

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software).

<<http://www.ati.es/novatica/>>
<<http://www.ati.es/reicis/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies), representa a España en **IFIP** (International Federation for Information Processing) y es miembro de **CLIE** (Centro Latinoamericano de Estudios de Informática) y de **CEGUA** (Confederación of European Computer User Associations). Asimismo tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery) y colabora con diversas asociaciones informáticas españolas.

Consejo Editorial

Guillem Alsina González, Juan Hernández Basora, Albert Jové, Miguel García-Menéndez (presidente del Consejo), Francesc Noguera Puig, Jordi Roca i Marimón

Coordinación Editorial

Encarna Quesada Ruiz <encarna.quesada@ati.es>

Composición y autoedición

Impresión Offset Derra S. L.

Traducciones

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lengua-informatica/>>

Administración

Tomás Brunete, María José Fernández

Secciones Técnicas - Coordinadores

Accesibilidad

Emmanuelle Guillérez y Restrepo (Fundación Sidar), <emmanuelle@sidar.org>

Loïc Martínez Normand (Fundación Sidar), <loic@sidar.org>

Acceso y recuperación de la Información

José María Gómez Hidalgo (Pragsis Technologies), <jmgomez@pragsis.com>

Enrique Puertas Sanz (Universidad Europea de Madrid), <enrique.puertas@universidadeuropea.es>

Administración Pública electrónica

Francisco López Crespo (MAE), <flc@ati.es>

Sebastià Justicia Pérez (Diputación de Barcelona) <sjusticia@ati.es>

Arquitecturas

Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>

José Flich Cardo (Universidad Politécnica de Valencia), <jflich@disca.upv.es>

Auditoría SITIC

Marina Tourinho Troitillo, <marinatourinho@marinatourinho.com>

Sergio Gómez-Landero Pérez (Endesa), <sergio.gomezlandero@endesa.es>

Derecho y tecnologías

Elena Davara Fernández de Marcos (Davara & Davara), <edavara@davara.com>

Enseñanza Universitaria de la Informática

Cristóbal Pareja Flores (DSIP-UCM), <cpareja@sip.ucm.es>

J. Ángel Velázquez Turbide (DLSI, URJC), <angel.velazquez@urjc.es>

Entorno digital personal

Andrés Marín López (Univ. Carlos III), <amarin@it.uc3m.es>

Diego Gachet Páez (Universidad Europea de Madrid), <gachet@uem.es>

Estándares Web

José Carlos del Arco Prieto (TCP Sistemas e Ingeniería), <jcarco@gmail.com>

Gestión del Conocimiento

Joan Baiget Solé (Cap Gemini Ernst & Young), <joan.baiget@ati.es>

Gobierno corporativo de las TI

Manuel Palao García-Suelto (ATI), <manuel@palao.com>

Miguel García-Menéndez (ITI) <mgarciamenendez@ititrendsinsitute.org>

Informática y Filosofía

José Ángel Olivás Varela (Escuela Superior de Informática, UCLM), <joseangel.olivas@uclm.es>

Roberto Feltrero Orjeda (UNED), <rfeltrero@gmail.com>

Informática Gráfica

Miguel Chover Sellés (Universitat Jaume I de Castellón), <chover@lsi.uji.es>

Roberto Vivó Hernando (Eurographics, sección española), <rvivo@dsic.upv.es>

Ingeniería del Software

Luis Fernández Sáenz, Daniel Rodríguez García (Universidad de Alcalá), <luis.fernandez.daniel.rodriguez@uah.es>

Inteligencia Artificial

Vicente Botti Navarro, Vicente Julián Inglada (DSIC-UPV), <vbotti,vinglada@dsic.upv.es>

Interacción Persona-Computador

Pedro M. Latorre Andrés (Universidad de Zaragoza, AIPO), <platorre@unizar.es>

Francisco L. Gutiérrez Vela (Universidad de Granada, AIPO), <fgutierrez@ugr.es>

Lenguajes Informáticos

Oscar Belmonte Fernández (Univ. Jaime I de Castellón), <hbelmonte@lsi.uji.es>

Inmaculada Coma Talay (Univ. de Valencia), <inmaculada.coma@uv.es>

Lingüística computacional

Xavier Gómez Guinovart (Univ. de Vigo), <xggo@uvigo.es>

Modelado de software

Jesus Garcia Molina (DIS-UM), <jmolina@um.es>

Gustavo Rossi (UFPA-UNLP Argentina), <gustavo@sol.info.unlp.edu.ar>

Mundo estudiantil y jóvenes profesionales

Federico G. Mon Trotti (RITSI), <gnu.fede@gmail.com>

Mikel Salazar Peña (Área de Jóvenes Profesionales, Junta de ATI Madrid), <mikelbo_uni@yahoo.es>

