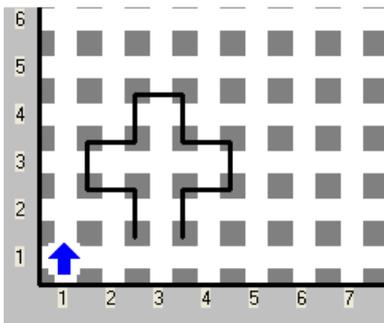
```

```

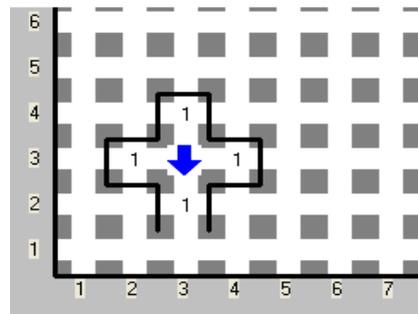
gira-izquierda;
avanza;
avanza;
coge-zumbador;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
apagate;
termina-ejecucion
finalizar-programa

```

**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

```

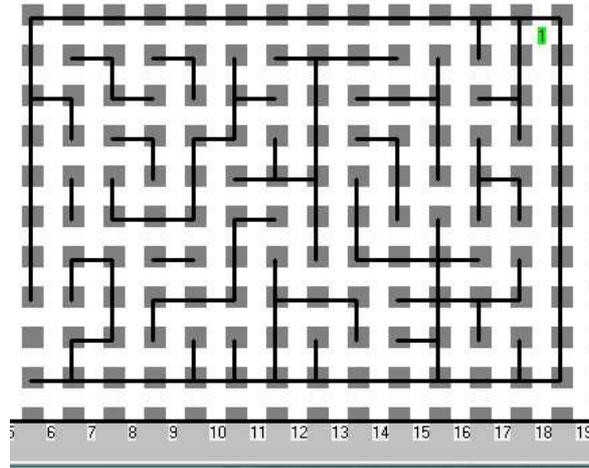
iniciar-programa
inicia-ejecucion
gira-izquierda;
gira-izquierda;
gira-izquierda;
avanza;
avanza;
gira-izquierda;
avanza;

```

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

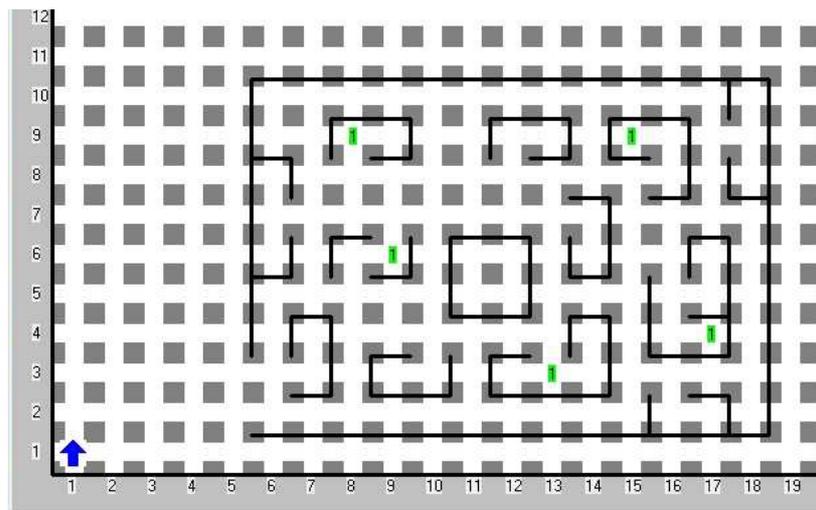
## Ejercicios Propuestos

**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).



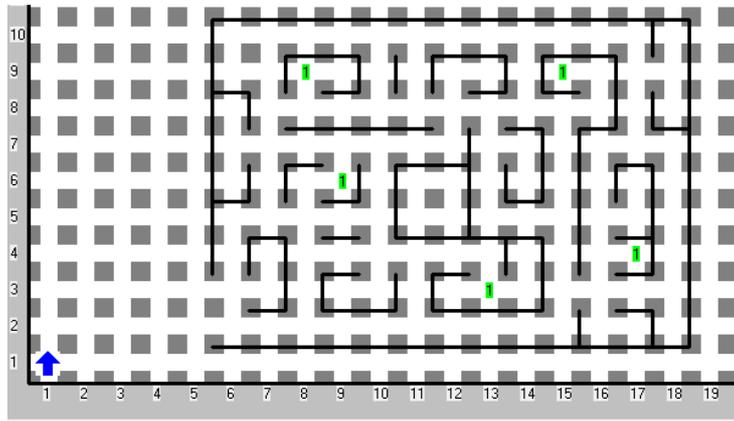
**Problema 2.** En un día de compras, Karel sale de su casa (posición 1,1) y se dirige a obtener los siguientes productos: leche, pan, huevo, carne y verduras, cada uno en diferentes establecimientos, al final regresa a casa con todas sus compras. Para agilizarlas Karel debe recorrer los establecimientos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

*Mundo Inicial*



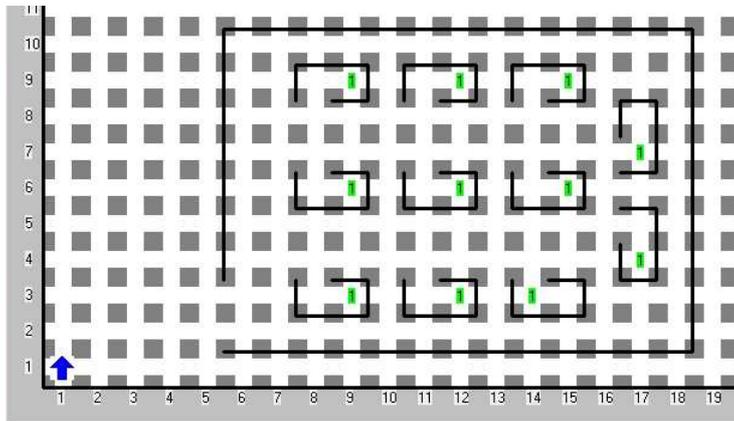
**Problema 3.** En un día de compras, Karel tiene que ir por los productos de siempre, sin embargo se da cuenta que algunas calles están cerradas por lo que necesita tomar vías alternas para llegar a los diferentes establecimientos y realizar sus compras.

*Mundo Inicial*



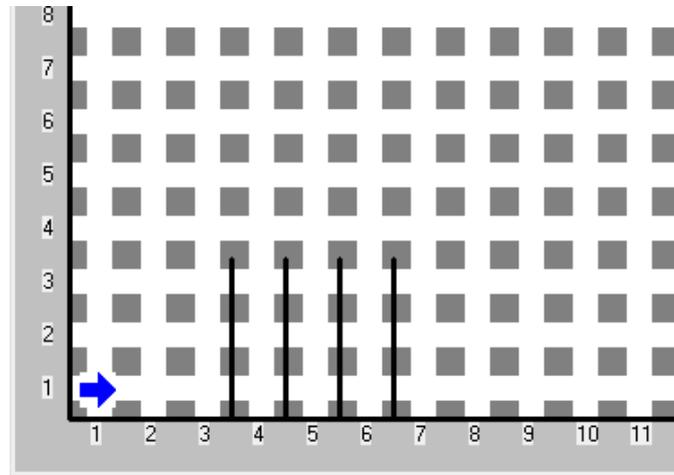
**Problema 4.** Karel consiguió un nuevo empleo en la tienda que renta películas, en este caso estas se entregan a domicilio y cada vez que Karel va por ellas

*Mundo Inicial*



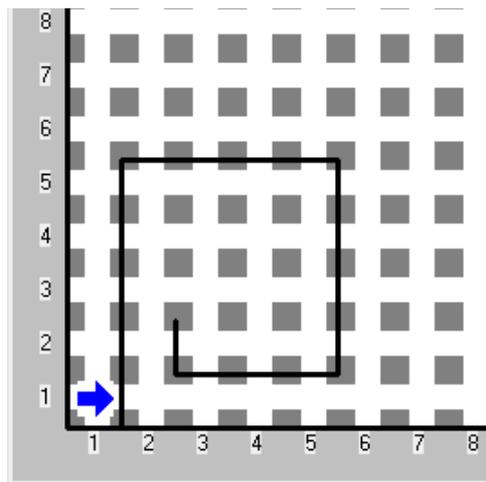
**Problema 5.** Se define el mundo de Karel, quien debe formar tres torres del mismo tamaño en las avenidas rodeadas por muros.

Mundo Inicial



