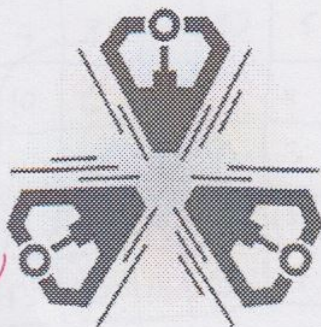


55

(пятьдесят пять)

Жан
(Корнихин Е.В.)



РОБОФЕСТ

Олимпиада школьников
«РОБОФЕСТ-2016»

Заключительный этап по информатике

Фамилия Имя Отчество участника Мезверев

Никита Александрович

Класс 10 Школа Инженерный музей ИГТУ

Дата 16.04.2016 Подпись участника Мезв

Шифр robo16_025

```
--- Solution 48-000002-34200-robo16_025-robo16_025-20160416153825.java ---
package com.company;
```

```
import java.util.Scanner;
```

```
public class Main {
```

```
public static void main(String[] args) {
    new Main();
}
```

```
private Scanner scanner;  
private int width;  
private int height;  
private char[][] map;  
private int[][] marks;
```

```
private StringBuilder movementSequenceBuilder = new StringBuilder(10000);
```

```
public Main() {
    scanner = new Scanner(System.in);
```

```
width = scanner.nextInt();
height = scanner.nextInt();
scanner.nextLine();
map = new char[width][height];
```

```
for (int y = 0; y < height; ++y) {
    String line = scanner.nextLine();
    for (int x = 0; x < width; ++x) {
        map[x][y] = line.charAt(x);
    }
}
```

```
marks = new int[width][height];
int exitX = -1, exitY = -1;
int robotX = -1, robotY = -1;
for (int i = 0; i < width; ++i)
    for (int j = 0; j < height; ++j)
        switch (map[i][j]) {
            case 'E':
                exitX = i;
                exitY = j;
                break;
            case '@':
                robotX = j;
                robotY = i;
                break;
        }
}
```

```
// nīSnīSnīSnīSnīSnīSnīSnīS nīSnīSnīSnīSnīSnīSnīSnīSnīSnīS nīSnīSnīSnīSnīSnīSnīSnīS,
nīSnīSnīSnīSnīSnīSnīSnīS nīSnīSnīSnīSnīSnīSnīSnīSnīSnīSnīSnīSnīS
mark(exitX, exitY, 1);
```

```
check(robotX, robotY, marks[robotX][robotY] + 1, '');
```

```
System.out.println(movementSequenceBuilder.substring(1));
```

```
void check(int x, int y, int previous, char direction) {
    if(x < 0 || y < 0 || x >= width || y >= height)
        return;
    if(marks[x][y] == 0)
```

```

        return;
    if(marks[x][y] >= previous)
        return;

    movementSequenceBuilder.append(direction);

    check(x - 1, y, previous - 1, 'U');
    check(x + 1, y, previous - 1, 'D');
    check(x, y - 1, previous - 1, 'L');
    check(x, y + 1, previous - 1, 'R');
}

void mark(int x, int y, int previous) {
    if(x < 0 || y < 0 || x >= width || y >= height)
        return;
    if(marks[x][y] != 0 && marks[x][y] <= previous)
        return;
    if(map[y][x] == '#')
        return;

    marks[x][y] = previous + 1;

    mark(x - 1, y, previous + 1);
    mark(x + 1, y, previous + 1);
    mark(x, y - 1, previous + 1);
    mark(x, y + 1, previous + 1);
}
}

--- Solution 48-000003-34200-robo16_025-robo16_025-20160416153855.java ---
package com.company;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        new Main();
    }

    private Scanner scanner;
    private int width;
    private int height;
    private char[][] map;
    private int[][] marks;

    private StringBuilder movementSequenceBuilder = new StringBuilder(10000);

    public Main() {
        scanner = new Scanner(System.in);

        width = scanner.nextInt();
        height = scanner.nextInt();
        scanner.nextLine();
        map = new char[width][height];

        for (int y = 0; y < height; ++y) {
            String line = scanner.nextLine();
            for (int x = 0; x < width; ++x) {
                map[x][y] = line.charAt(x);
            }
        }

        marks = new int[width][height];
        int exitX = -1, exitY = -1;
        int robotX = -1, robotY = -1;

