

UNIVERSITY OF DELHI

COLLEGE OF VOCATIONAL STUDIES

BSC (HONS) COMPUTER SCIENCE

SEMESTER - 3

PROGRAMMING IN PYTHON

ANUJ KUMAR SEN

2K21/CS/15

Question 1 :

Write a function that takes the lengths of three sides : side1,side2 and side3 of the triangle as the input from the user using the input function and return the area and perimeter of the triangle as a tuple. Also, assert that the sum of the length of any two sides is greater than the third side.

Solution 1 :

```
# Created By : ANUJ KUMAR SEN
import math

def calculate(side1, side2, side3):
    perimeter = 0
    area = 0
    # Checking condition : Sum of the length of any two sides is greater than the
    third side
    if side1 + side2 > side3 and side2 + side3 > side1 and side1 + side3 > side2:
        print("Given sides form a triangle.")
        perimeter = side1 + side2 + side3
        s = perimeter/2
        area = math.sqrt(s*(s-side1)*(s-side2)*(s-side3))
    else:
        print("Given sides do not form a triangle. The sum of two sides of a
        triangle should be greater than the third side")
        quit()

    return perimeter, area

if __name__ == "__main__":

    side1 = float(input("Enter the length of side 1: "))
    side2 = float(input("Enter the length of side 2: "))
    side3 = float(input("Enter the length of side 3: "))

    perimeter, area = calculate(side1, side2, side3)
    print(f'Perimeter : {perimeter} units')
    print(f"Area: {area:.3f} sq. units")
```

Output 1 :

```
E:\Semester 3\Pyhton\Codes>pyhton Practical1.py
'pyhton' is not recognized as an internal or external command,
operable program or batch file.

E:\Semester 3\Pyhton\Codes>python Practical1.py
Enter the length of side 1: 5.6
Enter the length of side 2: 1.1
Enter the length of side 3: 2.2
Given sides do not form a triangle.
The sum of two sides of a triangle should be greater than the third side

E:\Semester 3\Pyhton\Codes>python Practical1.py
Enter the length of side 1: 7
Enter the length of side 2: 4
Enter the length of side 3: 5
Given sides form a triangle.
Perimeter : 16.0 units
Area: 9.798 sq. units
```

Question 2 :

Consider a showroom of electronic products, where there are various salesmen. Each salesman is given a commission of 5%, depending on the sales made per month. In case the sale done is less than 50000, then the salesman is not given any commission. Write a function to calculate total sales of a salesman in a month, commission and remarks for the salesman. Sales done by each salesman per week is to be provided as input. Use tuples/list to store data of salesmen.

Assign remarks according to the following criteria :

Excellent : Sales \geq 80000

Good : Sales \geq 60000 and $<$ 80000

Average : Sales \geq 40000 and $<$ 60000

Work Hard : Sales $<$ 40000

Solution 2 :

```
# Created By : ANUJ KUMAR SEN
def calc(sale1, sale2, sale3, sale4):
    total_sales = sale1+sale2+sale3+sale4
    com_amt = 0

    if total_sales >= 50000:
        com_amt = total_sales*0.05

    remarks = ""
    if total_sales >= 80000:
        remarks = "Excellent"
    elif total_sales >= 60000:
        remarks = "Good"
    elif total_sales >= 40000:
        remarks = "Average"
    elif total_sales < 40000:
        remarks = "Work Hard"

    return total_sales, com_amt, remarks

if __name__ == "__main__":
    sale1 = float(input("Enter the sales for week 1: "))
    sale2 = float(input("Enter the sales for week 2: "))
    sale3 = float(input("Enter the sales for week 3: "))
    sale4 = float(input("Enter the sales for week 4: "))
    t_sales, comm, remarks = calc(sale1, sale2, sale3, sale4)

    print(f"Total Sales : Rs. {t_sales:.2f}")
    print(f"Commission : Rs. {comm:.2f}")
    print(f"Remarks : {remarks}")
```

