

U18CO018
Shubham Shekhaliya
Sub: AIML
Lab: Assignment-7
Topic: Constrains Satisfactory Problem

Consider the problem of solving crossword puzzles:

fitting words into a rectangular grid. The grid, which is given as part of the problem, specifies which squares are blank and which are shaded. For each word starting square you have a list of words that can be fitted (vertical and/or across).

○ Solution

```
import copy

row = -1
col = -1
initial = []
dist = {}
array = []
words = []

def put_word(i,j,a):
    word = words[i][j]
    word_length = len(word)
    pos = int(array[i][1])
    ii = int(pos/col)
    jj = pos%col
    dir = array[i][0]
    if dir == 'A':          # put horizontal
        if (jj + word_length > col) :
            return [False, None]

        for iii in range(word_length):
            ch = a[ii][jj+iii]
            if((ch == '#') or (ch != '.' and ch!=word[iii])):
                return [False, None]
            a[ii][jj + iii] = word[iii];
    else : #put vertical
        if(ii + word_length > row):
            return [False, None]
```

```

        for iii in range (word_length) :
            ch = a[ii + iii][jj]
            if((ch == '#') or (ch != '.' and ch!=word[iii])):
                return [False, None]
            a[ii + iii][jj] = word[iii];
        return [True, a]

def solve(i,puzzle):
    if(i== len(words)):
        for i in puzzle:
            for j in i:
                if j == '.':
                    return [False, None]
        return [True, puzzle]

    for jj in range(len(words[i])) :
        [a, b] = put_word(i,jj,copy.deepcopy(puzzle))
        if a == False :
            continue
        [c, d] = solve(i+1,copy.deepcopy(b))
        if c == True :
            return [c,d]

    return [False,None]

if __name__ == "__main__":
    solved = False

    maxx = 0
    line = input().split(' ')
    row = int(line[0])
    col = int(line[1])

    for i in range(row) :
        temp = []
        ss = input()
        for j in range(col) :
            ch = ss[j]
            if (ch >= '1' and ch <= '9') :
                temp.append('.')
                dist[ch]= i*col + j
                maxx += 1
            else :
                temp.append(ch)
        initial.append(temp)

    for i in range (maxx):
        line = input().split(' ')

```

```

ch = line[0]
dir = line[1]
num = int(line[2])
list = []
for j in range (num):
    temp = line[j + 3].replace(" ", "")
    list.append(temp)

tlist = [dir, dist.get(ch)]
array.append(tlist)
words.append(list)

for i in range(len(words[0])):
    [a, b] = put_word(0, i, copy.deepcopy(initial))
    if a == False :
        continue

    [c, d] = solve(1, copy.deepcopy(b))

    if c == True :
        for ii in d :
            solved = True
            print(ii)
            break

if solved == False :
    print('Solution Does not exist')

```

Example

1st

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

D:\xampp\htdocs\Assignments>8 A 5 hoses laser sails sheet steer
'8' is not recognized as an internal or external command,
operable program or batch file.

D:\xampp\htdocs\Assignments>python -u "d:\xampp\htdocs\Assignments\AIML\Assign-7.py"
6 5
1.2.3
##.#.
#4.5.
6#7..
8....
.##.#
1 A 5 hoses laser sails sheet steer
2 V 5 hoses laser sails sheet steer
3 V 5 hoses laser sails sheet steer
4 A 5 heel hike keel knot line
5 V 5 heel hike keel knot line
6 V 5 aft ale eel lee tie
7 A 5 aft ale eel lee tie
8 A 5 hoses laser sails sheet steer
['h', 'o', 's', 'e', 's']
['#', '#', 'a', '#', 't']
['#', 'h', 'i', 'k', 'e']
['a', '#', 'l', 'e', 'e']
['l', 'a', 's', 'e', 'r']
['e', '#', '#', 'l', '#']

```

2nd

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

D:\xampp\htdocs\Assignments>python -u "d:\xampp\htdocs\Assignments\AIML\Assign-7.py"
3 3
12.
#.#
#3.
1 A 3 MNP PQR XYZ
2 V 4 SMG HBD MGDS LK
3 A 2 OU DE
Solution Does not exist
```

3rd

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

D:\xampp\htdocs\Assignments>python -u "d:\xampp\htdocs\Assignments\AIML\Assign-7.py"
2 3
1.2
##.
1 A 3 ABC DEF GHI
2 V 2 CED CE
['A', 'B', 'C']
['#', '#', 'E']
```