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

Mundo Inicio

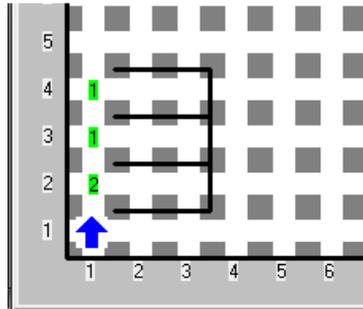


Problema 7. Karel se encuentra en la siguiente situación, y debe llenar el cuarto de zumbadores como se ilustra.

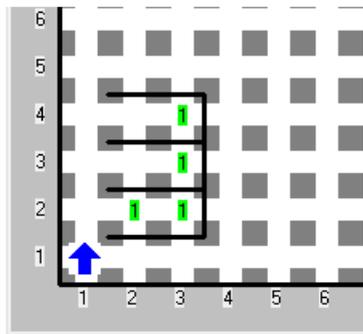


**Problema 10.** Karel trabaja en una librería y debe colocar todos los libros en los cuatro anaqueles de la biblioteca, en cada anaquel caben hasta dos libros; Karel debe guardarlos.

*Mundo inicial*

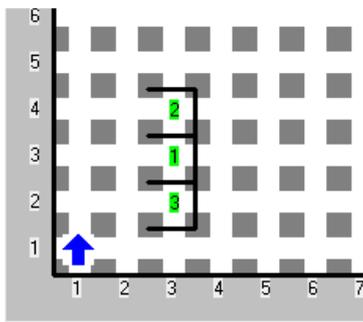


*Mundo final*

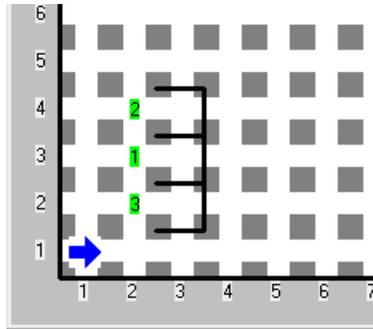


**Problema 10.** Karel tiene un criadero de conejos formado por cuatro corrales. Todos los días saca los conejos de su corral y los pone frente a la entrada para que coman. En cada corral puede haber 3 conejos. Karel debe sacar los conejos de cada corral y ponerlos frente a la entrada, Karel parte mirando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

*Mundo inicial*

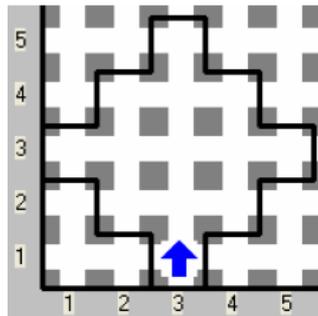


Mundo final

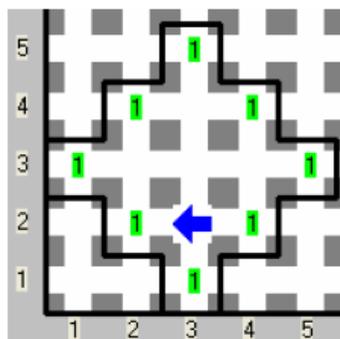


**Problema 11.** Karel debe poner zumbadores por toda la “orilla” del siguiente escalerágono.

Mundo inicial

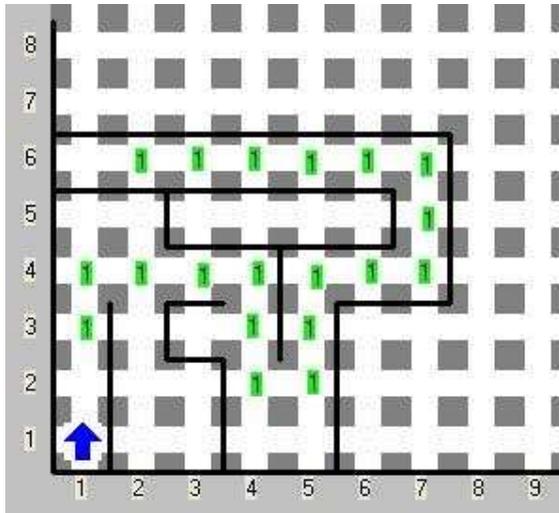


Mundo final

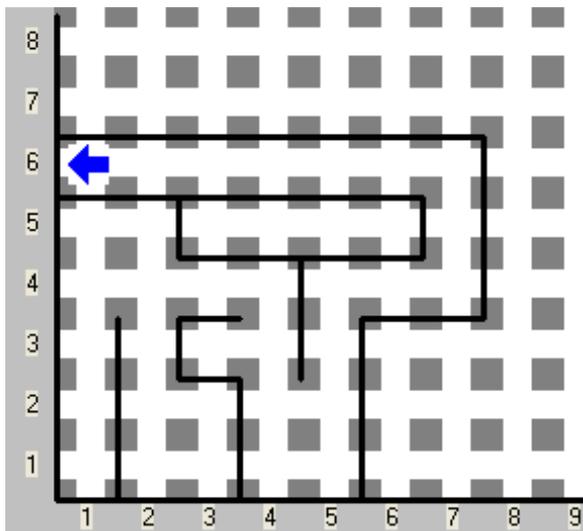


**Problema 12.** Karel y su hermanita Gretel fueron a la feria, y a los dos hermanos se les ocurrió entrar al juego del laberinto. Primero entró Gretel, a quien se le ocurrió dejar un zumbador a cada paso que daba al internarse en el laberinto para que Karel pudiera encontrarla. Al entrar Karel y encontrar 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*



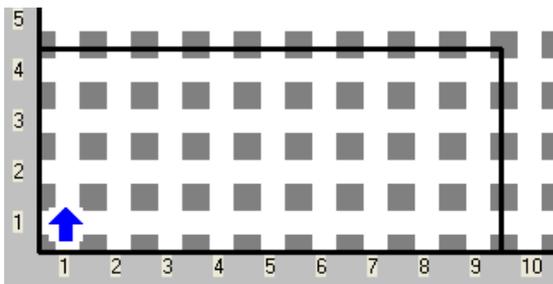