Output 2 :

```
E:\Semester 3\Pyhton\Codes>python Practical2.py
Enter the sales for week 1: 45000
Enter the sales for week 2: 78000.56
Enter the sales for week 3: 80000.21
Enter the sales for week 4: 20000.1
Total Sales : Rs. 223000.87
Commission : Rs. 11150.04
Remarks : Excellent
```

```
E:\Semester 3\Pyhton\Codes>python Practical2.py
Enter the sales for week 1: 2000.5
Enter the sales for week 2: 1500
Enter the sales for week 3: 6000
Enter the sales for week 4: 4500
Total Sales : Rs. 14000.50
Commission : Rs. 0.00
Remarks : Work Hard
```

```
E:\Semester 3\Pyhton\Codes>python Practical2.py
Enter the sales for week 1: 15000
Enter the sales for week 2: 20000
Enter the sales for week 3: 15000
Enter the sales for week 4: 15000
Total Sales : Rs. 65000.00
Commission : Rs. 3250.00
Remarks : Good
```

```
E:\Semester 3\Pyhton\Codes>python Practical2.py
Enter the sales for week 1: 10000
Enter the sales for week 2: 10000
Enter the sales for week 3: 10000
Enter the sales for week 4: 12000
Total Sales : Rs. 42000.00
Commission : Rs. 0.00
Remarks : Average
```

```
E:\Semester 3\Pyhton\Codes>
```

Question 3 :

Write a Python function to find the nth term of Fibonacci sequence and its factorial. Return the result as a list.

Solution 3 :

```
# Created By : ANUJ KUMAR SEN

def factorial(n):
    if n <= 1:
        return 1
    else:
        return n * factorial(n-1)

def fibonnaci(n):
    if n <= 0:
        return 0
    if n == 1:
        return 1
    return fibonnaci(n-1) + fibonnaci(n-2)

if __name__ == '__main__':
    num = int(input("Enter a number: "))
    print(f"{num} term of fibonacci series : ", fibonnaci(num))
    print(f"Factorial of {num} : ", factorial(num))
```

Output 3 :

```
E:\Semester 3\Pyhton\Codes>python Practical3.py
Enter a number: 6
6 term of fibonacci series : 8
Factorial of 6 : 720

E:\Semester 3\Pyhton\Codes>python Practical3.py
Enter a number: 15
15 term of fibonacci series : 610
Factorial of 15 : 1307674368000
```

Question 4 :

Write a function that takes a number (≥ 10) as an input and return the digits of the number as a set.

Solution 4 :

```
# Created By : ANUJ KUMAR SEN

num = int(input("Enter a number greater than or equal to 10 : "))

if num >= 10:
    _set = set()
    while num != 0:
        _set.add(num%10)
        num = int(num/10)
    print("Set: ", _set)
else:
    print("Oops! Number is less than 10")
```

Output 4 :

```
E:\Semester 3\Pyhton\Codes>python Practical4.py
Enter a number greater than or equal to 10 : 15987
Set: {1, 5, 7, 8, 9}

E:\Semester 3\Pyhton\Codes>python Practical4.py
Enter a number greater than or equal to 10 : 5
Oops! Number is less than 10

E:\Semester 3\Pyhton\Codes>
```

Question 5 :

Write a function that finds the sum of the n terms of the following series. Import the factorial function created in question 4.

