**Q118: Mutant Flatworld Expolrers**

原翻譯者：untitled

給你一塊矩形土地的長寬，再依序給定每個機器人的初始位置狀況及一連串的指令集，你必須用你的程式求出每個機器人最後的位置狀況。

一個機器人的位置狀況包括了其坐標（ x 坐標， y 坐標），和它面向的方向（用 N , S , E , W 來分別代表北、南、東、西）。至於一個機器人所收到的指令集，是一個由字母 ' *L* ' ， ' *R* ' ， 和 ' *F* ' 所構成的字串，其分別代表：

* **左轉（Left）**：機器人在原地往左轉 90 度。
* **右轉（Right）**: 機器人在原地往右轉 90 度。
* **前進（Forward）**: 機器人往其面向的方向向前走一格，且不改變其面向之方向。

從坐標 (*x*,*y*) 走至 (*x*,*y*+1) 的這個方向我們定義為*北方*。

因為此矩形土地是有邊界的，所以一旦一個機器人走出邊界掉落下去，就相當於永遠消失了。不過這個掉下去的機器人會留下「標記 ( scent ) 」，提醒以後的機器人，避免他們從同一個地方掉下去。掉下去的機器人會把標記，留在他掉落之前所在的最後一個坐標點。所以對於以後的機器人，當他正位在有標記的地方時，這個機器人就會忽略會讓他掉下去的指令。

**Input**

輸入裡的第一列有2個正整數，代表這個矩形世界右上角頂點的坐標，其中假設這個世界的左下角頂點坐標為 ( 0 , 0 )。

接下來是若干組有關機器人的初始位置狀況和指令集，每個機器人2列。第一列為位置狀況，包括了兩個整數和一個字元（ N , S , E 或 W ），代表機器人初始的位置坐標以及機器人最初所面對的方向。第二列則是指令集，是一個由 ' L ' ， ' R ' 和 ' F ' 所組成的字串。輸入以 end-of-file 作為結束。

各機器人是依序動作的，也就是說，直到一個機器人作完他全部的動作，下一個機器人才會開始動作。

所有機器人的初始位置皆會在矩形土地上，不會落在外面。任何坐標的最大值皆不會超過 50 。每個指令集的長度皆不會超過 100 個字元長。

**Output**

對於每一個機器人，你都必須輸出其最後所在的坐標和面對的方向。如果一個機器人會掉落出此世界外，你必須先輸出他在掉落前，最後的所在位置和面對的方向，再多加一個字： LOST 。

**Sample Input**

5 3

1 1 E

RFRFRFRF

3 2 N

FRRFLLFFRRFLL

0 3 W

LLFFFLFLFL

**Sample Output**

1 1 E

3 3 N LOST

2 3 S

程式碼

Imports System.IO

Public Class Form1

Dim fr As New FileInfo("input.txt")

Dim fw As New FileInfo("output.txt")

Dim sw As StreamWriter = fw.CreateText

Dim gg() As String = {"E", "S", "W", "N"}

Dim ii(3), aa As String

Dim n, a(1), x As Integer

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

If fr.Exists = False Then

End

ElseIf fw.Exists = False Then

fw.Create()

End If

Dim sr As StreamReader = fr.OpenText

ii(0) = Trim(sr.ReadLine)

For i = 1 To Len(ii(0)) '取前面兩個數

If Microsoft.VisualBasic.Mid(ii(0), i, 1) <> " " Then

a(n) &= Microsoft.VisualBasic.Mid(ii(0), i, 1)

ElseIf Microsoft.VisualBasic.Mid(ii(0), i, 1) = " " Then

n += 1

End If

Next

n = 0

Dim ob(2) As String

Do '取締2航開始ㄉ每行數 並帶入指令

ii(n) = Trim(sr.ReadLine)

If ii(n) = "" Then Exit Do

If (n + 1) Mod 2 = 0 Then

ob(x) = ii(n)

x += 1

ReDim Preserve ob(x)

End If

n += 1

ReDim Preserve ii(n)

Loop

Dim oa((n + 1) / 2, 3) As String

For i = 0 To n - 1 Step 2 '分解數

x = 0

For j = 1 To Len(ii(i))

If Microsoft.VisualBasic.Mid(ii(0), j, 1) <> " " Then

oa(i / 2, x) &= Microsoft.VisualBasic.Mid(ii(i), j, 1)

ElseIf Microsoft.VisualBasic.Mid(ii(0), j, 1) = " " Then

x += 1

End If

Next

Next

For i = 0 To n / 2

For j = 1 To Len(ob(i))

aa = Microsoft.VisualBasic.Mid(ob(i), j, 1)

Select Case aa

Case "R"

For k = 0 To 3

If oa(i, 2) = gg(k) Then

If k = 3 Then

oa(i, 2) = gg(0)

Else

oa(i, 2) = gg(k + 1) : Exit For

End If

End If

Next

Case "L"

For k = 0 To 3

If oa(i, 2) = gg(k) Then

If k = 0 Then

oa(i, 2) = gg(3) : Exit For

Else

oa(i, 2) = gg(k - 1) : Exit For

End If

End If

Next

Case "F"

For l = 0 To n / 2

If oa(l, 3) = "LOSE" And oa(l, 0) = oa(i, 0) And oa(l, 1) = oa(i, 1) Then

Select Case oa(i, 2)

Case "E"

If Val(oa(i, 0)) + 1 < a(0) Then

oa(i, 0) = Val(oa(i, 0)) + 1

End If

Exit For

Case "W"

If Val(oa(i, 0)) - 1 > 0 Then

oa(i, 0) = Val(oa(i, 0)) - 1

End If

Exit For

Case "N"

If Val(oa(i, 1)) + 1 < a(1) Then

oa(i, 1) = Val(oa(i, 1)) + 1

End If

Exit For

Case "S"

If Val(oa(i, 1)) - 1 > 0 Then

oa(i, 1) = Val(oa(i, 1)) - 1

End If

Exit For

End Select

ElseIf l = n / 2 Then

Select Case oa(i, 2)

Case "E"

If Val(oa(i, 0)) + 1 > a(0) Then GoTo 2

oa(i, 0) = Val(oa(i, 0)) + 1

Case "W"

If Val(oa(i, 0)) - 1 < 0 Then GoTo 2

oa(i, 0) = Val(oa(i, 0)) - 1

Case "N"

If Val(oa(i, 1)) + 1 > a(1) Then GoTo 2

oa(i, 1) = Val(oa(i, 1)) + 1

Case "S"

If Val(oa(i, 1)) - 1 < 0 Then GoTo 2

oa(i, 1) = Val(oa(i, 1)) - 1

Case Else

2: sw.WriteLine(oa(i, 0) & " " & oa(i, 1) & " " & oa(i, 2) & " LOSE")

oa(i, 3) = "LOSE" : GoTo 1

End Select

End If

Next

End Select

Next

sw.WriteLine(oa(i, 0) & " " & oa(i, 1) & " " & oa(i, 2))

1:

Next

sw.Flush() : sw.Close() : End

End Sub

End Class