

WPL LAB 4 HW

Name: Abhinav Singh Bhagtana

Section: A

Roll no: 29

Reg no: 230905225

Q1. Write a python program to select smallest element from a list in an expected linear time.

```
inp = [int(input(f'enter number {i+1} ')) for i in range(5)]  
print(min(inp))
```

OUTPUT:

```
$ python3 q1.py  
enter number 1 15  
enter number 2 12  
enter number 3 11  
enter number 4 44  
enter number 5 1  
1
```

Q2. Write a python program to implement bubble sort.

```
def optimized_bubble_sort(arr):  
    n = len(arr)  
    for i in range(n):  
        swapped = False  
        for j in range(0, 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
    return arr

n = int(input('how many numbers '))
numbers = [int(input(f'enter {i+1} num ')) for i in range(n)]
sorted_numbers = optimized_bubble_sort(numbers)
print(f"Sorted list: {sorted_numbers}")
```

OUTPUT:

```
$ python3 q2.py
```

```
how many numbers 6
```

```
enter 1 num 15
```

```
enter 2 num 12
```

```
enter 3 num 19
```

```
enter 4 num 48
```

```
enter 5 num 55
```

```
enter 6 num 100
```

```
Sorted list: [12, 15, 19, 48, 55, 100]
```

Q3. Write a python program to multiply two matrices

```
rows1 = int(input("Enter number of rows for Matrix 1: "))
```

```
matrix1 = [list(map(int, input(f"Row {i+1}: ").split())) for i in range(rows1)]
```

```
rows2 = int(input("Enter number of rows for Matrix 2: "))
```

```
matrix2 = [list(map(int, input(f"Row {i+1}: ").split())) for i in range(rows2)]
```

```
cols1 = len(matrix1[0])
```

```
if cols1 != rows2:  
    print(f"Error: Columns of Matrix 1 ({cols1}) must match Rows of Matrix 2 ({rows2})")  
    exit()
```

```
cols2 = len(matrix2[0])  
sol = [[0 for _ in range(cols2)] for _ in range(rows1)]
```

```
for i in range(rows1):  
    for j in range(cols2):  
        for k in range(rows2):  
            sol[i][j] += matrix1[i][k] * matrix2[k][j]
```

```
print('\nResult of matrix multiplication:')
```

```
for row in sol:
```

```
    print(row)
```

OUTPUT:

Enter number of rows for Matrix 1: 3

Row 1: 1 2 3

Row 2: 4 5 6

Row 3: 7 8 9

Enter number of rows for Matrix 2: 3

Row 1: 9 8 7

Row 2: 6 5 4

Row 3: 3 2 1

Result of matrix multiplication:

[30, 24, 18]

[84, 69, 54]

[138, 114, 90]

Q4. Write a Python class to find validity of a string of parentheses, '(', ')', '{', '}', '[' and ']'. These brackets must be close in the correct order, for example "()" and "()[{}]" are valid but "[]", "{[]}" and "{{{" are invalid.

```
string = input()
```

```
stack = []
```

```
for c in string:
```

```
    if c == '(' or c == '[' or c == '{':
```

```
        stack.append(c)
```

```
    elif c == ')':
```

```
        seek = stack[-1]
```

```
        if seek == '(':
```

```
            stack.pop()
```

```
            continue
```

```
        else:
```

```
            print('invalid')
```

```
            exit()
```

```
    if c == ']':
```

```
        seek = stack[-1]
```

```
        if seek == '[':
```

```
            stack.pop()
```

```
continue
else:
print('invalid')
exit()
if c == '}':
seek = stack[-1]
if seek == '{':
stack.pop()
continue
else:
print('invalid')
exit()
print('valid')
```

OUTPUT:

```
()(())
```

```
valid
```

Q5. Write a Python class to reverse a string word by word.

```
class reverse:
def rev(self,string):
reversed_text = " ".join(string.split()[::-1])
print(reversed_text)
obj = reverse()
string = input('enter a sentence ')
obj.rev(string)
```

OUTPUT:

```
$ python3 q5.py
```

enter a sentence HI HELLO MY NAME IS

IS NAME MY HELLO HI

Q6. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

```
import math as m
```

```
class Circle:
```

```
    def init(self,radius):
```

```
        self.r = radius
```

```
    def getArea(self):
```

```
        return m.pi * self.r ** 2
```

```
    def getCircumference(self):
```

```
        return 2 * m.pi * self.r radius = int(input('enter radius '))
```

```
obj = Circle(radius)
```

```
print('area of circle',obj.getArea(),'circumference',obj.getCircumference())
```

OUTPUT:

enter radius 5

area of circle 78.53981633974483 circumference 31.41592653589793