Seguridad

Rafael Fernández Calvo (ATI), <rfcalvo@ati.es>

Miguel Sarrías Grifó (ATI), <miquel@sarrias.net>

Redes y servicios telemáticos

Juan Carlos López López (UCLM), <juancarlos.lopez@uclm.es>

Ana Pont Sanjuán (UPV), <apont@disca.upv.es>

Robotica

José Cortés Arenas (Sopra Group), <joscortea@gmail.com>

Juan González Gómez (Universidad Carlos III), <juan@iearobotics.com>

Seguridad

Javier Arellano Bertolin (Univ. de Deusto), <jarellito@deusto.es>

Javier López Muñoz (ETSI Informática-UMA), <jlm@lcc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <jalonso@puentej@dit.upm.es>

Software Libre

Jesus M. Gonzalez Barahona (GSYC-URJC), <jgb@gysc.es>

Fernando Tricas García (Universidad de Zaragoza), <fricas@unizar.es>

Tecnologías para la Educación

Juan Manuel Dodero Beardo (UC3M), <jdodero@inf.uc3m.es>

César Pablo Córcoles Briongo (UOC), <ccorcoles@uoc.edu>

Tecnologías y Empresa

Didac López Viñas (Universidad de Girona), <didac.lopez@ati.es>

Alonso Álvarez García (TID) <aag@tid.es>

Tendencias tecnológicas

Gabriel Martí Fuentes (Interbits), <gabi@atinet.es>

Juan Carlos Vigo (ATI) <juancarlosvigo@atinet.es>

TID y Turismo

Andrés Agayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga), <agayo.guevara@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
Gutierre de Cetina 24, 28017 Madrid • Tfn.914029391 <novatica@ati.es>

Administración y Redacción ATI Cataluña
Calle Àvila 50, 3a planta, local 9, 08005 Barcelona
Tfn.934125235 <secregen@ati.es>

Redacción ATI Andalucía <secreand@ati.es>
Redacción ATI Galicia <secregal@ati.es>

Suscripción y Ventas <novatica.suscripciones@atinet.es>
Publicidad Gutierre de Cetina 24, 28017 Madrid
Tfn.914029391 <novatica@ati.es>

Imprenta: Impresión Offset Derra S.L., Lluís 41, 08005 Barcelona.
Depósito legal: B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVACQ

Portada: "El guardián" - Concha Arias Pérez / © ATI
Diseño: Fernando Agresta / © ATI 2003

editorial

La seguridad digital > 02

en resumen

Nuevos tiempos, nuevos aires > 02

Encarna Quesada Ruiz

noticias de CEPIS

Red sobre temas legales y seguridad CEPIS LSI SIN > 03

Maite Villalba de Benito

monografía

Seguridad digital

Editor invitado: Miguel García-Menéndez

Presentación. La hora de la seguridad digital > 05

Miguel García-Menéndez

El ciberpuzle. Cómo el sentido común puede resolverlo > 09

John McCarthy

In medio stat virtus > 12

Manolo Palao

¿Confía Ud. en los cuidados que su médico les dispensa a sus datos personales? > 17

Kerry Tomlinson

La nueva "3/113" mediática > 22

M^{ra} José de la Calle

¿Quién se hace cargo? > 27

Miguel García-Menéndez

Alfabetización digital. Desconectando los saberes previos de la junta directiva en clave digital > 33

Jeimy J. Cano M.

En el camino hacia la resiliencia > 37

Susana Asensio, Jose Valiente

secciones técnicas

Acceso y recuperación de la información

Benchmark de consultas de agrupamiento y ordenamiento difuso > 41

Soraya Carrasquel, David Coronado, Ricardo Monascal, Rosseline Rodríguez, Leonid Tineo

Gestión del conocimiento

El rol del conocimiento propio en la organización > 47

Joan Baiget i Solé

Tendencias tecnológicas

El éxito de Bitcoin: La economía de la deep web > 52

Roberto José Fernández García

Referencias autorizadas > 59

sociedad de la información

Programar es crear

El problema del robot de exploración de Marte > 65

(Competencia de Programación UTN-FRC 2016, problema 2, enunciado)

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas

El problema del robot de exploración de Marte > 66

(Competencia de Programación UTN-FRC 2014, problema 5, solución)

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas

asuntos interiores

Coordinación editorial / Programación de Novática / Socios Institucionales > 68

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas

Laboratorio de Investigación de Software MsLabs, Dpto. Ing. en Sistemas de Información, Facultad Regional Córdoba - Universidad Tecnológica Nacional (Argentina)

<jotacastillo@gmail.com>,
<diegojserrano@gmail.com>,
<ing.marinacardenas@gmail.com>

El problema del robot de exploración de Marte

El problema consiste en codificar un algoritmo que sea capaz de controlar remotamente un robot. El control del mismo debe hacerse de manera tal de evitar las posiciones de peligro y en el caso de que uno de los desplazamientos ponga en riesgo la misión se debe informarlo por consola.

El problema aclara también, que cualquier posición fuera del área de trabajo del robot se considera como no segura. Es por ello que los controles de los límites de las matrices deben hacerse constantemente en cada paso de un desplazamiento. Aunque el problema no lo indica, asumimos que el descenso del robot ha sido exitoso y que, por ello, siempre partimos de una posición segura.

