```
def horizontal(ans,puzzle):
    for x in range (0,len(puzzle)):
        temp = "".join(map(str,puzzle[x]))
        if temp.find(ans) > -1:
            print("Right to Left: ("+str(temp.find(ans)+1)+","+str(x+1)+")")
            return True
        if temp.find(ans[::-1]) > -1:
            print("Left to Right (Backwards): ("+str(temp.find(ans[::-1])+1)+","+str(x+1)+")")
            return True
    return False


def vertical(ans, puzzle):
    for x in range(0, len(puzzle[0])):
        temp = []
        for j in range(0, len(puzzle)):
            temp.append(puzzle[j][x])
        temp = "".join(map(str,temp))
        if temp.find(ans) > -1:
            print("Top to Bottom: ("+str(x+1)+","+str(temp.find(ans)+1)+")")
            return True
        if temp.find(ans[::-1]) > -1:
            print("Bottom to Top: ("+str(x+1)+","+str(temp.find(ans[::-1])+len(ans))+")")
            return True
    return False

def diagonal(ans,puzzle):
    #mid-bottom left
    temp = []
    for x in range(0,len(puzzle)):
        temp = []
        temp2 = x
        for j in range(0,len(puzzle)-x):
            temp.append(puzzle[temp2][j])
            temp2 += 1
        #print(temp)
        temp = "".join(map(str,temp))
        if temp.find(ans) > -1:
            print(temp)
            print("Diagonal to the Bottom Right starting at: ("+str(temp.find(ans))+","+str(j)+")")
            #get y cord of letter and more +1 in bot directions for however long the word is
            return True
        if temp.find(ans[::-1]) > -1:
            #print("Bottom to Top Left: ("+str(x+1)+","+str(temp.find(ans[::-1])+len(ans))+")")
            return True
        """Fix coords"""

def diagonal2(ans, puzzle):
    #mid-bottom right
    for x in range(0, len(puzzle)):
        temp = []
        j = len(puzzle)-1
        temp2 = x
        while temp2 < len(puzzle):
            #print(str(x)+","+str(temp2))
            temp.append(puzzle[temp2][j])
            j-=1
            temp2+=1
        #print(temp)
        temp = "".join(map(str,temp))
        if temp.find(ans) > -1:
            print("Diagonal  the Bottom Right: ("+str(x+1)+","+str(temp.find(ans)+1)+")")
            return True
        if temp.find(ans[::-1]) > -1:
            print("Bottom to Top Left: ("+str(x+1)+","+str(temp.find(ans[::-1])+len(ans))+")")
            return True
    return False

def diagonal3(ans, puzzle):
    #mid-top left
```

```
    for j in range (len(puzzle)-1,0,-1):
        temp = []
        x = 0
        temp2 = j
        while x != j+1:
            #print(str(x)+","+str(temp2))
            temp.append(puzzle[x][temp2])
            x+=1
            temp2 -=1
        temp = "".join(map(str,temp))
        if temp.find(ans) > -1:
            print("Diagonal to Bottom Right: ("+str(x+1)+","+str(temp.find(ans)+1)+")")
            return True
        if temp.find(ans[::-1]) > -1:
            print("Bottom to Top Left: ("+str(x+1)+","+str(temp.find(ans[::-1])+len(ans))+")")
            return True
    return False

def diagonal4(ans,puzzle):
    #mid-top right
    temp = []
    for x in range(0,len(puzzle)):
        temp = []
        j = 0
        temp2 = x
        for j in range(0,len(puzzle)-x):
            temp.append(puzzle[j][temp2])
            temp2 += 1
        #print(temp)
        temp = "".join(map(str,temp))
        if temp.find(ans) > -1:
            print("here")
            print("Diagonal to the Bottom Right: ("+str(j+1)+","+str(temp2+1)+")")
            return True
        if temp.find(ans[::-1]) > -1:
            print("Diagonal to the Top Left: ("+str(j+1)+","+str(temp2+1)+")")
            return True

def solve(ans,puzzle):
    Found = False
    while not Found:
        """if horizontal(ans,puzzle):
            break
        if vertical(ans,puzzle):
            break"""
        if diagonal(ans,puzzle):
            break
        else:
            print("Not found in puzzle")
            break
        """if diagonal2(ans,puzzle):
            break
        if diagonal3(ans,puzzle):
            break
        if diagonal4(ans,puzzle):
            break"""


l1 = "csmfsnow"
l2 = "ocbrcsmh"
l3 = "aaqoobvk"
l4 = "treslfei"
l5 = "wfotdqsz"
l6 = "lwkyqice"
l7 = "vqwigloo"
l8 = "hwintern"
puzzle = [1,2,3,4,5,6,7,8]
puzzle[0] = list(l1)
puzzle[1] = list(l2)
puzzle[2] = list(l3)
```

```
puzzle[3] = list(l4)
puzzle[4] = list(l5)
puzzle[5] = list(l6)
puzzle[6] = list(l7)
puzzle[7] = list(l8)

def print_puzzle(puz):
    for x in range (0,len(puz)):
        print(puz[x])
print_puzzle(puzzle)


while True:
    word = input("What word are you looking for: ")
    solve(word,puzzle)
```