OMIPS 2022 - Solución para *Distancia Manhattan*

Entiende el problema

¿Qué te piden hacer? Karel debe encontrar el único zumbador que hay en el mundo y medir la distancia manhattan que hay desde su posición de inicio hasta él.

¿Qué tienes que entregar como resultado? Un montón de zumbadores igual a la distancia manhattan entre su posición de inicio y el único zumbador del mundo.

¿Dónde tienes que entregar el resultado? En la posición de inicio de Karel.

Analiza el problema

¿Qué tienes que poder hacer para resolver el problema? 1. **Encontrar el único zumbador del mundo:** El zumbador puede estar en cualquier posición con respecto a Karel. 1. **Poder regresar desde el zumbador hasta tu posición de inicio:** El resultado debe dejarse en la posición de inicio de Karel, es necesario poder regresar a ella. 1. **Medir la *distancia Manhattan* entre dos puntos:** Debes poder medir la *distancia Manhattan* entre el inicio y el zumbador.

Este problema tiene 3 subtareas distintas. Cada una de ellas permite implementar los puntos anteriores de distinta manera, por lo que vale la pena revisarlas por separado.

Subtarea 1: Karel inicial en la posición $(1, 1)$ viendo al norte.

- **Encontrar el único zumbador del mundo:** El problema especifica que el mundo es rectangular y sin paredes internas. De modo que un recorrido columna por columna de izquierda a derecha debe ser suficiente para encontrar el zumbador. El siguiente código es una forma clásica de hacer dicho recorrido:

▼ Haz clic para ver el código

```
define-nueva-instrucción super-avanza como inicio
  si frente-libre entonces avanza
  si-no inicio
      gira-izquierda;
      gira-izquierda;
      mientras frente-libre hacer avanza;
      gira-izquierda;
      si frente-libre entonces inicio
          avanza;
          gira-izquierda;
      fin;
  fin;
fin;
```

- **Poder regresar desde el zumbador hasta la posición de inicio:** Dado que la posición de inicio siempre es la $(1, 1)$, regresar a ella es sencillo, basta avanzar desde dónde estés hasta la pared oeste y luego a la pared sur.

- **Medir la *distancia Manhattan* entre dos puntos:** La distancia se tiene que medir entre el zumbador y la posición de inicio, en este caso la $(1, 1)$. Una forma de medirlo es contar la cantidad de avanzas que se hacen hacia el oeste y el sur desde el zumbador. El siguiente código muestra una forma de hacerlo:

▼ Haz clic para ver el código

```
define-nueva-instrucción mide-distancia como inicio
  si frente-libre o no orientado-al-sur entonces inicio
      si frente-bloqueado entonces gira-izquierda;
      si frente-libre entonces inicio
          avanza;
          mide-distancia;
          deja-zumbador;
      fin;
  fin;
fin;
```

Subtarea 2: El zumbador está en la misma fila o la misma columna que Karel.

- **Encontrar el único zumbador del mundo:** Dado que el zumbador se encuentra en la misma fila o la misma columna que Karel siempre será posible encontrarlo avanzando en línea recta desde la posición de Karel hacia alguna de las 4 direcciones (norte, sur, este u oeste).
- **Poder regresar desde el zumbador hasta la posición de inicio:** Karel inicia en cualquier posición y siempre va a avanzar en línea recta. La forma más sencilla de ubicar su posición inicial es marcándola con un zumbador. De esa forma, al encontrar el zumbador del mundo basta con dar media vuelta y regresar hasta encontrar el zumbador que se usó como marca.
- **Medir la *distancia Manhattan* entre dos puntos:** Se puede usar una versión un poco más sencilla que en la subtarea anterior, ya que basta con contar los avanza en línea recta.

