

Name: Sapthami Upadhya

Section: CSE A

Roll No.: 15

Reg. No.: 230905090

Week 4 Additional

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

```
In [6]: li = []
n = int(input("Enter no. of elements: "))
print("Enter the list elements:")
for i in range(n):
    li.append(int(input()))
print(li)
print("Smallest element:", min(li))
```

Enter the list elements:

[2, 0, -4, 7, 4]

Smallest element: -4

2. Write a python program to implement bubble sort.

```
In [8]: li = []
n = int(input("Enter no. of elements: "))
print("Enter the list elements:")
for i in range(n):
    li.append(int(input()))
print(li)
for i in range(n-1):
    for j in range(n-1-i):
        if li[j] > li[j+1]:
            li[j], li[j+1] = li[j+1], li[j]
print("Sorted list:", li)
```

Enter the list elements:

[2, 0, -4, 7, 4]

Sorted list: [-4, 0, 2, 4, 7]

3. Write a python program to multiply two matrices

```
In [12]: matA, matB, matC = [], [], []
m1 = int(input("Enter no. of rows for first matrix:"))
n1 = int(input("Enter no. of cols for first matrix:"))
print("Enter first matrix:")
for i in range(m1):
    row = []
    for j in range(n1):
        row.append(int(input()))
    matA.append(row)
print(matA)

m2 = int(input("Enter no. of rows for second matrix:"))
```

```

n2 = int(input("Enter no. of cols for second matrix:"))
if n1 != m2:
    print("#col of first matrix should be same as #row of second matrix")
    exit()

print("Enter the second matrix:")
for i in range(m2):
    row = []
    for j in range(n2):
        row.append(int(input()))
    matB.append(row)
print(matB)

for i in range(m1):
    row = []
    for j in range(n2):
        row.append(0)
    matC.append(row)

for i in range(m1):
    for j in range(n2):
        matC[i][j] = 0
        for k in range(n1):
            matC[i][j] += matA[i][k] * matB[k][j]
print("Product:", matC)

```

Enter first matrix:

[[1, 2, 3], [4, 5, 6]]

Enter the second matrix:

[[7, 8], [9, 10], [11, 12]]

Product: [[58, 64], [139, 154]]

4. 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.

```

In [15]: class ValidatePar:
    def isValid(self, s):
        stack = []
        pairs = {
            ')': '(',
            '}': '{',
            ']': '['
        }
        for ch in s:
            if ch in ['(', '{', '[']:
                stack.append(ch)
            elif ch in [')', '}', ']']:
                if len(stack) == 0: #closing bracket with no prior opening
                    return False
                top = stack.pop()
                if top != pairs[ch]:
                    return False
            if len(stack) == 0:
                return True
            else:
                return False

obj = ValidatePar()

```

```

str = input("Enter a test case:")
print(str, ":", obj.isValid(str))

str = input("Enter a test case:")
print(str, ":", obj.isValid(str))

str = input("Enter a test case:")
print(str, ":", obj.isValid(str))

```

```

()[]{} : True
{[]} : False
{{{ : False

```

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

```

In [18]: class ReverseWord:
    def revwords(self, s):
        words = s.split()
        stack = []
        for w in words:
            stack.append(w)

        result = ""
        while len(stack) > 0:
            result += stack.pop()
            if len(stack) > 0:
                result += " "
        return result
    obj = ReverseWord()
    str = input("Enter a string:")
    print("Original:", str)
    print("Reversed:", obj.revwords(str))

```

```

Original: I love Python very much
Reversed: much very Python love I

```

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

```

In [20]: 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

obj = Circle(3)
print("Radius:", obj.radius, "| Perimeter:", obj.perimeter(), "| Area:",

```

```

Radius: 3 | Perimeter: 18.84955592153876 | Area: 28.274333882308138

```