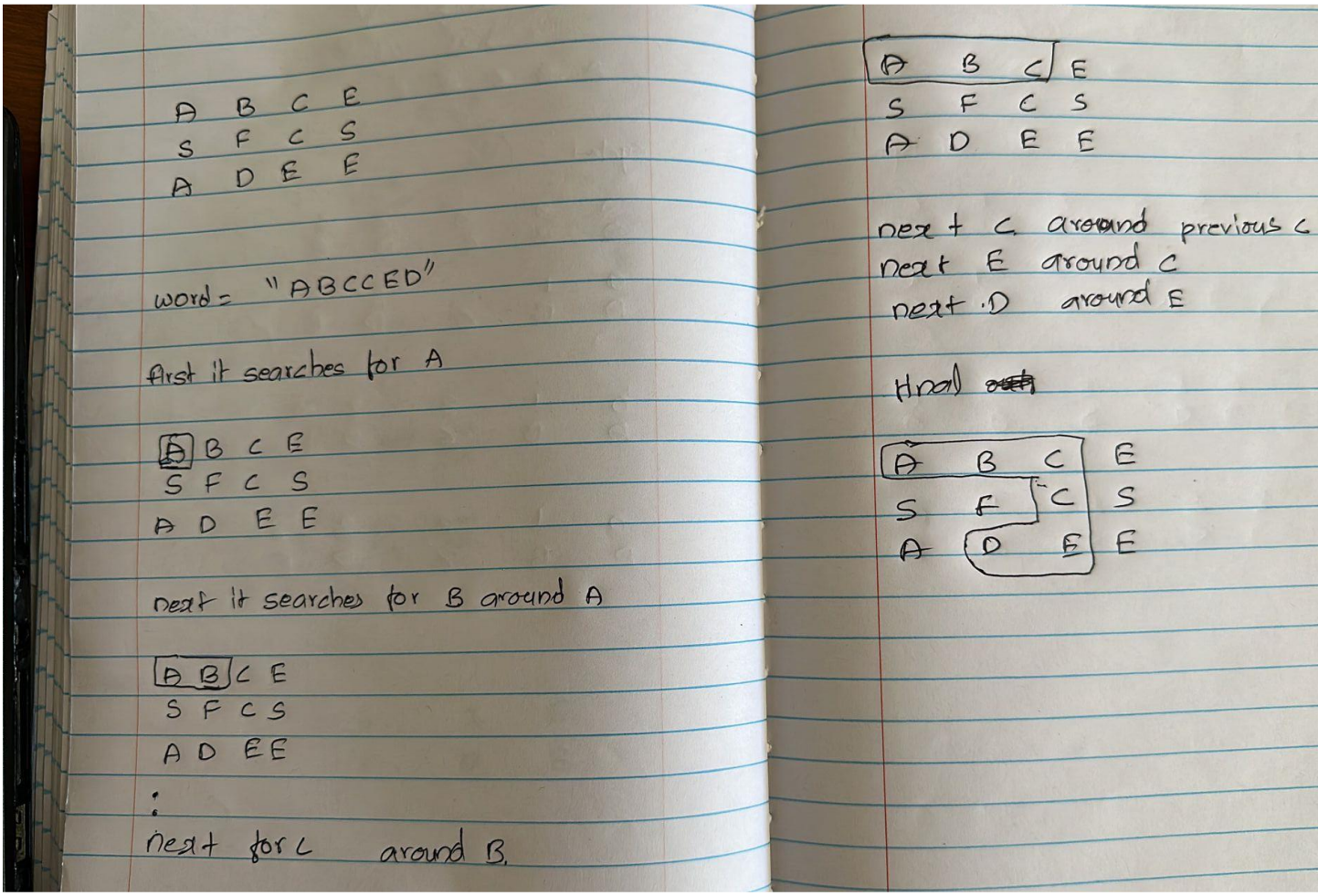
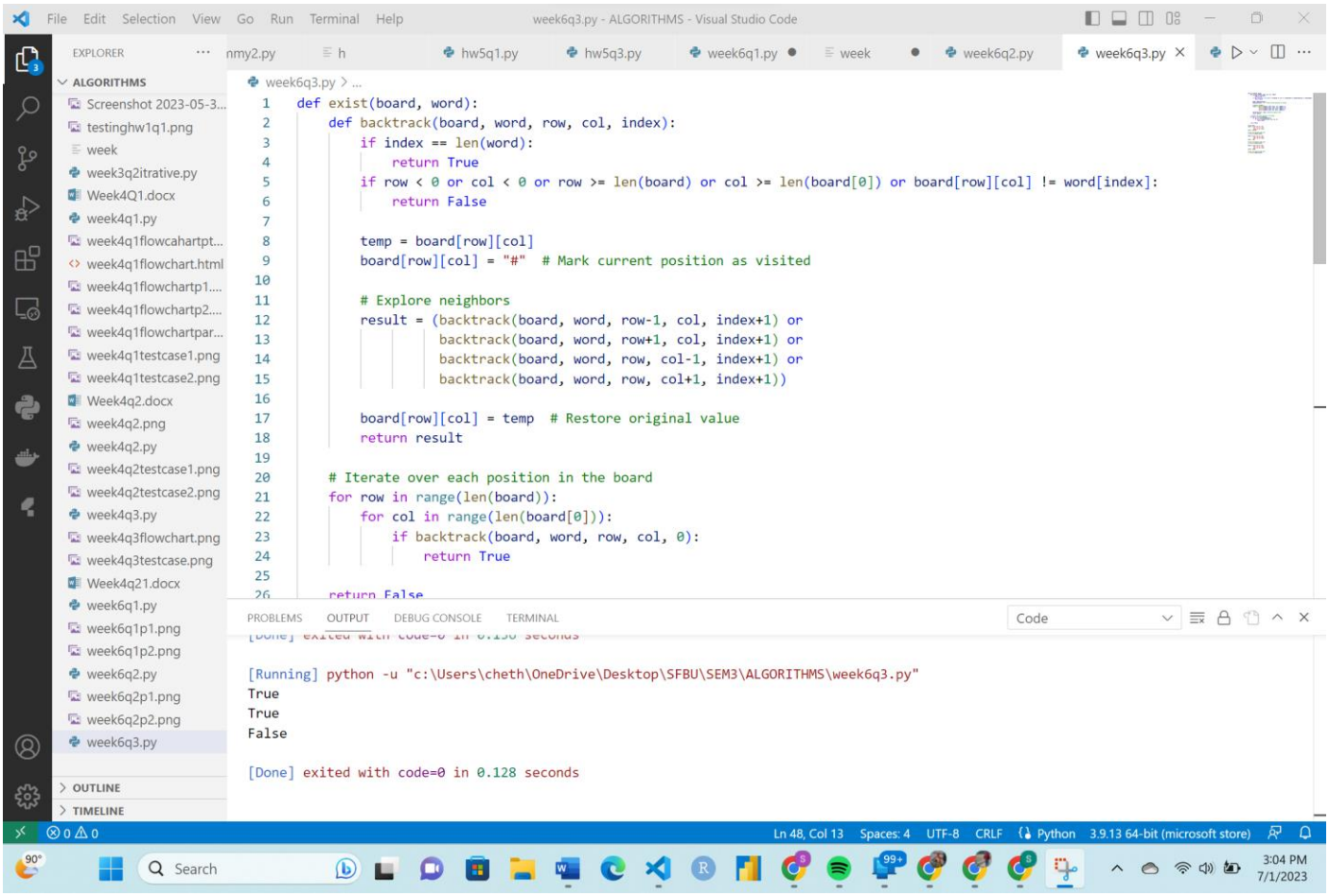
Week6q3

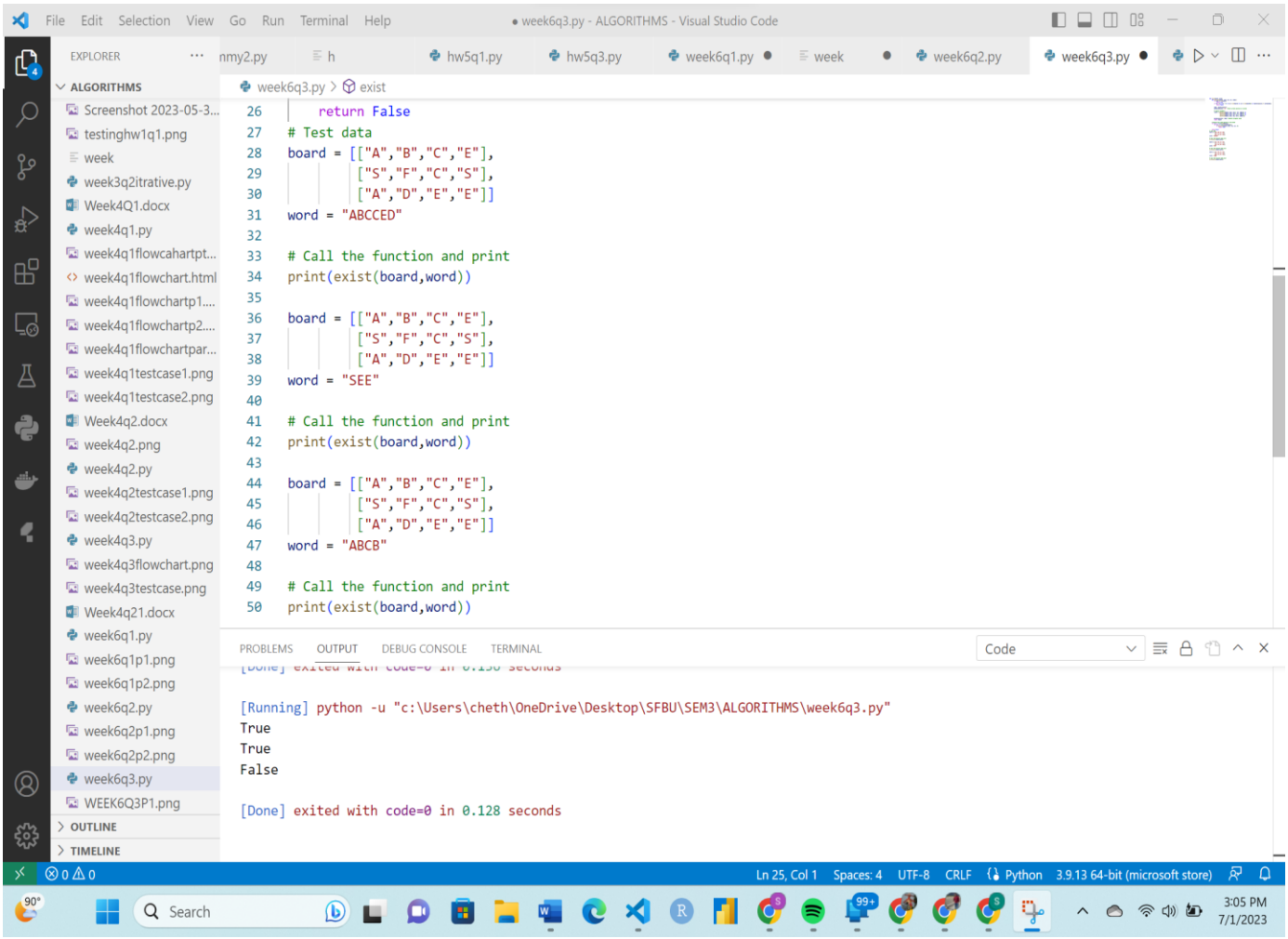

A B C E
S F C S
A D E E
word = "ABCCED"
first it searches for A
A B C E
S F C S
A D E E
next it searches for B around A
A B C E
S F C S
A D E E
:
next for C around B.
A B C E
S F C S
A D E E
next C around previous C
next E around C
next D around E
Final
A B C E
S F C S
A D E E

Chatgpt code

```
def exist(board, word):
    def backtrack(board, word, row, col, index):
        if index == len(word):
            return True
        if row < 0 or col < 0 or row >= len(board) or col >= len(board[0]) or
board[row][col] != word[index]:
            return False

        temp = board[row][col]
        board[row][col] = "#"  # Mark current position as visited

        # Explore neighbors
        result = (backtrack(board, word, row-1, col, index+1) or
                  backtrack(board, word, row+1, col, index+1) or
                  backtrack(board, word, row, col-1, index+1) or
                  backtrack(board, word, row, col+1, index+1))

        board[row][col] = temp  # Restore original value
        return result

    # Iterate over each position in the board
    for row in range(len(board)):
        for col in range(len(board[0])):
            if backtrack(board, word, row, col, 0):
                return True

    return False

# Test data
board = [["A","B","C","E"],
         ["S","F","C","S"],
         ["A","D","E","E"]]
word = "ABCCED"

# Call the function and print
print(exist(board,word))
```

test case


week6q3.py - ALGORITHMS - Visual Studio Code
EXPLORER
ALGORITHMS
Screenshot 2023-05-3...