```

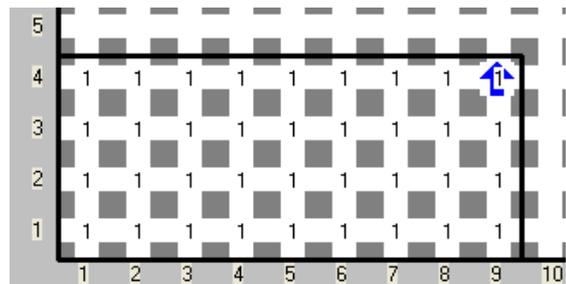
gira-izquierda;
avanza;
si frente-libre entonces inicio
  avanza;
fin
sino inicio
  gira-izquierda;
  gira-izquierda;
  gira-izquierda;
  fin;
  avanza;
  coge-zumbador;
  apagate;
  termina-ejecucion
finalizar-programa

```

**Problema 2.** Karel debe llenar una determinada área con zumbadores, escriba un programa que ayuda a Karel a colocar todos los zumbadores.



Mundo inicial, Karel se encuentra en la posición (1,1)



Mundo final, Karel se encuentra en la posición (9,4) y ha colocado todos los zumbadores

**Programa:**

```

iniciar-programa
  inicia-ejecucion
    deja-zumbador;
    repetir 4 veces inicio
      mientras frente-libre hacer inicio
        avanza;

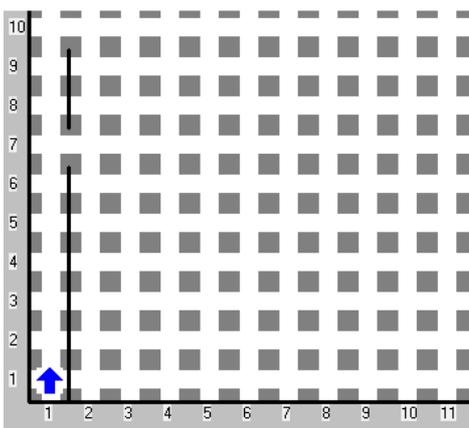
```

```

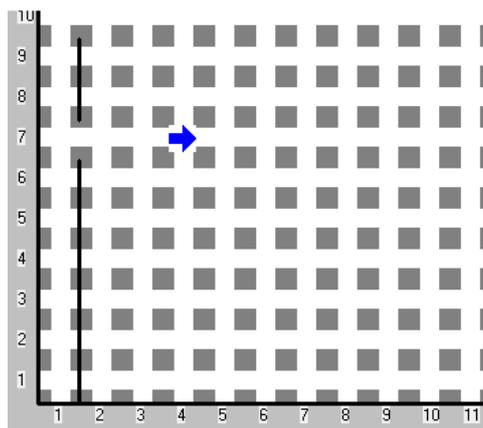
    deja-zumbador;
  fin;
  repetir 3 veces inicio
    gira-izquierda;
  fin;
  avanza;
  deja-zumbador;
  repetir 3 veces inicio
    gira-izquierda;
  fin;
  mientras frente-libre hacer inicio
    avanza;
    deja-zumbador;
  fin;
  gira-izquierda;
  avanza;
  deja-zumbador;
  gira-izquierda;
  fin;
  mientras frente-libre hacer inicio
    avanza;
    deja-zumbador;
  fin;
  apagate;
  termina-ejecucion
  finalizar-programa

```

**Problema 3.** Karel está en un pasillo en el cual las puertas están a la derecha, Karel avanzará hasta que encuentre una puerta y colocarse enfrente de esta, viéndola, es decir, para entonces avanzar y salir del pasillo.



Mundo inicial, posición inicial.

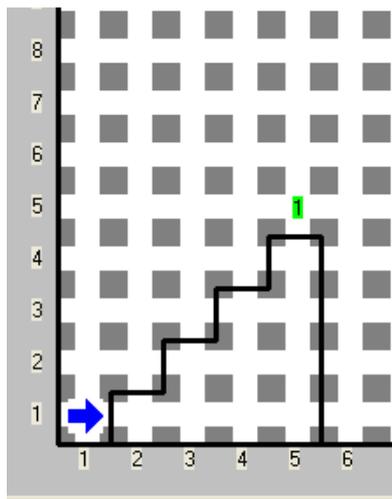


Mundo final.

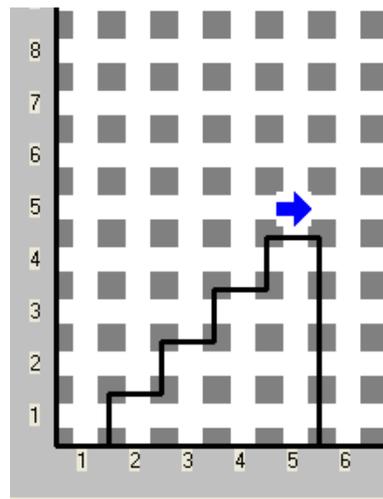
### Programa:

```
iniciar-programa
  inicia-ejecucion
    repetir 3 veces inicio
      gira-izquierda;
    fin;
  mientras frente-bloqueado hacer
    inicio
      gira-izquierda;
      avanza;
      repetir 3 veces
        inicio
          gira-izquierda;
        fin;
    fin;
  repetir 3 veces
    inicio
      avanza;
    fin;
  apagate;
  termina-ejecucion
finalizar-programa
```

**Problema 4.** Karel está enfrente de una escalera formada por paredes, deberá posicionarse en el último escalón, es decir, donde no encuentre más escalones que pisar, ahí deberá quedarse. Utilice un ciclo *repetir n veces* para llegar a la cima.



Estado inicial, Karel tiene 0 zumbadores



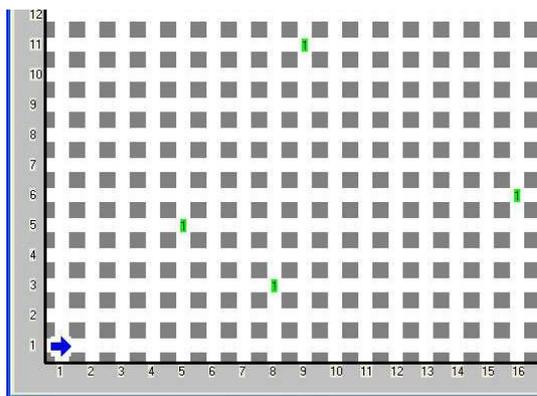
Estado final, Karel tiene 1 zumbador.

## Programa

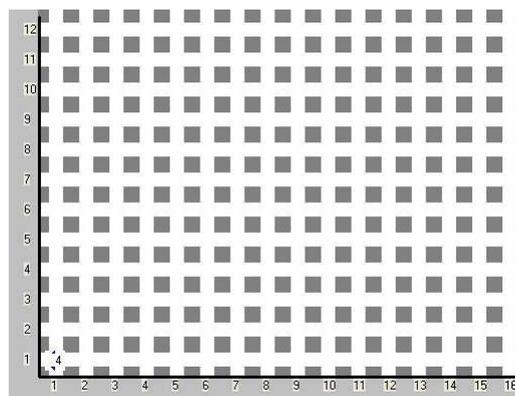
```
iniciar-programa
  inicia-ejecucion
    repetir 4 veces inicio
      gira-izquierda;
      avanza;
      gira-izquierda;
      gira-izquierda;
      gira-izquierda;
      avanza;
    fin;
  coge-zumbador;
  apagate;
  termina-ejecucion
finalizar-programa
```

**Problema 5.** 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. Resuelva este problema utilizando las estructuras de repetición cada vez que se encuentre con 3 o más acciones iguales.

- 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
    repetir 4 veces inicio
      avanza;
```

