**Daily Test**    **Happy Coding from necse**  ☑ **SkillRack**

Time Left:    00:00:00

## Chess Piece - Validate Move

An **8*8** chessboard is given below.
a8 b8 c8 d8 e8 f8 g8 h8
a7 b7 c7 d7 e7 f7 g7 h7
a6 b6 c6 d6 e6 f6 g6 h6
a5 b5 c5 d5 e5 f5 g5 h5
a4 b4 c4 d4 e4 f4 g4 h4
a3 b3 c3 d3 e3 f3 g3 h3
a2 b2 c2 d2 e2 f2 g2 h2
a1 b1 c1 d1 e1 f1 g1 h1

The program must accept the name of a chess piece (**B** - **Bishop** or **K** - **Knight** or **R** - **Rook**) and two positions **P1**, **P2** as the input. The program must print **Yes** if the given piece can move from P1 to P2. Else the program must print **No** as the output.

**Bishop:** In Chess, a bishop can move diagonally. The movement can happen till the end of the board is reached or another piece (like Rook, Knight, Bishop, Pawn etc is encountered).

**Knight:** In Chess, a knight can move two squares vertically and one square horizontally, or two squares horizontally and one square vertically (with both forming the shape of an L).

**Rook:** In Chess, a rook can move along a row or a column. The movement can happen till the end of the board is reached or another piece (like Rook, Knight, Bishop, Pawn etc is encountered).

**Note:** Assume that there are no pieces other than the given piece in the chessboard.

**Input Format:**
The first line contains the name of a chess piece.
The second line contains the two positions P1 and P2 in the chess board.

**Output Format:**
The first line contains Yes or No.

**Example Input/Output 1:**
Input:
B
d4 b6

Output:
Yes

Explanation:
The given piece is **Bishop**.
**P1 = d4** and **P2 = b6**.
The bishop can move from **d4** to **b6**.
Hence Yes is printed as the output.

**Example Input/Output 2:**
Input:
K
d4 b6

Output:
No

**Example Input/Output 3:**
Input:
R
g7 a7

Output:
Yes

**Max Execution Time Limit: 50 millisecs**

Ambiance

Java ( 12.0)

⤢

```java
1   // Timeout F*cks --- THis is java ...we need time.
2   import java.util.*;
3   public class Hello {
4
5       public static void main(String[] args) {
6
7           Scanner sc = new Scanner(System.in);
8
9           char item = sc.next().charAt(0);
10
11          Hashtable<String,Integer> ht = new Hashtable<>();
12          String start = sc.next();
13
14          if(item=='B'){
15
16              int num = start.charAt(1)-'0';
17              int bkp = num;
18              int bkp2 = num;
19              for(char i=start.charAt(0);i<='h';i++){
20                  ht.put(i+""+bkp++,1);
21                  ht.put(i+""+bkp--,1);
22              }
23
24              bkp = num;
25              bkp2 = num;
26              for(char i=start.charAt(0);i>='a';i--){
27                  ht.put(i+""+bkp++,1);
28                  ht.put(i+""+bkp--,1);
29              }
30
31          }
32          if(item=='R'){
33
34              for(int i=1;i<=8;i++){
35                  ht.put(start.charAt(0)+""+i,1);
36                  ht.put(((char)('a'-1+i))+""+start.charAt(1),1);
37              }
38          }
39          if(item=='K'){
40
41              char curr = start.charAt(0);
42              ht.put(((char)(curr-2))+""+(((int)(start.charAt(1)-'0'))+1),1);
43              ht.put(((char)(curr-2))+""+(((int)(start.charAt(1)-'0'))-1),1);
44              ht.put(((char)(curr+2))+""+(((int)(start.charAt(1)-'0'))+1),1);
45              ht.put(((char)(curr+2))+""+(((int)(start.charAt(1)-'0'))-1),1);
46              ht.put(((char)(curr-1))+""+(((int)(start.charAt(1)-'0'))+2),1);
47              ht.put(((char)(curr-1))+""+(((int)(start.charAt(1)-'0'))-2),1);
48              ht.put(((char)(curr+1))+""+(((int)(start.charAt(1)-'0'))+2),1);
49              ht.put(((char)(curr+1))+""+(((int)(start.charAt(1)-'0'))-2),1);
50
51          }
52
53          String end = sc.next();
54
55  //      System.out.println(ht);
56
57          if(ht.containsKey(end)){
58              System.out.println("Yes");
59          }
60          System.out.println("No");
61
62      }
63  }
```

**Code did not pass the execution**

— ✕

Hello.java:42: error: bad operand types for binary operator '-'
    ht.put(((char)(curr-2))+""+((int)(start.charAt(1)-'0'))-1,1);
               ^

```
    first type:  String
  second type: int
Hello.java:44: error: bad operand types for binary operator '-'
    ht.put(((char)(curr+2))+""+((int)(start.charAt(1)-'0'))-1,1);
                                                          ^
  first type:  String
  second type: int
Hello.java:46: error: bad operand types for binary operator '-'
    ht.put(((char)(curr-1))+""+((int)(start.charAt(1)-'0'))-2,1);
                                                          ^
  first type:  String
  second type: int
Hello.java:48: error: bad operand types for binary operator '-'
    ht.put(((char)(curr+1))+""+((int)(start.charAt(1)-'0'))-2,1);
                                                          ^
  first type:  String
  second type: int
4 errors
```

Save    Run