```

```

        for (int i = 0; i < width; ++i)
            for (int j = 0; j < height; ++j)
                switch (map[i][j]) {
                    case 'E':
                        exitX = i;
                        exitY = j;
                        break;
                    case '@':
                        robotX = j;
                        robotY = i;
                        break;
                }

        mark(exitX, exitY, 1);

        check(robotX, robotY, marks[robotX][robotY] + 1, '');

        System.out.println(movementSequenceBuilder.substring(1));
    }

    void check(int x, int y, int previous, char direction) {
        if(x < 0 || y < 0 || x >= width || y >= height)
            return;
        if(marks[x][y] == 0)
            return;
        if(marks[x][y] >= previous)
            return;

        movementSequenceBuilder.append(direction);

        check(x - 1, y, previous - 1, 'U');
        check(x + 1, y, previous - 1, 'D');
        check(x, y - 1, previous - 1, 'L');
        check(x, y + 1, previous - 1, 'R');
    }

    void mark(int x, int y, int previous) {
        if(x < 0 || y < 0 || x >= width || y >= height)
            return;
        if(marks[x][y] != 0 && marks[x][y] <= previous)
            return;
        if(map[y][x] == '#')
            return;

        marks[x][y] = previous + 1;

        mark(x - 1, y, previous + 1);
        mark(x + 1, y, previous + 1);
        mark(x, y - 1, previous + 1);
        mark(x, y + 1, previous + 1);
    }
}

--- Solution 48-000004-34200-robo16_025-robo16_025-20160416154325.java ---
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        new Main();
    }

    private Scanner scanner;
    private int width;
    private int height;
    private char[][] map;

```

```

private int[][] marks;

private StringBuilder movementSequenceBuilder = new StringBuilder(10000);

public Main() {
    scanner = new Scanner(System.in);

    width = scanner.nextInt();
    height = scanner.nextInt();
    scanner.nextLine();
    map = new char[width][height];

    for (int y = 0; y < height; ++y) {
        String line = scanner.nextLine();
        for (int x = 0; x < width; ++x) {
            map[x][y] = line.charAt(x);
        }
    }

    marks = new int[width][height];
    int exitX = -1, exitY = -1;
    int robotX = -1, robotY = -1;
    for (int i = 0; i < width; ++i)
        for (int j = 0; j < height; ++j)
            switch (map[i][j]) {
                case 'E':
                    exitX = i;
                    exitY = j;
                    break;
                case '@':
                    robotX = j;
                    robotY = i;
                    break;
            }

    mark(exitX, exitY, 1);

    check(robotX, robotY, marks[robotX][robotY] + 1, '');

    System.out.println(movementSequenceBuilder.substring(1));
}

void check(int x, int y, int previous, char direction) {
    if(x < 0 || y < 0 || x >= width || y >= height)
        return;
    if(marks[x][y] == 0)
        return;
    if(marks[x][y] >= previous)
        return;

    movementSequenceBuilder.append(direction);

    check(x - 1, y, previous - 1, 'U');
    check(x + 1, y, previous - 1, 'D');
    check(x, y - 1, previous - 1, 'L');
    check(x, y + 1, previous - 1, 'R');
}

void mark(int x, int y, int previous) {
    if(x < 0 || y < 0 || x >= width || y >= height)
        return;
    if(marks[x][y] != 0 && marks[x][y] <= previous)
        return;
    if(map[y][x] == '#')
        return;

    marks[x][y] = previous + 1;
}

```

```

        mark(x - 1, y, previous + 1);
        mark(x + 1, y, previous + 1);
        mark(x, y - 1, previous + 1);
        mark(x, y + 1, previous + 1);
    }
}

```

--- Solution 48-000005-34200-robo16_025-robo16_025-20160416154538.java ---
import java.util.Scanner;

```

public class Main {
    public static void main(String[] args) {
        new Main();
    }

    private Scanner scanner;
    private int width;
    private int height;
    private char[][] map;
    private int[][] marks;

    private StringBuilder movementSequenceBuilder = new StringBuilder(10000);

    public Main() {
        scanner = new Scanner(System.in);

        width = scanner.nextInt();
        height = scanner.nextInt();
        scanner.nextLine();
        map = new char[width][height];

        for (int y = 0; y < height; ++y) {
            String line = scanner.nextLine();
            for (int x = 0; x < width; ++x) {
                map[x][y] = line.charAt(x);
            }
        }

        marks = new int[width][height];

        int exitX = -1, exitY = -1;
        int robotX = -1, robotY = -1;

        for (int i = 0; i < width; ++i)
            for (int j = 0; j < height; ++j)
                switch (map[i][j]) {
                    case 'E':
                        exitX = i;
                        exitY = j;
                        break;
                    case '@':
                        robotX = j;
                        robotY = i;
                        break;
                }

        mark(exitX, exitY, 1);

        check(robotX, robotY, marks[robotX][robotY] + 1, '');

        System.out.println(movementSequenceBuilder.substring(1));
    }

    void check(int x, int y, int previous, char direction) {
        if(x < 0 || y < 0 || x >= width || y >= height)