Series : $1 - x^2/2! + x^4/4! - x^6/6! + \dots x^n/n!$

Solution 5 :

```
# Created By : ANUJ KUMAR SEN

from Practical3 import factorial

def sum_series(x, n):
    sum = 0
    for i in range(1, n+1):
        term = ((-1)**(i+1)) * (x**(2*i-2) / factorial(2*i-2))
        sum += term
    return sum

if __name__ == "__main__":
    n = int(input("Enter n: "))
    x = int(input("Enter x: "))
    sum = sum_series(x, n)
    print(f'Sum of {n} terms of series for x={x} is {sum}')
```

Output 5 :

```
E:\Semester 3\Pyhton\Codes>python Practical5.py
Enter n: 6
Enter x: 2
Sum of 6 terms of series for x=2 is -0.4161552028218696

E:\Semester 3\Pyhton\Codes>python Practical5.py
Enter n: 2
Enter x: 1
Sum of 2 terms of series for x=1 is 0.5
```


Question 6 :

Consider a tuple t1={1,2,5,7,9,2,4,6,8,10}.

Write a program to perform following operations:

- a) Print another tuple whose values are even numbers in the given tuple.
- b) Concatenate a tuple t2 = {11,13,15} with t1.
- c) Return maximum and minimum value from this tuple.

Solution 6 :

```
# Created By : ANUJ KUMAR SEN

t1 = (1,2,5,7,9,2,4,6,8,10)
print("Given tuple: ", t1)

# a) Print another tuple whose values are even numbers in the given tuple.
t_even = ()
for i in range(0, len(t1)):
    if t1[i] % 2 == 0:
        t_even += (t1[i],)
print("New tuple with even values : ", t_even)

# b) Concatenate a tuple t2 = {11,13,15} with t1.
t2 = (11, 13, 15)
t3 = t1 + t2
print("Concatented tuple : ", t3)

# c) Return maximum and minimum value from this tuple.
maximum = t3[0]
minimum = t3[0]
for i in range(0, len(t3), 1):
    if t3[i] > maximum:
        maximum = t3[i]
    if t3[i] < minimum:
        minimum = t3[i]
print("Maximum : ", maximum)
print("Minimum : ", minimum)
```

Output 6 :

```
E:\Semester 3\Pyhton\Codes>python Practical6.py
Given tuple: (1, 2, 5, 7, 9, 2, 4, 6, 8, 10)
New tuple with even values : (2, 2, 4, 6, 8, 10)
Concatented tuple : (1, 2, 5, 7, 9, 2, 4, 6, 8, 10, 11, 13, 15)
Maximum : 15
Minimum : 1
```

Question 7 :

Write a menu driven program to perform the following on strings:

- Find the length of the string.
- Return maximum of three strings.
- Accept a string and replace all vowels with “#”
- Find the number of words in the given string.
- Check whether the string is a palindrome or not.

Solution 7 :

```
# Created By : ANUJ KUMAR SEN

# a) Find the length of string.
def len_str():
    str = input("Enter the string: ")
    print("Length of string : ", len(str))

# b) Return maximum of three strings.
def maxof_three():
    str1 = input("Enter string 1 : ")
    str2 = input("Enter string 2 : ")
    str3 = input("Enter string 3 : ")
    maxstr = ""
    if str1 > str2 and str1 > str3:
        maxstr = str1
    elif str2 > str1 and str2 > str3:
        maxstr = str2
    else:
        maxstr = str3
    print("Maximum of above three: ", maxstr)

# c) Accept a string and replace all vowels with "#"
def replace_vowels():
```

```

str = input("Enter the string : ")
new_str = ""
vowels = ['a', 'e', 'i', 'o', 'u']
for i in range(len(str)):
    if str[i].lower() in vowels:
        new_str += "#"
    else:
        new_str += str[i]
print("Replaced string : ", new_str)

# d) Find number of words in the given string.
def numofwords():
    str = input("Enter the string : ")
    str = str.strip() + " "
    count = 0
    for i in range(len(str)):
        if str[i] == " ":
            count += 1
    print("No of words: ", count)

# e) Check whether the string is a palindrome or not.
def palindrome():
    str = input("Enter the string : ")
    new_str = ""
    for i in range(len(str)):
        new_str = str[i] + new_str
    if str.lower() == new_str.lower():
        print(f"{str} is a palindrome.")
    else:
        print(f"{str} is not a palindrome")

def main():
    print("\nMenu")
    print("-"*20)
    print("1. Length of string")
    print("2. Maximum of three strings")
    print("3. Replace vowels with '#')")
    print("4. No. of words")
    print("5. Check Palindrome")
    print("6. Exit")
    print("-"*20)
    option = input("Your choice: ")

    switcher = {
        '1': len_str,
        '2': maxof_three,