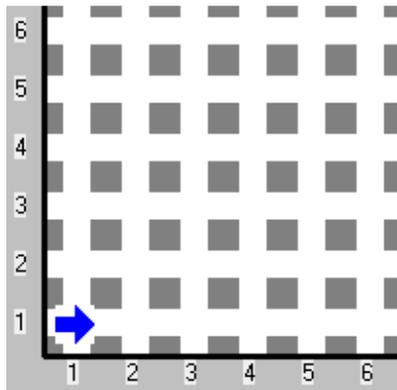
fin;  
repetir 3 veces inicio  
  gira-izquierda;  
fin;  
repetir 4 veces inicio  
  avanza;  
fin;  
coge-zumbador;  
repetir 4 veces inicio  
  avanza;  
fin;  
gira-izquierda;  
repetir 6 veces inicio  
  avanza;  
fin;  
coge-zumbador;  
repetir 3 veces inicio  
  gira-izquierda;  
fin;  
repetir 7 veces inicio  
  avanza;  
fin;  
repetir 3 veces inicio  
  gira-izquierda;  
fin;  
repetir 5 veces inicio  
  avanza;  
fin;  
coge-zumbador;  
repetir 3 veces inicio  
  avanza;  
fin;  
repetir 3 veces inicio  
  gira-izquierda;  
fin;  
repetir 8 veces inicio  
  avanza;  
fin;  
coge-zumbador;  
repetir 7 veces inicio  
  avanza;  
fin;  
gira-izquierda;  
avanza;  
avanza;  
repetir 4 veces inicio  
  deja-zumbador;  
fin;

```

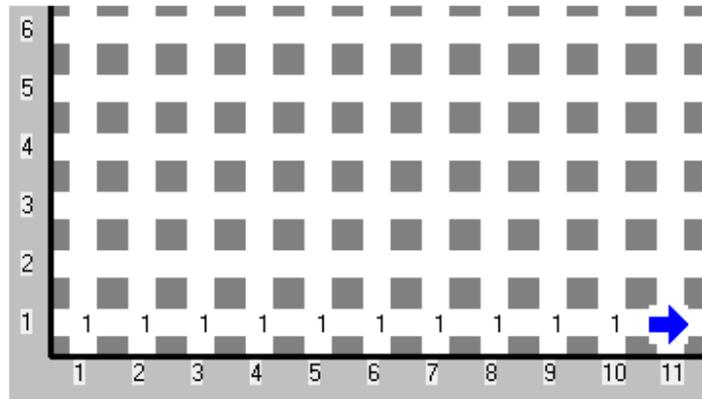
    apagate;
    termina-ejecucion
finalizar-programa

```

**Problema 6.** Iniciando con  $x$  zumbadores en la mochila, Karel deberá dejarlos uno a uno hasta terminarlos. Resuelva el problema utilizando una repetición con condición.



Estado inicial, Karel tiene 10 zumbadores en la mochila



Estado final, Karel tiene 0 zumbadores.

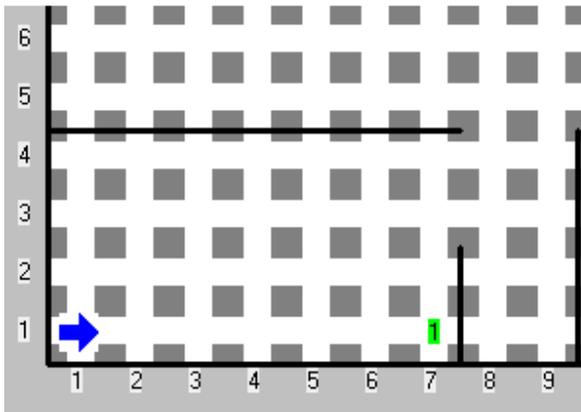
### Programa

```

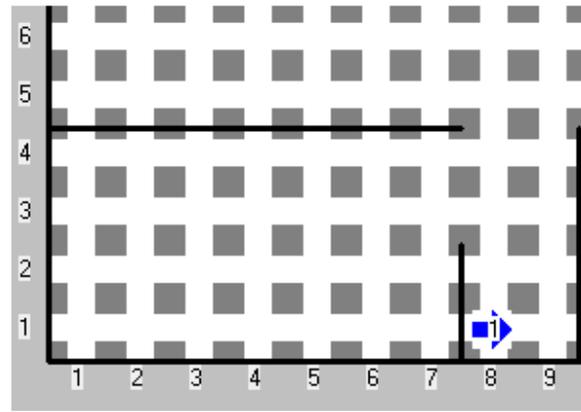
iniciar-programa
  inicia-ejecucion
    mientras algun-zumbador-en-la-mochila hacer inicio
      deja-zumbador;
      avanza;
    fin;
  apagate;
  termina-ejecucion
finalizar-programa

```

**Problema 7.** Orientado al este Karel avanza recogiendo la basura de su cuarto en forma lineal hasta encontrar una barda (2 de alto) la cual tiene que brincar para dejar la basura. Resuelva el problema utilizando repeticiones con condición (*mientras ... hacer*).



Estado inicial, Karel tiene 0 zumbadores



Estado final, Karel tiene 1 zumbador.