```

```

        return;
    if(marks[x][y] == 0)
        return;
    if(marks[x][y] >= previous)
        return;

    movementSequenceBuilder.append(direction);

    check(x - 1, y, previous - 1, 'U');
    check(x + 1, y, previous - 1, 'D');
    check(x, y - 1, previous - 1, 'L');
    check(x, y + 1, previous - 1, 'R');
}

void mark(int x, int y, int previous) {
    if(x < 0 || y < 0 || x >= width || y >= height)
        return;
    if(marks[x][y] != 0 && marks[x][y] <= previous)
        return;
    if(map[y][x] == '#')
        return;

    marks[x][y] = previous + 1;

    mark(x - 1, y, previous + 1);
    mark(x + 1, y, previous + 1);
    mark(x, y - 1, previous + 1);
    mark(x, y + 1, previous + 1);
}
}

--- Solution 48-000007-34200-robo16_025-robo16_025-20160416154853.java ---
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        new Main();
    }

    private Scanner scanner;
    private int width;
    private int height;
    private char[][] map;
    private int[][] marks;

    private StringBuilder movementSequenceBuilder = new StringBuilder(10000);

    public Main() {
        scanner = new Scanner(System.in);

        width = scanner.nextInt();
        height = scanner.nextInt();
        scanner.nextLine();
        map = new char[width][height];

        for (int y = 0; y < height; ++y) {
            String line = scanner.nextLine();
            for (int x = 0; x < width; ++x) {
                map[x][y] = line.charAt(x);
            }
        }

        marks = new int[width][height];
        int exitX = -1, exitY = -1;
        int robotX = -1, robotY = -1;

```

```

        for (int i = 0; i < width; ++i)
            for (int j = 0; j < height; ++j)
                switch (map[i][j]) {
                    case 'E':
                        exitX = i;
                        exitY = j;
                        break;
                    case '@':
                        robotX = j;
                        robotY = i;
                        break;
                }

        mark(exitX, exitY, 1);

        check(robotX, robotY, marks[robotX][robotY] + 1, '');

        System.out.println(movementSequenceBuilder.substring(1));
    }

    void check(int x, int y, int previous, char direction) {
        if(x < 0 || y < 0 || x >= width || y >= height)
            return;
        if(marks[x][y] == 0)
            return;
        if(marks[x][y] >= previous)
            return;

        movementSequenceBuilder.append(direction);

        check(x - 1, y, previous - 1, 'U');
        check(x + 1, y, previous - 1, 'D');
        check(x, y - 1, previous - 1, 'L');
        check(x, y + 1, previous - 1, 'R');
    }

    void mark(int x, int y, int previous) {
        if(x < 0 || y < 0 || x >= width || y >= height)
            return;
        if(marks[x][y] != 0 && marks[x][y] <= previous)
            return;
        if(map[y][x] == '#')
            return;

        marks[x][y] = previous + 1;

        mark(x - 1, y, previous + 1);
        mark(x + 1, y, previous + 1);
        mark(x, y - 1, previous + 1);
        mark(x, y + 1, previous + 1);
    }
}

```

--- Task B ---

```

--- Solution 48-000011-34200-robo16_025-robo16_025-20160416161803.java ---
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        new Main();
    }

    private Scanner scanner;
    private int width;

```



```

private int height;
private int doorTimeout = 0;
private boolean isDoorOpen = true;
private int currentTurn = 0;
private int doorWaiting = -1;
private char[][] map;
private int[][] marks;

private StringBuilder movementSequenceBuilder = new StringBuilder(10000);

public Main() {
    scanner = new Scanner(System.in);

    width = scanner.nextInt();
    height = scanner.nextInt();
    doorTimeout = scanner.nextInt();
    scanner.nextLine();
    map = new char[width][height];

    for (int y = 0; y < height; ++y) {
        String line = scanner.nextLine();
        for (int x = 0; x < width; ++x) {
            map[x][y] = line.charAt(x);
        }
    }

    marks = new int[width][height];
    int exitX = -1, exitY = -1;
    int robotX = -1, robotY = -1;

    for (int i = 0; i < width; ++i)
        for (int j = 0; j < height; ++j)
            switch (map[i][j]) {
                case 'E':
                    exitX = i;
                    exitY = j;
                    break;
                case '@':
                    robotX = j;
                    robotY = i;
                    break;
            }

    mark(exitX, exitY, 1);

    check(robotX, robotY, marks[robotX][robotY] + 1, '', 0);

    System.out.println(movementSequenceBuilder.substring(1));
}

void check(int x, int y, int previous, char direction, int turnNumber) {
    if(x < 0 || y < 0 || x >= width || y >= height)
        return;
    if(marks[x][y] == 0)
        return;
    if(marks[x][y] >= previous)
        return;

    while(!checkDoor() && (map[y][x] == 'C' || map[y][x] == 'O'))
        movementSequenceBuilder.append('S');

    movementSequenceBuilder.append(direction);

    check(x - 1, y, previous - 1, 'U', turnNumber + 1);
    check(x + 1, y, previous - 1, 'D', turnNumber + 1);
    check(x, y - 1, previous - 1, 'L', turnNumber + 1);
    check(x, y + 1, previous - 1, 'R', turnNumber + 1);
}