```

```

        '3': replace_vowels,
        '4': numofwords,
        '5': palindrome,
        '6': quit
    }

    func = switcher.get(option, lambda: print("Invalid Choice!"))
    func()

if __name__ == "__main__":

    ch = 'y'
    while ch.lower() == 'y':
        main()
        ch = input("\nWant to continue? [y/n]: ")

```

Output 7 :

```

E:\Semester 3\Python\Codes>python Practical17.py
Menu
-----
1. Length of string
2. Maximum of three strings
3. Replace vowels with '#'
4. No. of words
5. Check Palindrome
6. Exit
-----
Your choice: 1
Enter the string: Python
Length of string : 6

Want to continue? [y/n]: y

Menu
-----
1. Length of string
2. Maximum of three strings
3. Replace vowels with '#'
4. No. of words
5. Check Palindrome
6. Exit
-----
Your choice: 2
Enter string 1 : Java
Enter string 2 : Data Structures
Enter string 3 : Maven
Maximum of above three:  Maven

Want to continue? [y/n]: y

Menu
-----
1. Length of string
2. Maximum of three strings
3. Replace vowels with '#'
4. No. of words
5. Check Palindrome
6. Exit
-----
Your choice: 3
Enter the string : Computer Networks
Replaced string :  C#mp#t#r N#tw#rks

Want to continue? [y/n]: y

Menu
-----
1. Length of string
2. Maximum of three strings
3. Replace vowels with '#'
4. No. of words
5. Check Palindrome
6. Exit
-----
Your choice: 4
Enter the string : Git is a version control system
No of words:  6

Want to continue? [y/n]: y

Menu
-----
1. Length of string
2. Maximum of three strings
3. Replace vowels with '#'
4. No. of words
5. Check Palindrome
6. Exit
-----
Your choice: 5
Enter the string : Naman
Naman is a palindrome.

Want to continue? [y/n]: y

Menu
-----
1. Length of string
2. Maximum of three strings
3. Replace vowels with '#'
4. No. of words
5. Check Palindrome
6. Exit
-----
Your choice: 5
Enter the string : Operating
Operating is not a palindrome

Want to continue? [y/n]: n
E:\Semester 3\Python\Codes>

```

Question 8 :

Write a Python program to perform the following using list:

- a) Check if all elements in list are numbers or not.
- b) If it is a numeric list, then count number of odd values in it.
- c) If list contains all Strings, then display largest String in the list.
- d) Display list in reverse form.
- e) Find a specified element in list.
- f) Remove the specified element from the list.
- g) Sort the list in descending order.
- h) Accept 2 lists and find the common members in them