### Programa

iniciar-programa

  inicia-ejecucion

    mientras frente-libre hacer inicio

      avanza;

    mientras junto-a-zumbador hacer inicio

      coge-zumbador;

    fin;

  fin;

  gira-izquierda;

  avanza;

  avanza;

  gira-izquierda;

  gira-izquierda;

  gira-izquierda;

  avanza;

  gira-izquierda;

  gira-izquierda;

  gira-izquierda;

  mientras frente-libre hacer inicio

    avanza;

  fin;

  gira-izquierda;

  mientras algun-zumbador-en-la-mochila hacer inicio

    deja-zumbador;

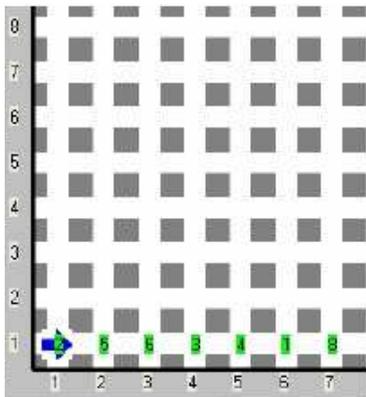
  fin;

  apagate;

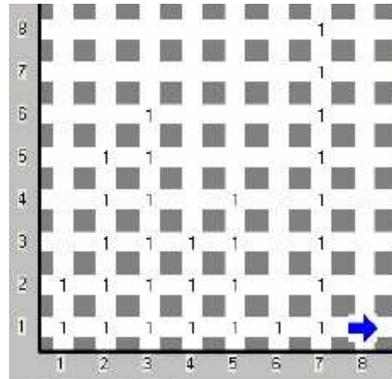
  termina-ejecucion

finalizar-programa

**Problema 8.** Karel deberá hacer una fila de zumbadores en cada avenida, con cada montón que encuentre en la primera calle. Se puede observar un mundo inicial de ejemplo y el mundo como deberá quedar, al finalizar la ejecución. Karel al iniciar no tendrá zumbadores en su mochila



Estado inicial.



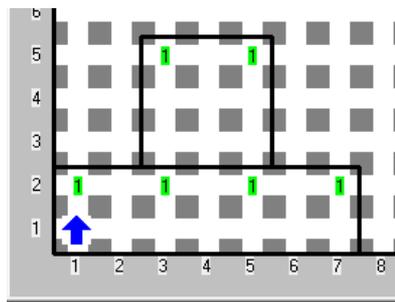
Estado final Karel.

### Programa

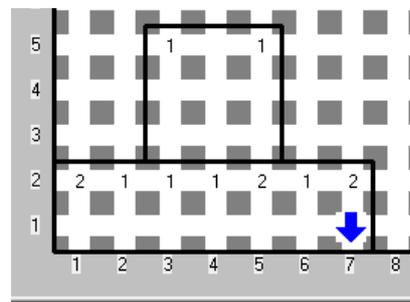
```

iniciar-programa
  inicia ejecución
    mientras junto-a-zumbador hacer
      inicio
        gira-izquierda;
        mientras junto-a-zumbador-en-la-mochila hacer
          inicio
            deja-zumbador;
            avanza;
            fin;
        gira-izquierda;
        gira-izquierda;
        mientras frente-libre hacer
          inicio
            avanza;
            fin;
        gira-izquierda;
        avanza;
        fin;
      apágate;
    termina-ejecucion
  finalizar-programa
  
```

**Problema 9.** Karel se encuentra encerrado, sin embargo debe pasar por los zumbadores y dejar en la pared de enfrente un zumbador por cada espacio vacío. Resuelva el problema usando la condición si-entonces.



Estado inicial.



Estado final Karel.

### Programa

```

iniciar-programa
  inicia-ejecucion
  avanza;
  deja-zumbador;
  si frente-libre entonces
    inicio
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    fin
  sino
    inicio
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    fin;
  si frente-libre entonces
    inicio
    deja-zumbador;
    avanza;
    fin;
  si frente-libre entonces
    inicio
    avanza;
    deja-zumbador;
    fin;
  si frente-libre entonces
    inicio
    avanza;
    deja-zumbador;
    fin;
  si frente-libre entonces
    inicio
    avanza;

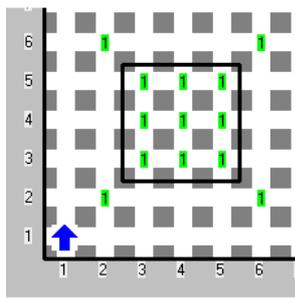
```

```

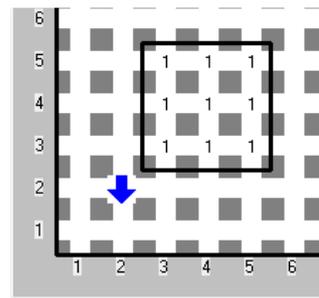
    deja-zumbador;
    fin ;
si frente-libre entonces
    inicio
    avanza;
    deja-zumbador;
    fin;
si frente-libre entonces
    inicio
    avanza;
    deja-zumbador;
    fin;
si frente-libre entonces
    inicio
    avanza;
    deja-zumbador;
    fin
sino
    inicio
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    fin;
apagate;
termina-ejecucion
finalizar-programa

```

**Problema 10.** Karel se encuentra alrededor de su cuarto, sin embargo puede o no tener sus cosas regadas en las esquinas externas, así que karel debe recorrer el cuarto y recoger las cosas que se encuentren tiradas y colocarse en la esquina inferior izquierda, como lo muestran las siguientes imágenes.



Estado inicial.



Estado final Karel.

### Programa

```

iniciar-programa
inicia-ejecucion

```

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

avanza;

si junto-a-zumbador entonces

inicio  
coge-zumbador;  
avanza  
fin

sino

inicio  
avanza;  
fin;  
avanza;  
avanza;  
avanza;

si junto-a-zumbador entonces

inicio  
coge-zumbador;  
gira-izquierda;  
avanza;  
fin

sino

inicio  
gira-izquierda;  
avanza;  
fin;

avanza;

avanza;

avanza;

si junto-a-zumbador entonces

inicio  
coge-zumbador;  
gira-izquierda;  
avanza;  
fin

sino

inicio  
gira-izquierda;  
avanza;  
fin;

avanza;

avanza;

avanza;

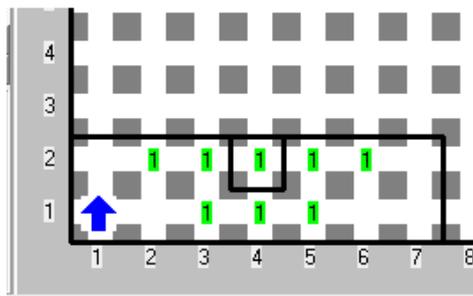
si junto-a-zumbador entonces

```

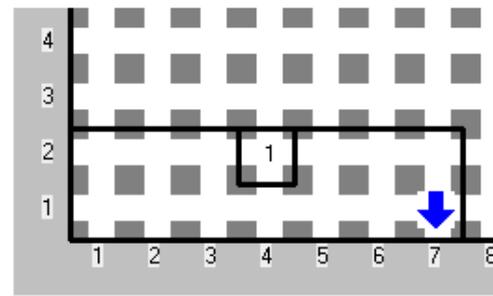
inicio
coge-zumbador;
gira-izquierda;
avanza;
fin
sino
  inicio
  gira-izquierda;
  avanza;
  fin;
avanza;
avanza;
avanza;
  apagate;
termina-ejecucion
finalizar-programa

```

**Problema 11.** Karel se encuentra alrededor de su cuarto, sin embargo se encuentra con obstáculos, ¿Podrá Karen recoger sus cosas a tiempo?



Estado inicial.



Estado final Karel.

