Roll the Dice

Write pseudocode to solve Roll the Dice

```
Define: the first character of string start with index 0

Define: length(string : pattern) mean get total length of the string

Define: charAt(string: pattern ,integer: i) mean get character of index i

Define: read() mean get input from the problem

Define: Dice as an object in this problem and it has own attribute which is
        integer : Dice.Top = 1
        integer : Dice.Bottom = 6
        integer : Dice.Front = 2
        integer : Dice.Back = 5
        integer : Dice.Left = 3
        integer : Dice.Right = 4
Define: x_rotate(integer:a,integer:b,integer:c,integer:d) :
                Dice.Front <- a
                Dice.Bottom <- b
                Dice.Back <- c
                Dice.Top <- d
Define: y_rotate(integer:a,integer:b,integer:c,integer:d) :
                Dice.Top <- a
                Dice.Left <- b
                Dice.Bottom <- c
                Dice.Right <- d
Define: z_rotate(integer:a,integer:b,integer:c,integer:d) :
                Dice.Front <- a
                Dice.Left <- b
                Dice.Back <- c
                Dice.Right <- d
```

```
Start
    Let pattern:string <- read()
    For Let i <- 0 To length(pattern) – 1 Step i By 1 Then
        If charAt(pattern, i) == ‘F’ Then
            x_rotate(Dice.Top, Dice.Front, Dice.Bottom, Dice.Back)
        Else if charAt(pattern,i) == ‘B’ Then
            x_rotate(Dice.Bottom, Dice.Back, Dice.Top, Dice.Front)
        Else if charAt(pattern,i) == ‘L’ Then
            y_rotate(Dice,Right, Dice.Top, Dice.Left, Dice.Bottom)
        Else if charAt(pattern,i) == ‘R’ Then
            y_rotate(Dice.Left, Dice.Bottom, Dice.Right, Dice.Top)
        Else if charAt(pattern, i) == ‘C’ Then
            z_roll(Dice.Right, Dice.Front, Dice.Left, Dice.Back)
        Else if charAt(pattern,i) == ‘D’ Then
            z_roll(Dice.Left, Dice.Back, Dice.Right, Dice.Front)
        Endif
    Endfor
    Display Dice.Front
End
```

2. what are inputs outputs conditions.

Input:

- The String of pattern that represent a rotation of the dice

Output:

- The number of the front side after finish rotating

Condition:

- Predefine condition (entity condition)
  - -Dice only have number 1 to 6
  - -Dice start with
  - -Top side is number 1
  - -Bottom side is number 6
  - -Front side is number 2
  - -Back side is number 5
  - -Left side is number 3
  - -Right side is number 4
- Input condition:
  - Input should be only Characters in this set {F, B, L, R, C, D}
  - And each Character mean:
    - “F” means rotate the dice forward
    - “B” means rotate the dice backward
    - “L” means rotate the dice in counterclockwise by front side
    - “R” means rotate the dice in clockwise by front side
    - “C” means rotate the dice in clockwise by top side
    - “D” means rotate the dice in counterclockwise by top side
  - Example if we give an input pattern “FBBL” it should mean:
    - Rotate the dice 1 forward, 2 backward and 1 counterclockwise by front side
- Output condition:
  - Output should be only a number that in range 1 to 6 and it much represent a current front number of the dice

3. show that your algorithm is a correct algorithm.

To prove that algorithm is correct. This is 3 input testcase That I am going to follow to

1. FBBL

Expect 6

1. Start position

Current value:

Front	Back	Top	Bottom	Left	Right
2	5	1	6	3	4

2. F

in the code if pattern is F it will called the predefine function named x_rotate and it will reassign the properties of dice as follow

```
x_rotate(Dice.Top, Dice.Front, Dice.Bottom, Dice.Back)
```

now in function x_rotate the Dice will reassign as

```
x_rotate(a<-1, b<-2, c<-6, d<-5) :
            Dice.Front <- 1
            Dice.Bottom <- 2
            Dice.Back <- 6
            Dice.Top <- 5
```

Current value after F is pass

Front	Back	Top	Bottom	Left	Right
1	6	5	2	3	4

3. B

in the code if pattern is F it will called the predefine function named x_rotate and it will reassign the properties of dice as follow

x_rotate(Dice.Bottom, Dice.Back, Dice.Top, Dice.Front)

now in function x_rotate the Dice will reassign as

```
x_rotate(a<-2, b<-6, c<-5, d<-1) :
        Dice.Front <- 2
        Dice.Bottom <- 6
        Dice.Back <- 5
        Dice.Top <- 1
```

Current value after B is pass

Front	Back	Top	Bottom	Left	Right
2	5	1	6	3	4

4. B

in the code if pattern is F it will called the predefine function named x_rotate and it will reassign the properties of dice as follow

x_rotate(Dice.Bottom, Dice.Back, Dice.Top, Dice.Front)

now in function x_rotate the Dice will reassign as

```
x_rotate(a<-6, b<-5, c<-1, d<-2) :
        Dice.Front <- 6
        Dice.Bottom <- 5
        Dice.Back <- 1
        Dice.Top <- 2
```

Current value after B is pass

Front	Back	Top	Bottom	Left	Right
6	1	2	5	3	4

5. L

in the code if pattern is F it will called the predefine function named y_rotate and it will reassign the properties of dice as follow

```
y_rotate(Dice,Right, Dice.Top, Dice.Left, Dice.Bottom)
```

now in function y_rotate the Dice will reassign as

```
y_rotate(a<-4, b<-2, c<-3,d<-5) :
        Dice.Top <- 4
        Dice.Left <- 2
        Dice.Bottom <- 3
        Dice.Right <- 5
```

Current value after B is pass

Front	Back	Top	Bottom	Left	Right
6	1	4	3	2	5

Current value after finished the pattern

Front	Back	Top	Bottom	Left	Right
6	1	4	3	2	5

Output is number of the front side of current position which is 6

Front	Back	Top	Bottom	Left	Right
6	1	4	3	2	5

Expect output is also 2

Thus, this algorithm is correct

Sahachan Tippimwong 622115039