Solution 8 :

```
# Created By : ANUJ KUMAR SEN

# a) Check if all elements in list are numbers or not.
def check_int(l):
    for i in range(0, len(l), 1):
        if not l[i].isnumeric():
            return False
    return True

# b) If it is a numeric list, then count number of odd values in it.
def count_odd(l):
    if check_int(l):
        count = 0
        for i in range(0, len(l), 1):
            if int(l[i]) % 2 != 0:
                count += 1
        print("Count of odd numbers : ", count)

# c) If list contains all Strings, then display largest String in the list.
def largest_str(l):
    flag = True
    for i in range(len(l)):
        if type(l[i]) != str:
            flag = False

    if flag:
        largest = l[0]
        for i in l:
            if len(i) > len(largest):
                largest = i
        print("Largest string: ", largest)
    else:
```

```

        print("List does not contain all strings!")

# d) Display list in reverse form.
def display_reverse(l):
    for i in range(len(l)-1, -1, -1):
        print(l[i], end=" ")
    return

# e) Find a specified element in list.
def find_item(l):
    item = input("\nEnter an element: ")
    for i in range(0, len(l), 1):
        if item == l[i]:
            print("Item found at index: ", i)
            return
    print("Item not found")

# f) Remove the specified element from the list.
def remove_item(l):
    item = input("\nEnter an element: ")
    for i in range(0, len(l), 1):
        if item == l[i]:
            l.remove(item)
            print("Item removed")
    return

# g) Sort the list in descending order.
def sort_desc(l):
    print(sorted(l, reverse=True))
    return

# h) Accept 2 lists and find the common members in them
def common(l1, l2):
    common = []
    for i in range(0, len(l1), 1):
        for j in range(0, len(l2), 1):
            if l1[i] == l2[j]:
                common.append(l1[i])
    if common:
        print("Common elements: ", common)
    else:
        print("No common element")
    return

def main(l):
    print("\nMenu")

```

```

print("-"*20)
print("1. Check if all elements are numbers")
print("2. Count odd numbers if list is numeric")
print("3. Display largest string in list")
print("4. Reverse the list")
print("5. Find item in list")
print("6. Remove item from list")
print("7. Sort in Descending order")
print("8. Find common elements from another list")
print("9. Exit")
print("-"*20)
option = input("Your choice: ")
switcher = {
    '2': count_odd,
    '3': largest_str,
    '4': display_reverse,
    '5': find_item,
    '6': remove_item,
    '7': sort_desc,
    '8': common,
    '9': quit
}
if option == '1':
    if check_int(1):
        print("All elements are numbers")
    else:
        print("All elements are not numbers")
elif option == '8':
    l2 = []
    n = int(input("Enter the size of new list: "))
    for i in range(0, n, 1):
        l2.append(input("Enter element: "))
    common(1, l2)
else:
    func = switcher.get(option, lambda: print("Invalid Choice!"))
    func(1)

if __name__ == "__main__":
    l = []
    n = int(input("Enter the size of list: "))
    for i in range(0, n, 1):
        l.append(input("Enter element: "))
    ch = 'y'
    while ch.lower() == 'y':
        main(l)
        ch = input("\nWant to continue? [y/n]: ")

```

Output 8 :

```
E:\Semester 3\Pyhton\Codes>python Practical8.py
Enter the size of list: 4
Enter element: 4
Enter element: 5
Enter element: 7
Enter element: 2
```

```
Menu
-----
1. Check if all elements are numbers
2. Count odd numbers if list is numeric
3. Display largest string in list
4. Reverse the list
5. Find item in list
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 1
All elements are numbers

Want to continue? [y/n]: y
```

```
Menu
-----
1. Check if all elements are numbers
2. Count odd numbers if list is numeric
3. Display largest string in list
4. Reverse the list
5. Find item in list
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 2
Count of odd numbers : 2

Want to continue? [y/n]: y
```

```
Menu
-----
1. Check if all elements are numbers
2. Count odd numbers if list is numeric
3. Display largest string in list
4. Reverse the list
5. Find item in list
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 4
2 7 5 4
Want to continue? [y/n]: y
```

```
Menu
-----
1. Check if all elements are numbers
2. Count odd numbers if list is numeric
3. Display largest string in list
4. Reverse the list
5. Find item in list
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 5

Enter an element: 1
Item not found
```

```
Want to continue? [y/n]: y
```

```
E:\Semester 3\Pyhton\Codes>python Practical8.py
Enter the size of list: 3
Enter element: Java
Enter element: 5
Enter element: Python
```

```
Menu
-----
1. Check if all elements are numbers
2. Count odd numbers if list is numeric
3. Display largest string in list
4. Reverse the list
5. Find item in list
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 1
All elements are not numbers

Want to continue? [y/n]: y
```

```
Menu
-----
1. Check if all elements are numbers
2. Count odd numbers if list is numeric
3. Display largest string in list
4. Reverse the list
5. Find item in list
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 3
Largest string: Python

Want to continue? [y/n]: y
```

```
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 3
Largest string: Python

Want to continue? [y/n]: y
```

```
Menu
-----
1. Check if all elements are numbers
2. Count odd numbers if list is numeric
3. Display largest string in list
4. Reverse the list
5. Find item in list
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 7
['Python', 'Java', '5']

Want to continue? [y/n]: y
```

```
Menu
-----
1. Check if all elements are numbers
2. Count odd numbers if list is numeric
3. Display largest string in list
4. Reverse the list
5. Find item in list
6. Remove item from list
7. Sort in Descending order
8. Find common elements from another list
9. Exit
```

```
-----
Your choice: 8
Enter the size of new list: 4
Enter element: Python
Enter element: Go
Enter element: SQL
Enter element: Solidity
Common elements: ['Python']
```

```
Want to continue? [y/n]: n
```