### Programa

```

iniciar-programa
  inicia-ejecucion
    avanza;
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
    si junto-a-zumbador o frente-libre entonces
      inicio
        coge-zumbador;
        avanza;
      fin
    sino
      inicio
        gira-izquierda;

```

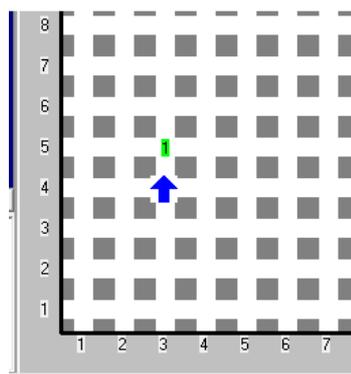
gira-izquierda;  
gira-izquierda;  
avanza;  
fin;  
si frente-libre y junto-a-zumbador entonces  
inicio  
coge-zumbador;  
avanza;  
fin  
sino  
inicio  
coge-zumbador;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
si frente-libre o junto-a-zumbador entonces  
inicio  
coge-zumbador;  
fin;  
gira-izquierda;  
avanza;  
si frente-libre o junto-a-zumbador entonces  
inicio  
coge-zumbador;  
fin;  
avanza;  
si frente-libre o junto-a-zumbador entonces  
inicio  
coge-zumbador;  
fin;  
gira-izquierda;  
avanza;  
si frente-libre o junto-a-zumbador entonces  
inicio  
coge-zumbador;  
fin;  
gira-izquierda;  
gira-izquierda;  
gira-izquierda;  
avanza;  
fin;  
si frente-libre o junto-a-zumbador entonces  
inicio  
coge-zumbador;  
fin;  
avanza;  
si frente-libre o junto-a-zumbador entonces

```

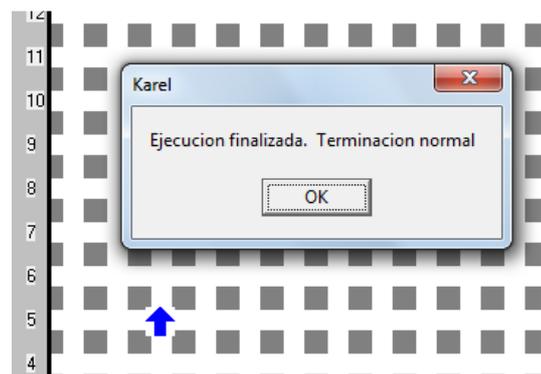
inicio
  coge-zumbador;
  gira-izquierda;
  avanza;
fin
sino
  inicio
    gira-izquierda;
    gira-izquierda;
    gira-izquierda;
    avanza;
  fin;
apagate;
termina-ejecucion
finalizar-programa

```

**Problema 12.** Karel deberá encontrar un zumbador, que solamente se puede encontrar enfrente de él o detrás de él, y deberá recogerlo.



Estado inicial.



Estado final Karel.

### Programa

```

iniciar-programa
  inicia-ejecucion
    avanza;
    si junto-a-zumbador entonces
      inicio
        coge-zumbador;
      fin
    sino
      inicio
        gira-izquierda;
        gira-izquierda;
        avanza;
        avanza;
        coge-zumbador;
      fin
  fin

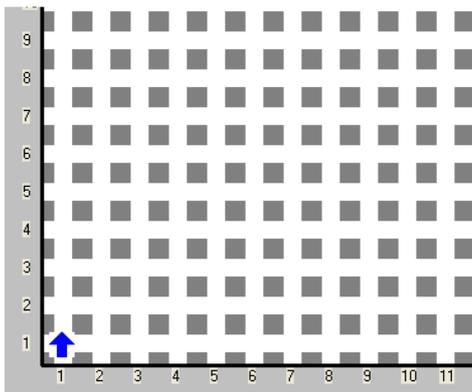
```

```

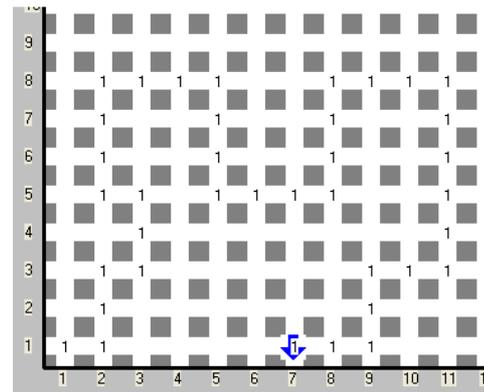
    fin;
  apagate;
  termina-ejecucion
finalizar-programa

```

**Problema 13.** Karel fue de paseo al bosque, estando allí se encontró con una ardilla, el robot desea conocer la madriguera del pequeño animal, por lo que decide seguirla, para no perderse decide marcar el camino con zumbadores, a continuación se muestra los pasos que siguió Karel para llegar a la madriguera de la ardilla.



Karel se encuentra con la ardilla



Karel ha encontrado la madriguera de la ardilla

**Programa**

```

iniciar-programa
  inicia-ejecucion
  repetir 2 veces inicio
    si no-junto-a-zumbador entonces
      inicio
        deja-zumbador;
      fin;
    repetir 3 veces inicio
      gira-izquierda;
    fin;
    avanza;
    deja-zumbador;
    gira-izquierda;
  repetir 2 veces inicio
    avanza;
    deja-zumbador;
  fin;
fin;
gira-izquierda;
avanza;
deja-zumbador;

```

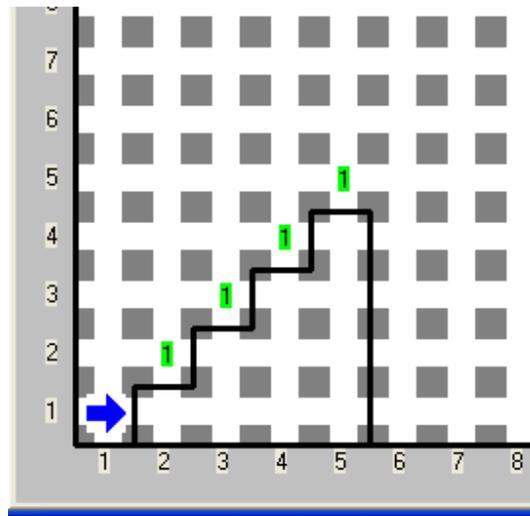
```
repetir 3 veces inicio
  gira-izquierda;
fin;
repetir 2 veces inicio
  avanza;
  deja-zumbador;
fin;
avanza;
deja-zumbador;
repetir 2 veces inicio
  repetir 3 veces inicio
    gira-izquierda;
  fin;
  repetir 3 veces inicio
    avanza;
    deja-zumbador;
  fin;
fin;
repetir 2 veces inicio
  gira-izquierda;
  repetir 3 veces inicio
    avanza;
    deja-zumbador;
  fin;
fin;
repetir 2 veces inicio
  repetir 3 veces inicio
    gira-izquierda;
  fin;
  repetir 3 veces inicio
    avanza;
    deja-zumbador;
  fin;
fin;
repetir 2 veces inicio
  repetir 2 veces inicio
    repetir 2 veces inicio
      avanza;
      deja-zumbador;
    fin;
  repetir 3 veces inicio
    gira-izquierda;
  fin;
fin;
repetir 2 veces inicio
  gira-izquierda;
fin;
fin;
```