▼ Haz clic para ver el código completo de subtarea 2

```
iniciar-programa
  define-nueva-instrucción gira(n) como repetir n veces gira-izquierda;
  define-nueva-instrucción mide-distancia como inicio
    si no junto-a-zumbador entonces inicio
        avanza;
        mide-distancia;
        deja-zumbador;
    fin;
  fin;
  define-nueva-instrucción zumbador-o-pared como inicio
    mientras frente-libre y no junto-a-zumbador hacer avanza;
    gira(2);
  fin;
  define-nueva-instrucción resuelve como inicio
    repetir 4 veces inicio
        si frente-libre entonces inicio
            avanza;
            zumbador-o-pared;
            si junto-a-zumbador entonces inicio
                avanza;
                mide-distancia;
                apagate;
            fin
            si-no inicio
                avanza;
                mientras no junto-a-zumbador hacer avanza;
                gira(2);
            fin;
        fin;
        gira-izquierda;
    fin;
  fin;
  inicia-ejecución
    si junto-a-zumbador entonces coge-zumbador
    si-no inicio
        deja-zumbador;
        resuelve;
    fin;
  termina-ejecución
finalizar-programa
```

Subtarea 3: Sin restricciones adicionales.

Existen distintas formas de resolver esta versión del problema con variantes de las subtareas anteriores. Para ilustrar una idea distinta, la solución oficial no marca nunca la posición de salida sino que va recorriendo recursivamente de columna en columna, de fila en fila contando la cantidad de casillas que ha avanzado. Cuando encuentra el zumabdor deja el número de casillas que ha avanzado y regresa (usando la recursión) copiando ese número hasta la posición de inicio.

▼ Haz clic para ver el código de la solución oficial

```
iniciar-programa
  define-nueva-instrucción norte como
    mientras no-orientado-al-norte hacer gira-izquierda;

  define-nueva-instrucción este como
    mientras no-orientado-al-este hacer gira-izquierda;

  define-nueva-instrucción sur como
    mientras no-orientado-al-sur hacer gira-izquierda;

  define-nueva-instrucción oeste como
    mientras no-orientado-al-oeste hacer gira-izquierda;

  define-nueva-instrucción pared como mientras frente-libre hacer avanza;

  define-nueva-instrucción espacio como
    mientras junto-a-zumbador y frente-libre hacer avanza;

  define-nueva-instrucción gira(n) como repetir n veces gira-izquierda;

  define-nueva-instrucción atras como inicio
    gira(2);
    avanza;
    gira(2);
  fin;

  define-nueva-instrucción deja(n) como repetir n veces deja-zumbador;

  define-nueva-instrucción coge-todo como
    mientras junto-a-zumbador hacer coge-zumbador;

  define-nueva-instrucción ultimo-zumbador como inicio
    espacio;
    si no junto-a-zumbador entonces atras;
  fin;

  define-nueva-instrucción copia como inicio
    si junto-a-zumbador entonces inicio
      coge-zumbador;
      copia;
      deja-zumbador;
    fin si-no
      avanza;
  fin;

  define-nueva-instrucción columna como inicio
    si frente-libre entonces inicio
      avanza;
      columna;
      si junto-a-zumbador entonces inicio
        copia;
        deja-zumbador;
      fin si-no
        avanza;
    fin si-no
      gira(2);
  fin;

  define-nueva-instrucción checa-columna(n) como inicio
    norte;
    columna;
    si no junto-a-zumbador entonces columna;
    si no junto-a-zumbador entonces inicio
      si si-es-cero(n) entonces
        este
      si-no
        oeste;

      si frente-libre entonces inicio
        avanza;
        checa-columna(n);
        si junto-a-zumbador entonces inicio
          copia;
          deja-zumbador;
        fin si-no
          avanza;
      fin si-no
        gira(2);
    fin si-no inicio
      si si-es-cero(n) entonces
        oeste
      si-no
```

```
        este;
    fin;

  fin;

  inicia-ejecución
    este;
    checa-columna(0);
    si no junto-a-zumbador entonces checa-columna(1);
    coge-zumbador;
    apágate;
  termina-ejecución
finalizar-programa
```