Question 9 :

Use dictionary to store marks of the students in 4 subjects. Write a function to find the name of the student securing highest percentage.
(Hint: Names of students are unique).

Solution 9 :

```
# Created By : ANUJ KUMAR SEN

def findTopStudent(marks):
    name = ''
    percentage = 0.0
    for student in marks:
        marks_list = marks[student]
        total = 0.0
        for i in range(0, len(marks_list), 1):
            total += marks_list[i]
        if percentage < total/4:
            percentage = total/4
            name = student
    return name

if __name__ == "__main__":
    marks = {}
    num = int(input('Enter the no. of students: '))
    for i in range(1, num+1, 1):
        name = input('\nEnter the name of Student {}: '.format(i))
        temp = []
        for j in range(1, 5, 1):
            mark = float(input('Marks in Subject {} (out of 100): '.format(j)))
            temp.append(mark)
        marks[name] = temp

    topper = findTopStudent(marks)
    print('\n{} secured the highest percentage.'.format(topper))
```

Output 9 :

```
E:\Semester 3\Pyhton\Codes>python Practical9.py
Enter the no. of students: 3

Enter the name of Student 1: Raj
Marks in Subject 1(out of 100): 98
Marks in Subject 2(out of 100): 95
Marks in Subject 3(out of 100): 87
Marks in Subject 4(out of 100): 92

Enter the name of Student 2: Iyer
Marks in Subject 1(out of 100): 78
Marks in Subject 2(out of 100): 95
Marks in Subject 3(out of 100): 88
Marks in Subject 4(out of 100): 90

Enter the name of Student 3: Dev
Marks in Subject 1(out of 100): 89
Marks in Subject 2(out of 100): 75
Marks in Subject 3(out of 100): 99
Marks in Subject 4(out of 100): 84

Raj secured the highest percentage.
```

Question 10 :

Write a function that takes a sentence as input from the user and calculates the frequency of each letter.

Use a variable of dictionary type to maintain the count.

Solution 10 :

```
# Created By : ANUJ KUMAR SEN

def calcFreqOfLetters(sentence):
    freq = {}
    for i in range(0, len(sentence), 1):
        letter = sentence[i].lower()
        if letter not in [' ', '.', ',', '\'', '\"', '!', ';']:
            if letter in freq.keys():
                freq[letter] += 1
            else:
```

```

        freq[letter] = 1
    return freq

if __name__ == "__main__":
    sentence = input('\nEnter a sentence : ')
    freq = calcFreqOfLetters(sentence)

    print('\nFrequencies of each letter in the given sentence : ')
    for letter in freq:
        print('\''{}\' --> {}'.format(letter, freq[letter]))

