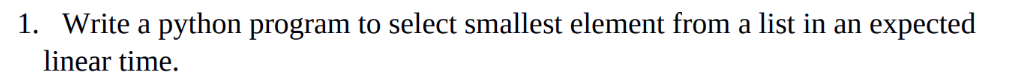
**Name: Areen Chakraborty**

**220905496**

**CSE A Roll No. 53**

**Batch A2**

**Web Programming Lab – Week 4 Assignment**



Code:

def get\_min(arr):

    min\_val = arr[0]

    for num in arr:

        if num < min\_val:

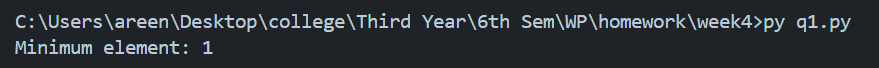
            min\_val = num

    return min\_val

arr = [3, 1, 4, 1, 5, 9, 2, 6, 5]

print("Minimum element:", get\_min(arr))

Output:





Code:

def bubble\_sort(arr):

    n = len(arr)

    for i in range(n):

        swapped = False

        for j in range(n - i - 1):

            if arr[j] > arr[j + 1]:

                arr[j], arr[j + 1] = arr[j + 1], arr[j]

                swapped = True

        if not swapped:

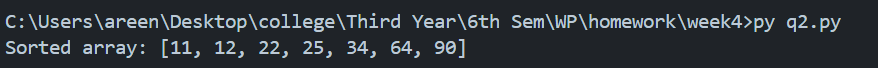
            break

arr = [64, 34, 25, 12, 22, 11, 90]

bubble\_sort(arr)

print("Sorted array:", arr)

Output:





Code:

def matrix\_multiply(A, B):

    rows\_A, cols\_A = len(A), len(A[0])

    rows\_B, cols\_B = len(B), len(B[0])

    if cols\_A != rows\_B:

        raise ValueError("Matrix dimensions do not match for multiplication")

    result = [[0] \* cols\_B for \_ in range(rows\_A)]

    for i in range(rows\_A):

        for j in range(cols\_B):

            for k in range(cols\_A):

                result[i][j] += A[i][k] \* B[k][j]

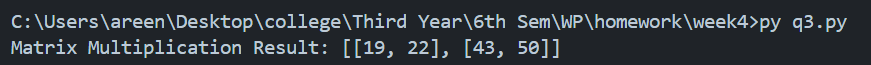
    return result

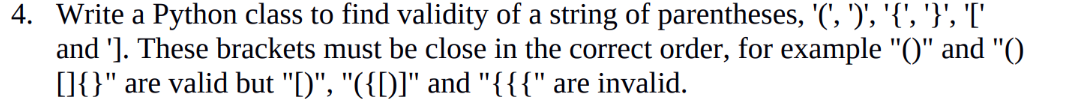
A = [[1, 2], [3, 4]]

B = [[5, 6], [7, 8]]

print("Matrix Multiplication Result:", matrix\_multiply(A, B))

Output:





Code:

class ValidParentheses:

    def is\_valid(self, s: str) -> bool:

        stack = []

        mapping = {')': '(', '}': '{', ']': '['}

        for char in s:

            if char in mapping:

                top\_element = stack.pop() if stack else '#'

                if mapping[char] != top\_element:

                    return False

            else:

                stack.append(char)

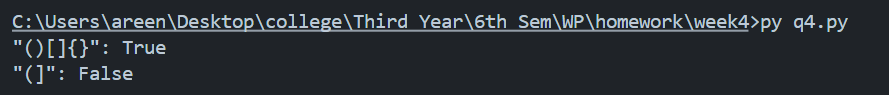
        return not stack

vp = ValidParentheses()

print(f'"()[]{{}}": {vp.is\_valid("()[]{}")}')

print(f'"(]": {vp.is\_valid("(]")}')

Output:





Code:

class ReverseString:

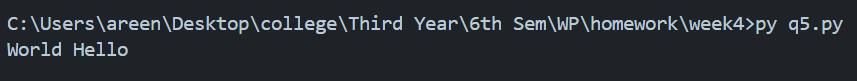
    def reverse\_words(self, s: str) -> str:

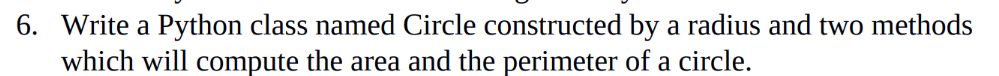
        return ' '.join(s.split()[::-1])

rs = ReverseString()

print(rs.reverse\_words("Hello World"))

Output:





Code:

import math

class Circle:

    def \_\_init\_\_(self, radius):

        self.radius = radius

    def area(self):

        return math.pi \* self.radius \*\* 2

    def perimeter(self):

        return 2 \* math.pi \* self.radius

c = Circle(5)

print(f"Area: {c.area():.2f}")

print(f"Perimeter: {c.perimeter():.2f}")

Output:

