Nama : Fuad Zauqi Nur

NIM : 1301164392

Kelas : IF-40-03

I. DESKRIPSI MASALAH

Bangunlah sebuah program *Q-learning* untuk menemukan *optimum policy* sehingga *Agent* yang berada di posisi *Start* (1,1) mampu menemukan *Goal* yang berada di posisi (15,15) dengan mendapatkan ***Total Reward*** maksimum pada *grid world* Gambar di bawah ini. Data pada gambar tersebut dapat dilihat di file DataTugas3ML2019.txt. Pada kasus ini, *Agent* hanya bisa melakukan empat aksi: N, E, S, dan W yang secara berurutan menyatakan *North* (ke atas), *East* (ke kanan), *South* (ke bawah), dan *West* (ke kiri). Anda boleh menggunakan skema apapun dalam mengimplementasikan sebuah *episode*.

II. PENYELESAIAN MASALAH

Berikut adalah penyelesaian masalah yang dilakukan dengan menggunakan Jupyter Notebook :

Menginport library yang diperlukan seperti numpy, pandas dan random.

In [530]:
```
import numpy as np
import pandas as pd
import random
```

Menginisiasi Tabel R

In [531]:
```
TR = pd.read_csv('DataTugas3ML2019.txt', sep="\t", header=None)
```

Menginisiasi Tabel Q

In [532]:
```
nol = np.zeros(225)
Q = {'N': nol, 'E': nol, 'S': nol, 'W': nol}
TQ = pd.DataFrame(data=Q)
```

Menginisiasi gamma dan alpha dengan 1 dan jumlah episode 100

In [533]:
```
gamma= 1
jumlahepisode= 100
alpha = 1
```

Nama : Fuad Zauqi Nur

NIM : 1301164392

Kelas : IF-40-03

Fungsi Move adalah fungsi untuk mengubah kordinat x dan y sesuai dengan arah gerakan sekaligus agar arah gerakan agen valid, yang artinya tetap di kordinat 15x15.

In [534]:
```
def Move(x, y):
    act = ''
    valid = False

    while valid == False:
        newx = x
        newy = y
        action = random.randint(1, 4)
        if (action == 1): #north
            act = 'N'
            newy = y -1
            if (newy  <= 14 and newy >= 0):
                valid = True
        elif (action == 2): #east
            newx = x + 1
            act = 'E'
            if (newx  <= 14 and newx >= 0):
                valid = True
        elif (action == 3): #west
            act = 'W'
            newx = x -1
            if (newx  <= 14 and newx >= 0):
                valid = True
        elif (action == 4): #south
            act ='S'
            newy = y + 1
            if (newy <= 14 and newy >= 0):
                valid = True
    result = [newx, newy, act]
    return result
```

Main program dari q-learning, pada bagian ini LR adalah list reward yang berfungsi untuk mencari max reward, statex dan statey adalah kordinat x dan y, dan koordinat dimulai pada 0,14 dan diakhiri pada 14,0.

In [535]:
```
LR = []
path =[]
print('Learning...')
for i in range(0, jumlahepisode):
    statex = 0
    statey = 14
    reward = 0
    while statex != 14 or statey != 0:
        action = Move(statex, statey)
        TQ[action[2]][statex + statey] = TQ[action[2]][statex + statey] + (alpha * (TR[action[0]][action[1]] + (gamma * max
        reward = reward + TR[statex][statey]
        statex = action[0]
        statey = action[1]
    reward = reward + TR[14][0]
    path.append(reward)
    LR.append(reward)
print('Learning Successful!')
print('Total Reward Maximum is :', max(LR))
for j in range(0, jumlahepisode):
    if (path[j]==max(LR)):
        print('The Optimal Path is at Episode :',j)
```

```
Learning...
Learning Successful!
Total Reward Maximum is : 312
The Optimal Path is at Episode : 39
```