```

Output 10 :

```

E:\Semester 3\Pyhton\Codes>python Practical10.py

Enter a sentence : India is the fifth largest economy in the world

Frequencies of each letter in the given sentence :
'i' --> 5
'n' --> 3
'd' --> 2
'a' --> 2
's' --> 2
't' --> 4
'h' --> 3
'e' --> 4
'f' --> 2
'l' --> 2
'r' --> 2
'g' --> 1
'c' --> 1
'o' --> 3
'm' --> 1
'y' --> 1
'w' --> 1

```

Question 11 :

Write a menu-driven program to accept a list of student names and perform the following

- a. Search an element using linear search/ binary search.
- b. Sort the elements using bubble sort/ insertion sort/ selection sort.

Solution 11 :

```
# Created By : ANUJ KUMAR SEN

def bubbleSort(list):
    for i in range(0, len(list)-1, 1):
        for j in range(0, len(list) - i - 1, 1):
            if list[j] > list[j+1]:
                temp = list[j]
                list[j] = list[j+1]
                list[j+1] = temp
    return list

def insertionSort(list):
    for i in range(0, len(list), 1):
        temp = list[i]
        j = i - 1
        while j >= 0 and list[j] > temp:
            list[j+1] = list[j]
            j -= 1
        list[j+1] = temp
    return list

def selectionSort(list):
    for i in range(0, len(list)-1, 1):
        minIndex = i
        for j in range(i+1, len(list), 1):
            if list[minIndex] > list[j]:
                minIndex = j
        temp = list[minIndex]
        list[minIndex] = list[i]
        list[i] = temp
    return list

def linearSearch(list, element):
    for i in range(0, len(list), 1):
        if list[i].lower() == element.lower():
            return i
    return -1
```

```

def binarySearch(list, element):
    list = selectionSort(list)
    low = 0
    high = len(names) - 1
    while low <= high:
        mid = int((low + (high - low) / 2))
        if (list[mid] == element):
            return mid
        if (list[mid] > element):
            high = mid - 1
        else:
            low = mid + 1
    return -1

if __name__ == "__main__":
    num = int(input('\nEnter the number of students: '))
    print('Enter the names of students:')
    names = []
    for i in range(0, num, 1):
        names.append(input('{}: '.format(i+1)))

    choice = 'y'
    while choice.lower() == 'y':
        print('\n----- Menu -----')
        print('1. Search a name')
        print('2. Sort the list of names')
        choice = input('Your Choice: ')
        if choice == '1':
            name = input('\nEnter a name to search: ')
            choice = input('Choose a searching algorithm:\n1. Linear, 2. Binary: ')
            index = -1
            if choice == '1':
                index = linearSearch(names, name)
            elif choice == '2':
                index = binarySearch(names, name)
            else:
                print('Invalid Choice!')
            if index == -1:
                print('Name is not in the list.')
            else:
                print('Name found in the list.')
        elif choice == '2':
            choice = input('Choose a sorting algorithm:\n1. Bubble, 2. Insertion,
3. Selection: ')
            sorted_names = []
            if choice == '1':

```

```

        sorted_names = bubbleSort(names)
    elif choice == '2':
        sorted_names = insertionSort(names)
    elif choice == '3':
        sorted_names = selectionSort(names)
    else:
        print('Invalid Choice!')
    print('Sorted list: ', end='')
    for i in range(0, len(sorted_names), 1):
        print('\n{}\n'.format(sorted_names[i]), end=' ')
    print('\n')
else:
    print('Invalid Choice!')

choice = input('\nWould you like to continue? (y/n): ')

```

Output 11 :

```
E:\Semester 3\Python\Codes>python Practical11.py
```

```

Enter the number of students: 3
Enter the names of students:
1: Rahul
2: Priyanka
3: Varun

```

```

----- Menu -----
1. Search a name
2. Sort the list of names
Your Choice: 1

```

```

Enter a name to search: Amit
Choose a searching algorithm:
1. Linear, 2. Binary: 1
Name is not in the list.

