Refer Examples section for better understanding.

Time Limit (secs)Snake and Ladder is a board game consisting of snakes and ladders, where the player must reach End position, starting from Start position. Here, snake on the

board makes player demotes the player to a lower numbered square and ladder promotes player to higher numbered square on the board.

For e.g., given below is the snake and ladder board, where S('pos') represents snake and 'pos' indicates where the player's coin will move down to once a player lands

on that square. Similarly, L('pos') indicates ladder and 'pos' indicates where the player's coin will move up to once the player lands on that square.

Player always starts from the Start square and must go towards End based on the rolls of the die.

You are supposed to find if it is possible for a player to reach the End or not, based on die inputs. If it is possible, display 'Possible' with number of snakes

and ladders encountered during his/her play else display 'Not possible' along with number of snakes, number of ladders and square where the player's coin has ended at.

Note: A player can reach the end if he has the exact number on die input to reach end.

For e.g., if the player is at 99, he/she can reach end only if the die input is 1 and not any other input. So, he/she must wait till input on die is 1.

The actual Snake and Ladder board will look as depicted below. This format will be used for providing inputs.

End 99 S(2) 97 96 95 94 93 92 91

81 82 83 84 85 86 87 88 89 90

80 79 78 77 76 L(99) 74 73 72 71

61 62 S(22) 64 65 66 67 68 69 70

60 59 58 57 56 55 54 53 52 51

41 42 43 44 L(68) 46 47 48 49 50

40 39 S(6) 37 36 35 34 33 32 31

21 22 23 24 25 26 27 28 29 30

20 19 18 17 S(9) 15 14 13 12 11

Start 2 3 4 5 6 7 8 9 10

6 6 6 6 6 6 6 3 5 2 1=>Possible

Start 2 3 L(36) 5 6 7 8 9 10

11 12 13 14 15 16 17 18 19 20

21 22 23 24 25 26 27 28 29 30

31 32 33 34 35 36 37 38 L(96) 40

41 42 43 44 45 46 47 48 49 50

51 52 S(14) 54 55 56 57 58 59 60

61 62 63 64 65 66 67 68 69 70

71 72 73 74 75 S(34) 77 78 79 80

81 82 83 84 85 86 87 88 89 90

91 92 93 94 95 96 97 S(12) 99 End

Constraints

1 <= die_inputs <= 6

Number of die inputs >= 0

Input

First 10 lines contains snake and ladder board where each line has 10 tokens separated by a space. The tokens can either be integers, Start, End, S(number), L(number)

where

Integer denotes the square number

Start denotes the left bottom position on the board from where player start the game

End denotes the left top position on the board

S(number) denotes that the current square has a snake that will take you down to a square number mentioned in the parenthesis.

L(number) denotes that the current square has a ladder that will take you up to a square number mentioned in the parenthesis.

Second line contains die_inputs separated by a space.

Output

Find if the player is possible to reach the End or not, based on die_inputs and the board. If it is possible, display 'Possible' with number of snakes and ladders

encountered during his/her play else display 'Not possible' along with number of snakes, number of ladders and the square where the player's coin has ended at.

Print all the outputs delimited by a space.

