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)

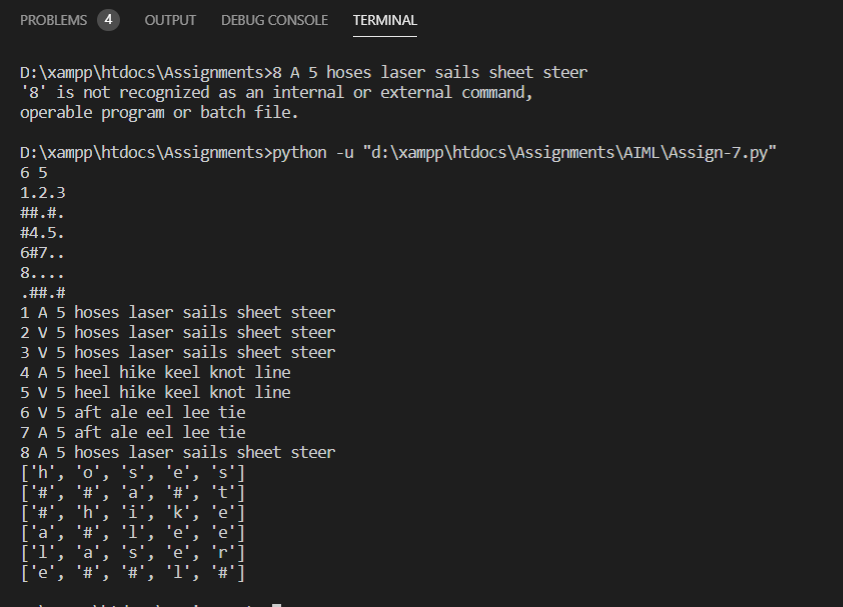
            break

    if solved == False :

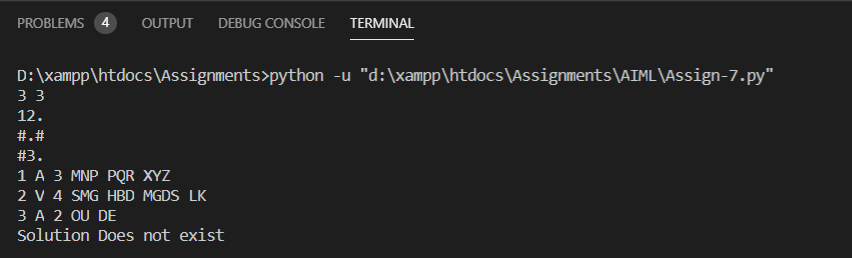
        print('Solution Does not exist')

# Example

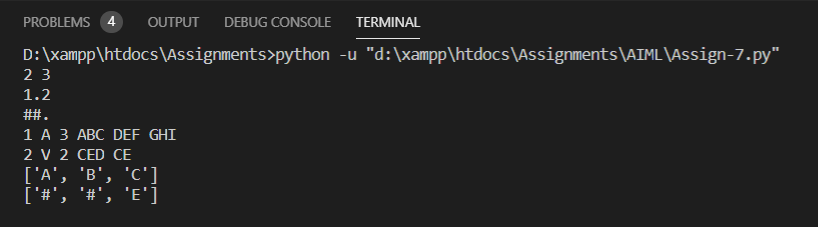
**1st**

****

**2nd**

****

**3rd**

****