apagate;  
termina-ejecucion  
finalizar-programa

## Ejercicios Propuestos

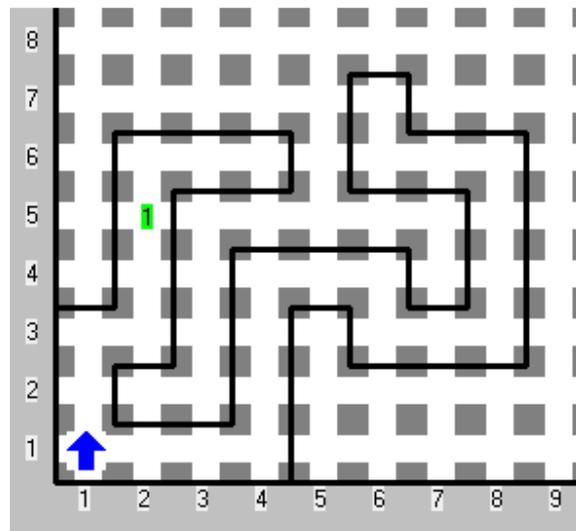
**Problema 1.** Karel está enfrente de una escalera formada por paredes, deberá posicionarse en el último escalón e ir recogiendo los zumbadores que se encuentre en su camino. Utilice un ciclo *repetir n veces* para llegar a la cima.

*Mundo final*



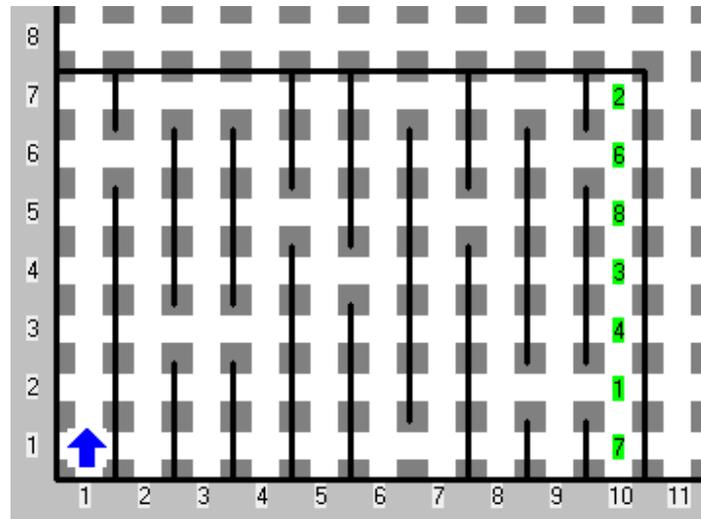
**Problema 2.** Encontrar en un laberinto, un zumbador que tenga una pared a la izquierda utilizando condiciones y estructuras de repetición.

*Mundo Inicial*



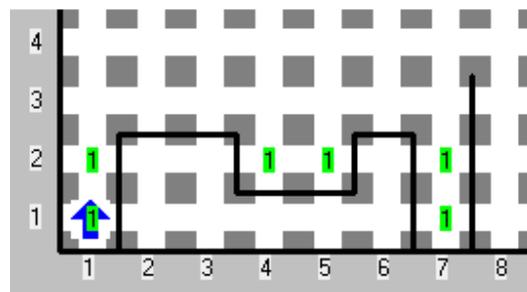
**Problema 3.** Karel se encuentra en un laberinto, existe una línea de zumbadores en la parte derecha de este. Karel deberá llegar hasta esa línea y recoger todos los zumbadores colocándolos en la parte inferior de la misma. Utilice condiciones y estructuras de repetición.

*Mundo Inicial*



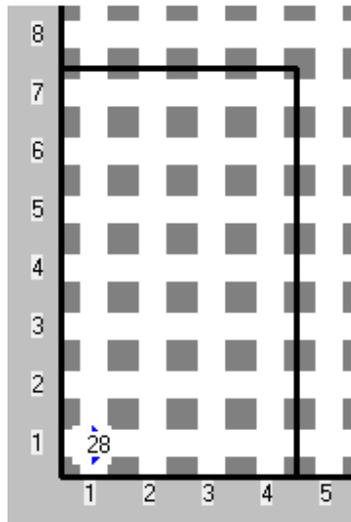
**Problema 4.** Karel deberá rellenar los baches que se encuentran enfrente, los baches se terminan con una pared. Utilice condiciones cada vez que esté de frente a una pared y estructuras de repetición

*Mundo Final*



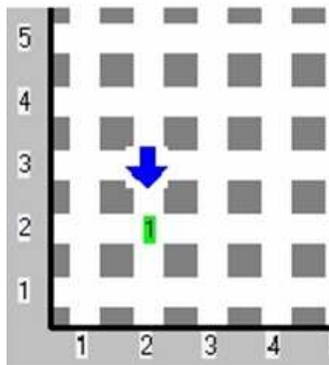
**Problema 5.** Karel deberá calcular el área de un cuadrilátero y colocar el número que la representa, en la esquina inferior izquierda. Karel lleva en su mochila 100 zumbadores, el área no supera 100 zumbadores. Utilice estructuras de repetición con condición.

*Mundo Final*



**Problema 6.** Karel deberá recoger un zumbador que puede haber o no, en la parte de debajo de él (una esquina hacia el sur). Si hacia el sur no hay pared, quiere decir que si hay zumbador, pero si hay pared, Karel no recogerá nada. Para que Karel pueda trabajar deberá empezar orientado al sur, sino está orientado al sur, no hará nada.

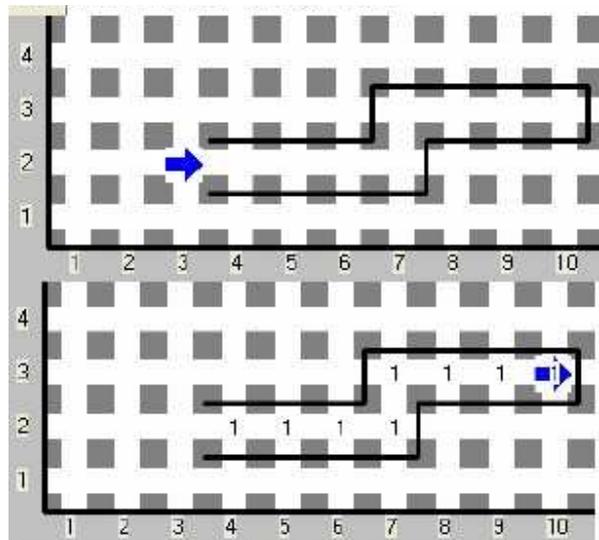
**Mundo Final**



**Problema 7.** Karel se encontrará en alguna esquina del mundo y orientado hacia cualquier punto. Deberás hacer que Karel se oriente hacia el sur y posteriormente se apague.