1

PROGRAM CODE:

```
package utilp;

import java.util.Scanner;

public class SnakeAndLadder

{

  public static void main(String[] args)

  {

    System.out.println("enter the values for the board");

    Scanner sc=new Scanner(System.in);

    String values[]=new String[100];

    for(int i=0;i<values.length;i++)

    {

      if(sc.hasNext()) //to check if there are any more elements to scan(used to prevent exceptions such as "NoSuchElementExceptions")

      {

        values[i]=sc.next();//next is used to retrieve the next element present in file/array, it is often paired with hasnext() to prevent the above mentioned exception

      }

    }

    //left to right

    int m=0;

    String s="left";

    int count=0;

    for(int i=values.length-1;i>=0;i--)

    {

      if(s.equals("left")) //if we use equalIgnoresCase, it ignores whether it is uppercase or lowercase

      {

        System.out.print(" "+values[i]);
```

```
      count++;
      if(count==10)
      {
        System.out.println();
        s="right";
        m=i;
        count=0;
        m=m-10;
      }
    }
    //right to left
    else if(s.equals("right"))
    {
      System.out.print(" "+values[m]);
      m++;
      count++;
      if(count==10)
      {
        System.out.println();
        s="left";
        m=i;
        count=0;
      }
    }
  }
  int position = 0;
  while (position <= 100)
```

```
    {


        System.out.println("Throw dice");
        int num = sc.nextInt(); //nextInt reads the next token of input as integer. it waits for user to enter an integer and press enter


        if (num < 1 || num > 6)
        {
           System.out.println("Invalid input. Please throw the dice again.");
           continue; //skips to next iteration without checking any other code in this loop body
        }
        position += num;
        if (position == 100)
        {
           System.out.println("Congratulations! You reached the end of the board.");
           break;
        }
        else if (position > 100)
        {
           System.out.println("Position exceeds 100. Please throw the dice again.");
           position -= num; // Undo the last move(i.e, the move which makes the position above 100)
           continue;
        }


        System.out.println("New position: " + position);
        String element = values[position - 1];
        if (element != null && element.charAt(0) == 'S') //charAt(0) retrieves the character at the first index (index 0) of the element string
```

```
        {
            String strNum = element.substring(2, 4); //S(32) here (2,4) means include element at position 2 until 3 [(2,4) 2 is inclusive and 4 is exclusive]
            int backNum = Integer.parseInt(strNum); //parseInt(strNum) converts the string present in strNum to integer and stores it in backNum
            position = backNum;
            System.out.println("Oops you landed on snake, go back to " + backNum + " position");
            System.out.println("New position after snake: " + position);
        } else if (element != null && element.charAt(0) == 'L')
        {
            String strNum = element.substring(2, 4);
            int forwardNum = Integer.parseInt(strNum);
            position = forwardNum;
            System.out.println("hooray, You climbed a ladder. Go forward to" + forwardNum + " position");
            System.out.println("New position after climbing the ladder: " + position);
        }
    }
  }
}
```

OUTPUT:

```
_______

---------

enter the values for the board

Start 2 3 L(36) 5 6 7 8 9 10

11 12 13 14 15 16 17 18 19 20

21 22 23 24 25 26 27 28 29 30

31 32 33 34 35 36 37 38 L(96) 40
```

41 42 43 44 45 46 47 48 49 50

51 52 S(14) 54 55 56 57 58 59 60

61 62 63 64 65 66 67 68 69 70

71 72 73 74 75 S(34) 77 78 79 80

81 82 83 84 85 86 87 88 89 90

91 92 93 94 95 96 97 S(12) 99 End

End 99 S(12) 97 96 95 94 93 92 91

81 82 83 84 85 86 87 88 89 90

80 79 78 77 S(34) 75 74 73 72 71

61 62 63 64 65 66 67 68 69 70

60 59 58 57 56 55 54 S(14) 52 51

41 42 43 44 45 46 47 48 49 50

40 L(96) 38 37 36 35 34 33 32 31

21 22 23 24 25 26 27 28 29 30

20 19 18 17 16 15 14 13 12 11

Start 2 3 L(36) 5 6 7 8 9 10

Throw dice

2

New position: 2

Throw dice

5

New position: 7

Throw dice

3

New position: 10

Throw dice

5

New position: 15

Throw dice

6

New position: 21

Throw dice

1

New position: 22

Throw dice

2

New position: 24

Throw dice

5

New position: 29

Throw dice

5

New position: 34

Throw dice

3

New position: 37

Throw dice

2

New position: 39

hooray, You climbed a ladder. Go forward to96 position

New position after climbing the ladder: 96

Throw dice

2

New position: 98

Oops you landed on snake, go back to 12 position

New position after snake: 12

Throw dice

4

New position: 16

Throw dice

5

New position: 21

Throw dice

1

New position: 22

Throw dice

2

New position: 24

Throw dice

4

New position: 28

Throw dice

6

New position: 34

Throw dice

4

New position: 38

Throw dice

5

New position: 43

Throw dice

4

New position: 47

Throw dice

6

New position: 53

Oops you landed on snake, go back to 14 position

New position after snake: 14

Throw dice

6

New position: 20

Throw dice

4

New position: 24

Throw dice

5

New position: 29

Throw dice

6

New position: 35

Throw dice

2

New position: 37

Throw dice

1

New position: 38

Throw dice

4

New position: 42

Throw dice

3

New position: 45

Throw dice

6

New position: 51

Throw dice

4

New position: 55

Throw dice

1

New position: 56

Throw dice

3

New position: 59

Throw dice

2

New position: 61

Throw dice

5

New position: 66

Throw dice

2

New position: 68

Throw dice

5

New position: 73

Throw dice

6

New position: 79

Throw dice

1

New position: 80

Throw dice

5

New position: 85

Throw dice

5

New position: 90

Throw dice

6

New position: 96

Throw dice

3

New position: 99

Throw dice

1

Congratulations! You reached the end of the board.