**Answer 1)**

1. **Algorithm given below**

def search(grid, word):

for i in range(len(grid)): # iterate through rows of grid

for j in range(len(grid[0])): # iterate through cloumns of grid

if grid[i][j] == word[0]: # if first character matches

word\_Index = 1 # point to the next index of the word

for x in range(i+1, len(grid)): # search vertically

if word[word\_Index] != grid[x][j]: # if word character do not match with grid character then break

break

word\_Index += 1 # increament the word index

if word\_Index == len(word): # if word index overflows, means whole word matched

print("starting coordinates: ", i, j,

" and ending coordinates: ", x, j) # print the coordinates

break

word\_Index = 1

for y in range(j+1, len(grid[0])): # search horizontally

if word[word\_Index] != grid[i][y]: # if word character do not match with grid character then break

break

word\_Index += 1 # increament the word index

if word\_Index == len(word): # if word index overflows, means whole word matched

print("starting coordinates: ", i, j,

" and ending coordinates: ", i, y) # print the coordinates

break

grid = ["BIROKNMO", "TABNBICT", "GMIRIAOO", "MKTNGNIG",

"UCCOSNAR", "NAONBCAS", "GLIITCOI", "OBNUGRAC"]

search(grid, 'BITCOIN')

**Answer b)**

As we are matching each character in the grid n\*n, due to the iteration we can conclude that complexity for sure will be O(n\*n)

Taking the length of the word into account by looking at the grid the algorithm will give off O(L) complexity in the grid due to the length of the word regardless of it being in horizontal or vertical direction. Hence the overall complexity of the algorithm will be **O(n \* n \* L).**

**Answer c)**

Upon running this:

grid = ["BIROKNMO", "TABNBICT", "GMIRIAOO", "MKTNGNIG",

"UCCOSNAR", "NAONBCAS", "GLIITCOI", "OBNUGRAC"]

search(grid, 'BITCOIN')

The program outputs:

starting coordinates: 1 2 and ending coordinates: 7 2

Which does correspond with the input grid in the program which is executed with the help of search algorithm.