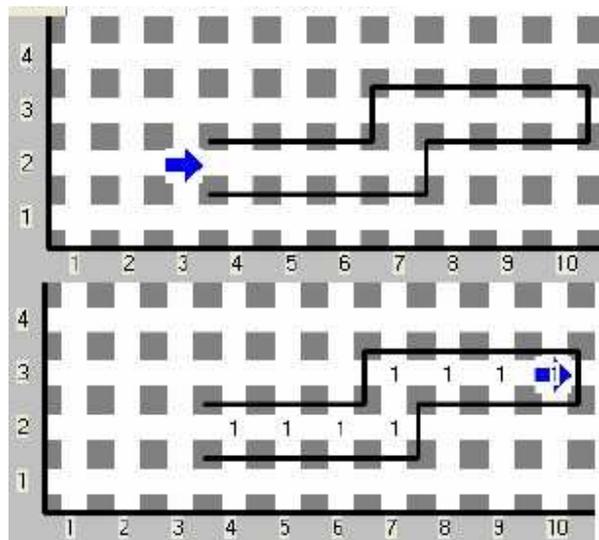
**Problema 8.** Se pide que Karel, deje 1 zumbador en cada esquina que se encuentre dentro de las paredes. Al iniciar la ejecución Karel deberá tener en su mochila 10 zumbadores.

**Mundo Final**



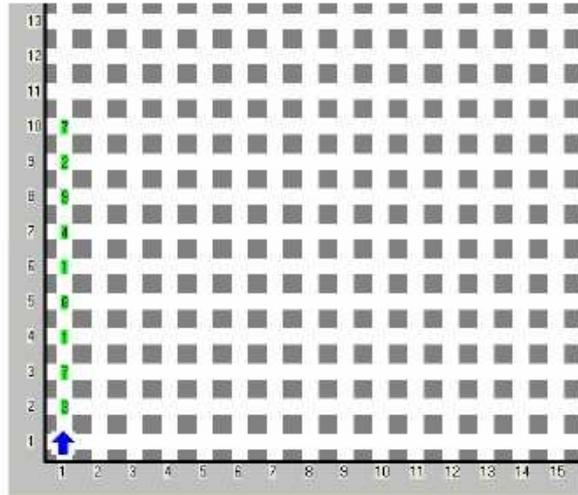
**Problema 9.** Karel resolverá el mismo problema anterior, pero ahora sin usar la sentencia repetir, sino ahora con el mientras. Recuerden los mundos:

*Mundo Final*



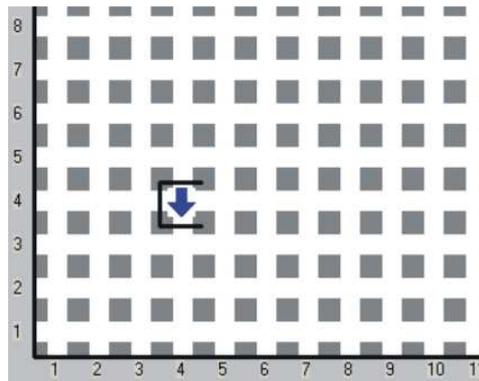
**Problema 10.** Karel, caminará de frente, por las 10 primeras calles, en las cuales habrán zumbadores tirados; él tendrá que recoger todos los zumbadores que encuentre en las primeras 10 calles hacia su frente.

*Mundo Final*

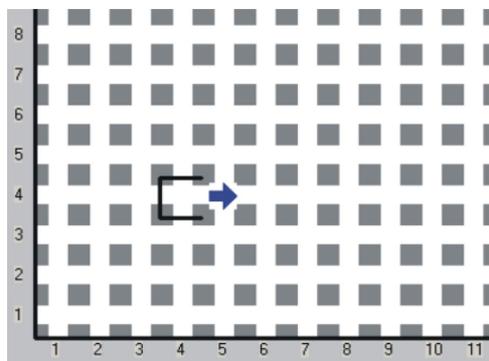


**Problema 11.** Karel se encuentra en una calle que está rodeada inmediatamente por tres paredes, es decir, un lado está sin pared: el de enfrente, el de la izquierda, el de la derecha o el de atrás. Karel deberá “salir” por el lado donde no hay pared y avanzar una calle

*Mundo Inicial*

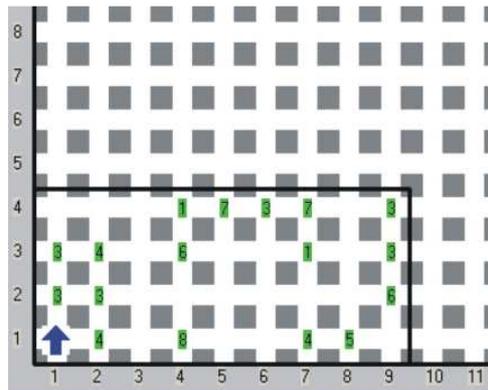


*Mundo Final*

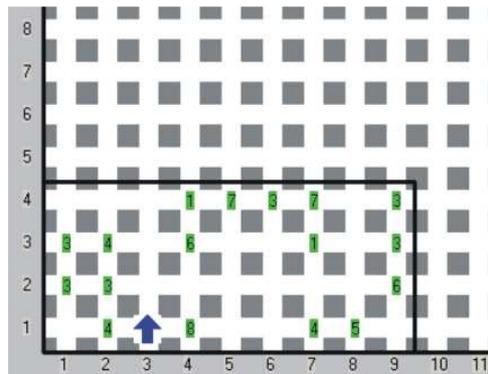


**Problema 12.** Karel se encuentra dentro de un cuadrilátero cubierto de columnas de zumbadores, Karel deberá posicionarse en la columna (calle) donde no exista ningún zumbador, es decir, la calle donde no se encuentre en ninguna esquina de esta, algún zumbador. Al menos existe una columna sin zumbadores.

**Mundo Inicial**

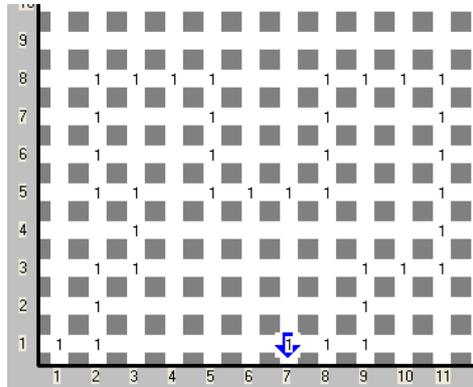


**Mundo Final**



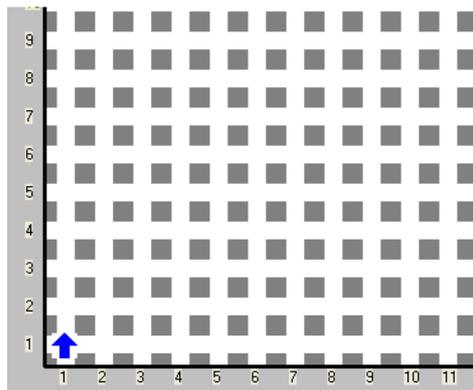
**Problema 13.** Una vez que Karel encontró la madriguera de la ardilla, debe regresar a la entrada del bosque, para ello deberá regresar recogiendo todos los zumbadores que colocó anteriormente.

*Mundo Inicial*



Karel se encuentra en la madriguera de la ardilla.

*Mundo Final*



Karel regresó al principio del bosque.

## 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.