testinghw1q1.png
week
week3q2itrative.py
Week4Q1.docx
week4q1.py
week4q1flowcahartpt...
week4q1flowchart.html
week4q1flowchartp1....
week4q1flowchartp2....
week4q1flowchartpar...
week4q1testcase1.png
week4q1testcase2.png
Week4q2.docx
week4q2.png
week4q2.py
week4q2testcase1.png
week4q2testcase2.png
week4q3.py
week4q3flowchart.png
week4q3testcase.png
Week4q21.docx
week6q1.py
week6q1p1.png
week6q1p2.png
week6q2.py
week6q2p1.png
week6q2p2.png
week6q3.py
OUTLINE
TIMELINE
hw5q1.py
hw5q3.py
week6q1.py
week
week6q2.py
week6q3.py
week6q3.py > ...
def exist(board, word):
def backtrack(board, word, row, col, index):
if index == len(word):
return True
if row < 0 or col < 0 or row >= len(board) or col >= len(board[0]) or board[row][col] != word[index]:
return False
temp = board[row][col]
board[row][col] = "#" # Mark current position as visited
# Explore neighbors
result = (backtrack(board, word, row-1, col, index+1) or
backtrack(board, word, row+1, col, index+1) or
backtrack(board, word, row, col-1, index+1) or
backtrack(board, word, row, col+1, index+1))
board[row][col] = temp # Restore original value
return result
# Iterate over each position in the board
for row in range(len(board)):
for col in range(len(board[0])):
if backtrack(board, word, row, col, 0):
return True
return False
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\week6q3.py"
True
True
False
[Done] exited with code=0 in 0.128 seconds
Ln 48, Col 13
Spaces: 4
UTF-8
CRLF
Python
3.9.13 64-bit (microsoft store)
Search
3:04 PM
7/1/2023


File Edit Selection View Go Run Terminal Help
• week6q3.py - ALGORITHMS - Visual Studio Code
EXPLORER
ALGORITHMS
Screenshot 2023-05-3...
testinghw1q1.png
week
week3q2itrative.py
Week4Q1.docx
week4q1.py
week4q1flowcahartpt...
week4q1flowchart.html
week4q1flowchartp1....
week4q1flowchartp2....
week4q1flowchartpar...
week4q1testcase1.png
week4q1testcase2.png
Week4q2.docx
week4q2.png
week4q2.py
week4q2testcase1.png
week4q2testcase2.png
week4q3.py
week4q3flowchart.png
week4q3testcase.png
Week4q21.docx
week6q1.py
week6q1p1.png
week6q1p2.png
week6q2.py
week6q2p1.png
week6q2p2.png
week6q3.py
WEEK6Q3P1.png
OUTLINE
TIMELINE
nmy2.py
h
hw5q1.py
hw5q3.py
week6q1.py
week
week6q2.py
week6q3.py
week6q3.py > exist
return False
# Test data
board = [["A","B","C","E"],
["S","F","C","S"],
["A","D","E","E"]]
word = "ABCCED"
# Call the function and print
print(exist(board,word))
board = [["A","B","C","E"],
["S","F","C","S"],
["A","D","E","E"]]
word = "SEE"
# Call the function and print
print(exist(board,word))
board = [["A","B","C","E"],
["S","F","C","S"],
["A","D","E","E"]]
word = "ABCB"
# Call the function and print
print(exist(board,word))
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\week6q3.py"
True
True
False
[Done] exited with code=0 in 0.128 seconds
Ln 25, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.9.13 64-bit (microsoft store)
Search
3:05 PM
7/1/2023