

[codingame](#), [power](#), [of](#), [thor](#)

[codingame] Power of Thor

```
SOLVE IT      , Mjöllnir      ....
가?

:      (if...) .

      가      “      ”      “      .      .      ,

가  40,      18      .      ,

      ,

• N (North)
• NE (North-East)
• E (East)
• SE (South-East)
• S (South)
• SW (South-West)
• W (West)
• NW (North-West)

      .

      :

•      가      .
•      가
```

```
<sxh perl> select(STDOUT); $| = 1; # DO NOT REMOVE
```

Auto-generated code below aims at helping you parse

the standard input according to the problem statement.

LX: the X position of the light of power

LY: the Y position of the light of power

TX: Thor's starting X position

TY: Thor's starting Y position

```
chomp($tokens=<STDIN>); ($LX, $LY, $TX, $TY) = split(/ /,$tokens);
```

game loop

```
while (1) {
```

```
    chomp($E = <STDIN>); # The level of Thor's remaining energy, representing
    the number of moves he can still make.
    # Write an action using print
    # To debug: print STDERR "Debug messages...\n";
```

```
    print "SE\n"; # A single line providing the move to be made: N NE E SE S
    SW W or NW
```

```
} </sxh>
```

```
<sxh perl> select(STDOUT); $| = 1; # DO NOT REMOVE
```

Auto-generated code below aims at helping you parse

the standard input according to the problem statement.

LX: the X position of the light of power

LY: the Y position of the light of power

TX: Thor's starting X position

TY: Thor's starting Y position

```
chomp($tokens=<STDIN>); ($LX, $LY, $TX, $TY) = split(/ /,$tokens);
```

```
$x = $LX - $TX; $y = $LY - $TY;
```

```
# game loop while (1) {
```

```
    chomp($E = <STDIN>); # The level of Thor's remaining energy, representing  
    the number of moves he can still make.
```

```
    # Write an action using print
```

```
    # To debug: print STDERR "Debug messages...\n";
```

```
    print STDERR "energy:", $E, "\n";
```

```
    print STDERR "X:", $x, "\tY:", $y, "\n";
```

```
    if ($x > 0) {
```

```
        if ($y > 0) {
```

```
            $d = "SE";
```

```
            $x--;
```

```

        $y--;
    }
    elsif ($y < 0) {
        $d = "NE";
        $x--;
        $y++;
    }
    else{
        $d = "E";
        $x--;
    }
}
elsif ($x < 0) {
    if ($y > 0) {
        $d = "SW";
        $x++;
        $y--;
    }
    elsif ($y < 0) {
        $d = "NW";
        $x++;
        $y++;
    }
    else{
        $d = "W";
        $x++;
    }
}
else{
    if ($y > 0) {
        $d = "S";
        $y--;
    }
    elsif ($y < 0) {
        $d = "N";
        $y++;
    }
}
print $d,"\n"; # A single line providing the move to be made: N NE E SE S
SW W or NW

```

} </sxh>

if

<sxh perl> select(STDOUT); \$| = 1; # DO NOT REMOVE

chomp(\$tokens=<STDIN>); (\$LX, \$LY, \$TX, \$TY) = split(/ /,\$tokens);

```
$x = $LX - $TX; $y = $LY - $TY;
```

```
# game loop while (1) {
```

```
    chomp($E = <STDIN>); # The level of Thor's remaining energy, representing  
    the number of moves he can still make.
```

```
    $d = "";  
    if ($y > 0) {$d.="S";$y--}  
    elsif ($y < 0) {$d.="N";$y++}  
    if ($x > 0) {$d.="E";$x--}  
    elsif ($x < 0) {$d.="W";$x++};  
    print $d,"\n"; # A single line providing the move to be made: N NE E SE S  
    SW W or NW
```

```
} </sxh>
```

Plugin Backlinks:

From:

<https://moro.kr/> - **Various Ways**

Permanent link:

<https://moro.kr/open/power-of-thor>

Last update: **2020/06/02 09:25**

