****



**Department of Computer Science & Engineering**

**School of Technology**

**S.Y. B. Tech**

**Academic year: 2020 – 21**

Synopsis On

**“Karel The Robot”**

Under the guidance of

Mrs. Shubhada Marwadkar

**BY**

1. Sanyukta Phatate - 40
2. Pratiksha Patil - 44
3. Dhanashri Patil - 45
4. Aishwarya Patil - 54

**SYNOPSIS**

1. **Title of Mini Project: “**Karel The Robot**”**
2. **Introduction**

Did you know that the word “robot” is almost 100 years old? It was ﬁrst introduced in 1920, in the science-ﬁction theatrical play R.U.R., written by Karel ˇCapek. As a tribute to this Czech writer an educational programming language was named Karle many years later at Stanford university. Your task is to implement an interpreter of a simpliﬁed version of this programming language. The Karel programming language controls a robot named Karel, who lives in a grid of unit squares. Some of the squares are free, while others contain a barrier. Karel always occupies one of the free squares and faces one of the four cardinal directions.

The two basic commands are “move forward” and “turnleft”. The language also provides conditional and looping statements. The main educational potential of the language lies in the possibility of deﬁning new procedures for more complex tasks.

1. **Problem Statement**

We have to develop the Interpreter Having name the karel who lives in a grid of unit squares while some of the squares and faces one of the four cordinal directions. The basic command are “moveforward” and “turnleft”. Karel cannot move if there is barrier in the next square in Karel’s current heading i.e. North(n), South(s), East(e), West(w) which is satisfied if and only if Karels current heading is North, south, east, west respectively.

**Objectives:**

To experience the use of data structure and problem solving skills in project. To provide the system that will interpret the Karel through the square grids except barriers as per our command.

1. **Description of the working of the system**

The ﬁrst line of input contains four integers r, c, d, and e, where r and c (1 ≤ r,c ≤ 40) are the dimensions of the grid in which Karel lives, d (0 ≤ d ≤ 26) is the number of procedure dentitions, and e (1 ≤ e ≤ 10) is the number of programs to be executed. Then follow r lines describing the grid (running north to south), each containing c characters (running west to east), each character being either ‘.’ (Denoting a free square) or ‘#’ (denoting a barrier). All squares outside this given area are considered barriers, which mean Karel may never leave the area. Each of the next d lines contains a procedure deﬁnition, associating a procedure name (one uppercase letter) with a program forming the procedure body. No procedure name is deﬁned more than once. Procedure bodies may contain invocations of procedures that have not yet been deﬁned. The last 2e lines describe the programs to be executed. Each such description consists of a pair of lines. The ﬁrst line of each pair contains two integers i and j and a character h, where i (1 ≤ i ≤ r) is the row and j (1 ≤ j ≤ c) is the column of Karel’s initial position, and h ∈{n,s,e,w}represents Karel’s initial heading. It is guaranteed that the initial position is a free square. The second line of each pair contains a program to be executed from that initial position. and only contain invocations of procedures that are deﬁned. The lines with procedure deﬁnitions and programs to be executed contain no whitespace characters.

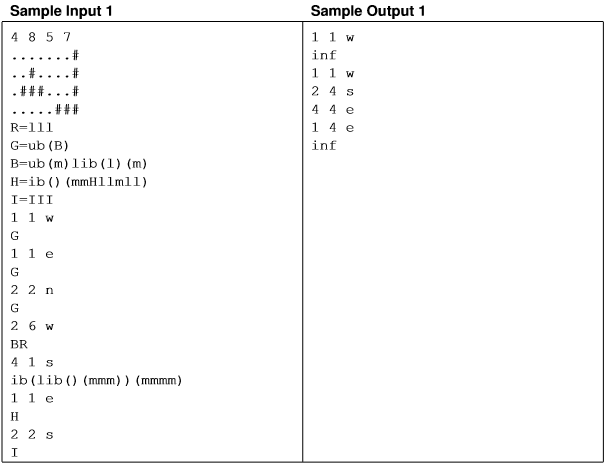


Figure Number 1: Sample Input and Output

1. **Algorithm**

1. START

2. Check the Heading position of the KAREL.

3. Check if the barrier is in the path i.e. ib().

4. If “NO” then “Move forward”.

5. If “YES” then the direction of head i.e e(EAST), w(WEST), n(NORTH), s(SOUTH) and repeat from step number 2.

6. Check if all the squares are visited expect barriers.

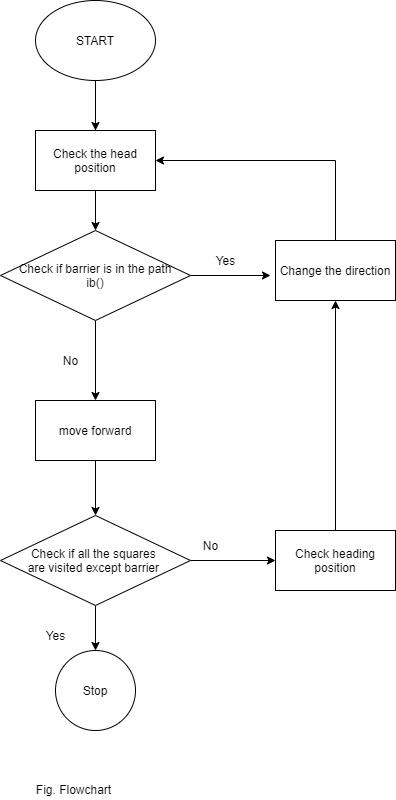
7. If “NO” then check heading and repeat from step number 5.

8. If “YES” then End the program.

1. **Data Structures:**

* 2D Array
* Array traversal methods

1. **Flow chart**



1. **Requirements**

* Hardware Requirements - Personal computer with standard configuration.

Processor: Intel Core i3, RAM:2GB

System Type: 32 bit Operating System

* Software Requirements-

Programming Language: (C)

1. **References**

<http://karel.sourceforge.net/>

<https://www.cs.mtsu.edu/~untch/karel/>

<https://en.wikipedia.org/wiki/Karel_(programming_language)>

<https://www.youtube.com/watch?v=FkYvKUEADi0>