```

```

    }

    boolean checkDoor() {
        currentTurn++;
        doorWaiting++;
        if(isDoorOpen && doorWaiting >= 2) {
            isDoorOpen = false;
            currentTurn = 1;
        } else if(!isDoorOpen && currentTurn >= doorTimeout) {
            isDoorOpen = true;
            doorWaiting = 0;
        }
        System.out.println("Checking door " + isDoorOpen);
        return isDoorOpen;
    }

    void mark(int x, int y, int previous) {
        if(x < 0 || y < 0 || x >= width || y >= height)
            return;
        if(marks[x][y] != 0 && marks[x][y] <= previous)
            return;
        if(map[y][x] == '#')
            return;

        marks[x][y] = previous + 1;

        mark(x - 1, y, previous + 1);
        mark(x + 1, y, previous + 1);
        mark(x, y - 1, previous + 1);
        mark(x, y + 1, previous + 1);
    }
}

--- Solution 48-000012-34200-robo16_025-robo16_025-20160416163318.java ---
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        new Main();
    }

    private Scanner scanner;
    private int width;
    private int height;
    private int doorTimeout = 0;
    private boolean isDoorOpen = true;
    private int currentTurn = 0;
    private int doorWaiting = -1;
    private char[][] map;
    private int[][] marks;

    private StringBuilder movementSequenceBuilder = new StringBuilder(10000);

    public Main() {
        scanner = new Scanner(System.in);

        width = scanner.nextInt();
        height = scanner.nextInt();
        doorTimeout = scanner.nextInt();
        scanner.nextLine();
        map = new char[width][height];

        for (int y = 0; y < height; ++y) {
            String line = scanner.nextLine();
            for (int x = 0; x < width; ++x) {
                map[x][y] = line.charAt(x);
            }
        }
    }
}

```

```

    }
}

marks = new int[width][height];
int exitX = -1, exitY = -1;
int robotX = -1, robotY = -1;

for (int i = 0; i < width; ++i)
    for (int j = 0; j < height; ++j)
        switch (map[i][j]) {
            case 'E':
                exitX = i;
                exitY = j;
                break;
            case '@':
                robotX = j;
                robotY = i;
                break;
        }

mark(exitX, exitY, 1);

check(robotX, robotY, marks[robotX][robotY] + 1, '', 0);

System.out.println(movementSequenceBuilder.substring(1));
}

void check(int x, int y, int previous, char direction, int turnNumber) {
    if(x < 0 || y < 0 || x >= width || y >= height)
        return;
    if(marks[x][y] == 0)
        return;
    if(marks[x][y] >= previous)
        return;

    while(!checkDoor() && (map[y][x] == 'C' || map[y][x] == 'O'))
        movementSequenceBuilder.append('S');

    movementSequenceBuilder.append(direction);

    check(x - 1, y, previous - 1, 'U', turnNumber + 1);
    check(x + 1, y, previous - 1, 'D', turnNumber + 1);
    check(x, y - 1, previous - 1, 'L', turnNumber + 1);
    check(x, y + 1, previous - 1, 'R', turnNumber + 1);
}

boolean checkDoor() {
    currentTurn++;
    doorWaiting++;
    if(isDoorOpen && doorWaiting >= 2) {
        isDoorOpen = false;
        currentTurn = 1;
    } else if(!isDoorOpen && currentTurn >= doorTimeout) {
        isDoorOpen = true;
        doorWaiting = 0;
    }
    return isDoorOpen;
}

void mark(int x, int y, int previous) {
    if(x < 0 || y < 0 || x >= width || y >= height)
        return;
    if(marks[x][y] != 0 && marks[x][y] <= previous)
        return;
    if(map[y][x] == '#')
        return;
}

```

```
marks[x][y] = previous + 1;

mark(x - 1, y, previous + 1);
mark(x + 1, y, previous + 1);
mark(x, y - 1, previous + 1);
mark(x, y + 1, previous + 1);
    }
}
```