El problema se resuelve con dos lenguajes de programación alternativos: Java y Python. La solución en Java muestra el uso de una sola clase, con el método estático main que permite realizar la lectura y llenado de la matriz de exploración, la cual se conoce que será cuadrada y binaria. Seguidamente, se realiza la lectura de la cantidad de casos de prueba a analizar, y a través del empleo de una estructura repetitiva se procesan cada uno de estos casos de prueba. Luego, es necesario analizar la secuencia de movimientos de cada desplazamiento par-

El enunciado de este problema apareció en el número 237 de *Novática* (julio-octubre 2016, p. 78).

tiendo del centro de la matriz. Ya que los casos de prueba vienen separados entre sí por una “;”, es necesario realizar un split (división) del String para obtener cada uno de los movimientos del desplazamiento. Esta información nos permite comenzar a recorrer la matriz, y esto se refleja en el ciclo for anidado, puesto que el algoritmo procesa un caso de prueba a la vez.

En la implementación, el par (x,y) representa inicialmente el centro de la matriz y un ciclo for permite recorrer esta matriz en alguna de las cuatro direcciones permitidas y con una determinada cantidad de movimientos (longitud). Cada movimiento en un paso se refleja en la actualización de los valores de las coordenadas x e y, en la dirección que corresponda.

Si en algún momento se detecta una posición insegura, el algoritmo lo refleja mediante el cambio de valor de la bandera seguro. Esto es realizado controlando que las posiciones se mantengan en el rango de la matriz y siempre que no se cumpla la condición “area[x][y] == 1”.

Si en algún momento se detecta una posición no segura para ese desplazamiento, entonces se interrumpe su procesamiento, se

informa esa situación y se prosigue con el análisis del siguiente desplazamiento.

La segunda solución propuesta implementa el mismo algoritmo pero con el lenguaje de programación Python. Se utiliza el script principal y cuatro funciones que simplemente permiten actualizar la posición del desplazamiento, y que es representada por una tupla de dos componentes.

Notamos que el área se crea utilizando listas por comprensión, y que los movimientos se almacenan en una lista de funciones. Esta lista puede ser indizada y mediante una tupla de dos elementos como argumentos, es posible invocar a las funciones de dirección para que actualicen el sentido del desplazamiento.

Este problema también podría ser abordado empleando un enfoque basado en grafos, pero se lo resuelve con un enfoque matricial puesto que naturalmente se representa como una matriz, y además, su recorrido se reduce a simples desplazamientos en formas de filas o columnas.

A continuación se presentan las soluciones en ambos lenguajes de programación.

Solución en Java:

```
import java.util.Scanner;

public class RobotExplorador {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int[][] area;
        int tamaño = sc.nextInt();

        area = new int[tamaño][tamaño];

        for (int i = 0; i < tamaño; i++) {
            for (int j = 0; j < tamaño; j++) {
                area[i][j] = sc.nextInt();
            }
        }

        int c = sc.nextInt();
        sc.nextLine();
    }
}
```

```

int x, y, distancia, direccion, longitud;
for (int i = 0; i < c; i++) {
    x = y = (tamaño-1) / 2;
    boolean seguro = true;
    distancia = 0;
    String l = sc.nextLine();
    String[] linea = l.split(",");

    for (String movimiento : linea) {
        String []tok = movimiento.trim().split(" ");
        direccion = Integer.parseInt(tok[0]);
        longitud = Integer.parseInt(tok[1]);

        for (int j = 0; j < longitud && seguro; j++) {
            switch (direccion) {
                case 0: x--; break;
                case 1: x++; break;
                case 2: y++; break;
                case 3: y--;
            }

            x < 0 || y < 0 || x >= tamaño || y >= tamaño || area[x][y] == 1)
                seguro = false; break;
        }
        distancia += longitud;
        if (!seguro) break;
    }
    System.out.println((seguro) ? distancia : "NO SEGURO");
}
}
}

```

Solución en Python:

```

def arriba(x, y): return x-1,y
def abajo(x, y): return x+1,y
def izquierda(x, y): return x,y+1
def derecha(x, y): return x,y-1

movimiento = [arriba, abajo, izquierda, derecha]

tamaño = int(input())
area = [None] * tamaño

for i in range(tamaño):
    area[i] = [int(x) for x in input().split()]

c = int(input())

for i in range(c):
    movimientos = [x.strip().split() for x in input().split(",")]
    x = y = (tamaño - 1) // 2
    seguro = True
    distancia = 0
    for m in movimientos:
        direccion = int(m[0])
        longitud = int(m[1])

        for j in range(longitud):
            x, y = movimiento[direccion](x, y)
        if (not 0 <= x < tamaño) or (not 0 <= y < tamaño) or area[x][y] == 1:
            seguro = False
            break

        distancia += longitud
        if not seguro: break

    print(distancia if seguro else "NO SEGURO")

```