## Coding Test

* A backend REST service which takes the json file as a POST request and returns the output.

JAVA + REST SERVICE + JSON(INPUT DOSYASI) + JSON OUTPUT DOSYASI

# The problem

Mars Surveillance Robot ( Mars Keşif Robotu )

Instructions: moving forward (F) or backwards (B), turning (L or R), taking samples (S), getting energy (E).

Komutlar: ileri haraket ( F ) ---- geri haraket ( B ) … sola dön ( L ) ---- sağa dön ( R )

Numune topla (S) << Numuneler: Fe, Se,W, Si, Zn>>

Enerji şarj et ( E )

Each of the commands will consume a given quantity of the battery.

Not: Robotun başlangıç enerjisi olacak ( örnek: 50 birim ). Her haraket robotun mevcut enerjisini azaltacak

* Move Forward (F):  ileri haraket ( F )
* Consumes 3 battery units. 3 birim enerji azalır
* Move the unit one square forward in the current facing direction

Robotun yönünün dönük olduğu yöne 1 birim gider ( doğu batı kuzey güney )

* Move Backwards (B): geri haraket ( B )
* Consumes 3 battery units. 3 birim enerji azalır
* Move the unit one square backwards from the current facing direction

Robotun yönünün dönük olduğu yönün tam tersine 1 birim gider(doğu batı kuzey güney)

* Turn Left (L):  sola dön ( L )
* Consumes 2 battery units. 2 birim enerji azalır
* Changes the facing direction 90º to the right. Robotun yönünü 90 derece sağa döndürür
* Turn Right (R): sağa dön ( R )
* Consumes 2 battery units. 2 birim enerji azalır

Changes the facing direction 90º to the left. Robotun yönünü 90 derece sola döndürür

* Take Sample (S):  Numune topla (S)
* Consumes 8 battery units. 8 birim enerji azalır
* Takes and stores a sample of whatever material is primary in the current

location. Bulunduğu koordinattaki maddenin numunesini alır

Extend solar panels (E):  Enerji şarj et ( E )

* Consumes 1 battery unit 1 birim enerji azalır
* Recharges 10 battery units. 10 birim enerji artar ( şarj eder )

**Numuneler**

* Fe: Ferrum. A deposit of iron. Demir ( Fe )
* Se: Selenium. A deposit of selenium. Selenium ( Se )
* W: Water. A deposit that contains water. Su ( W )
* Si: Silicon. A deposit that contains silicon. Silicon ( Si )
* Zn: Zinc. A deposit that contains zinc. Çinko ( Zn )
* **Obs**: An obstacle cell in which the robot can’t go. ( Robotun giremediği bölge )

Whenever the robot detects an obstacle ahead of the execution of the command, it must automatically apply a back off strategy *instead,* in order to continue with the execution. The robot contains a list of back off strategies to try in order, if the execution of 1 strategy results in hitting another obstacle, the robot will jump to the next strategy (battery will nonetheless be consumed):

Robot, **Obs** ile karşılaşırsa otomatik olarak aşağıdaki stratejilerin birini uygulamak zorundadır. İlk önce 1. Stratejiyi uygular. 1. Strateji de Obs’ye denk geliyorsa 2. Strateji’yi uygular. 2. Stateji de Obs’ye denk geliyorsa 3. Strateji. Bu şekilde 7. Stratejiye kadar devam eder.

**Battery, her haraket sonucunda azalacaktır. Aşağıdaki 7 stratejinin de ilk adımı “E” ( Enerji Şarj et ) olduğundan ilk önce 1 azalıp sonra 10 eklenecektir ve diğer adımlar yapılacaktır**

1. E, R, F  Enerji şarj et ( E ), sağa dön ( R ), ileri git ( F )
2. E, L, F Enerji şarj et ( E ), sola dön ( L ), ileri git ( F )
3. E, L, L, F Enerji şarj et ( E ), sola dön ( L ), sola dön ( L ), ileri git ( F )
4. E, B, R, F Enerji şarj et ( E ), geri haraket ( B ), sağa dön ( R ), ileri git ( F )
5. E, B, B, L, F Enerji şarj et ( E ), geri haraket ( B ), geri haraket ( B ), sola dön ( L ), ileri git ( F )
6. E, F, F Enerji şarj et ( E ), ileri git ( F ), ileri git ( F )
7. E, F, L, F, L, F Enerji şarj et ( E ), ileri git ( F ), sola dön (L), ileri git (F), sola dön (L) ileri git (F)

Examples

Example – Input

{

  "terrain": [

    ["Fe", "Fe", "Se"],

    ["W", "Si", "Obs"],

    ["W", "Obs", "Zn"]

  ],

  "battery": 50,

  "commands": [

    "F", "S", "R", "F", "S", "R", "F", "L", "F", "S"

    ],

  "initialPosition": {

    "location" : {

      "x" : 0,

      "y" : 0

    },

    "facing" : "East"

  }

}

The terrain ( harita, alan ) represents a multidimensional array describing the resources on the terrain.

Terrain: multidimensional array ( haritadaki numune kaynaklarını gösteriyor)

The array represents the terrain in a direct way, with 0 based indexes, that is, in the example above, for X = 1

0 we have “Fe” and for X = 0 and Y = 1 we have “W”.

Array’de koordinatlar 0’dan başlıyor. Yani sol üst köşe ( x,y ) 🡺 (0,0)’a denk geliyor

Battery represents the starting battery for the robot.

**“battery” : robotun başlangıç enerji birimi**

Commands represents the set of commands to execute.

**Commands: tüm komutlar ( ileri sağa sola numune topla …. )**

InitialPosition represents the initial position of the robot. Location (X, Y) represents the starting position of the robot in the 0 based array as described above.

**InitialPosition: Robotun başlangıç koordinatları**

Facing represents the initial facing direction of the Robot, there are 4 possible valid values:

* North,
* East
* South
* West

**Facing: Robotunun yönünün dönük olduğu yön.**

**4 değeri olabilir**

* North ( Kuzey )
* East ( Doğu )
* South ( Güney )
* West ( Batı )

Example – Output

Running the Example above will produce the following output:

**Rest service POST request’in Response cevabı aşağıdaki JSON dosyası gibi olmalıdır**

{

    "VisitedCells": [

        {

            "X": 0,

            "Y": 0

        },

        {

            "X": 1,

            "Y": 0

        },

        {

            "X": 1,

            "Y": 1

        },

        {

            "X": 0,

            "Y": 1

        },

        {

            "X": 0,

            "Y": 2

        }

    ],

    "SamplesCollected": [

        "Fe",

        "Si",

        "W"

    ],

    "Battery": 8,

    "FinalPosition": {

        "Location": {

            "X": 0,

            "Y": 2

        },

        "Facing": "South"

    }

}

**JSON OUTPUT DETAYLARI**

* **VisitedCells** fields contains all the fields that the robot has traversed on its execution without any repetitions.

**VisitedCells: Ziyaret edilen koordinatlar**

* **SamplesCollected** contains all the fields that the robot has taken samples of, including

duplicates, that is, If the robot samples Water twice then it should appear twice.

**SamplesCollected: Toplanan Numuneler ( Robot 2 defa herhangi bir numuneye geldiyse, listede aynı numune 2 defa gözükmelidir )**

* **Battery** contains the remaining battery of the unit after the execution

**Battery: Kalan Enerji birimi miktarı**

* **FinalPosition** contains the final position of the robot after the execution.

**FinalPosition: Son koordinatı**