```

```
Would you like to continue? (y/n): y
```

```

----- Menu -----
1. Search a name
2. Sort the list of names
Your Choice: 1

```

```

Enter a name to search: Varun
Choose a searching algorithm:
1. Linear, 2. Binary: 2
Name found in the list.

```

```
Would you like to continue? (y/n): y
```

```

----- Menu -----
1. Search a name
2. Sort the list of names
Your Choice: 2
Choose a sorting algorithm:
1. Bubble, 2. Insertion, 3. Selection: 1
Sorted list: 'Priyanka' 'Rahul' 'Varun'

```

```
Would you like to continue? (y/n): y
```

```

----- Menu -----
1. Search a name
2. Sort the list of names
Your Choice: 2
Choose a sorting algorithm:
1. Bubble, 2. Insertion, 3. Selection: 2
Sorted list: 'Priyanka' 'Rahul' 'Varun'

```

```
Would you like to continue? (y/n): y
```

```

----- Menu -----
1. Search a name
2. Sort the list of names
Your Choice: 2
Choose a sorting algorithm:
1. Bubble, 2. Insertion, 3. Selection: 3
Sorted list: 'Priyanka' 'Rahul' 'Varun'

```

```
Would you like to continue? (y/n): n
```

Question 12 :

Write a program that makes use of a function to accept a list of n integers and displays a histogram.

Solution 12 :

```
# Created By : ANUJ KUMAR SEN

import matplotlib.pyplot as plt

def inputList():
    '''
    To take a input of list of integers from user
    Returns the list
    '''
    ls = []
    length = int(input('Enter the length of list: '))
    for i in range(0, length, 1):
        ls.append(int(input('Enter element {}:'.format(i+1))))

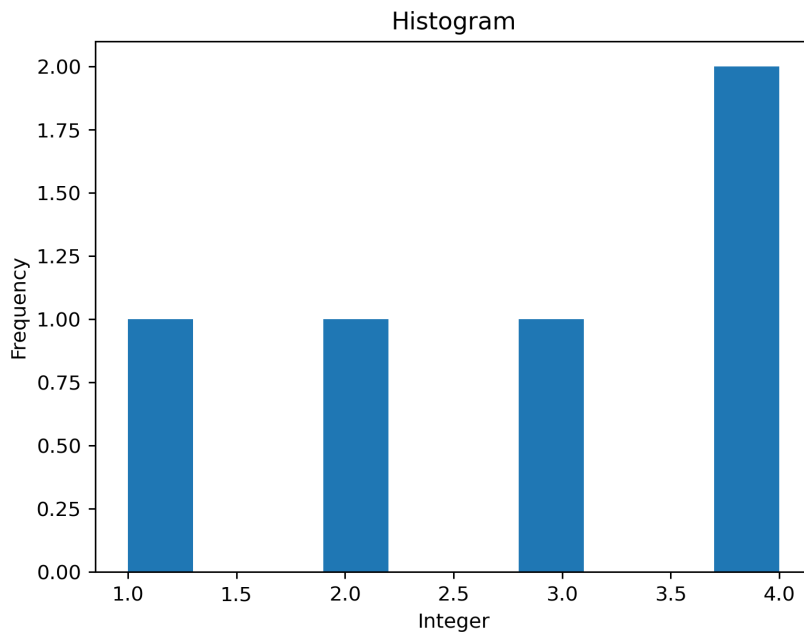
    return ls

if __name__ == "__main__":

    list = inputList()
    plt.hist(list, )
    plt.xlabel('Integer')
    plt.ylabel('Frequency')
    plt.title('Histogram')
    plt.savefig('hist.png', dpi=275, bbox_inches='tight') # To save the plot
    plt.show()
```

Output 12 :

```
E:\Semester 3\Pyhton\Codes>python Practical12.py
Enter the length of list: 5
Enter element 1:4
Enter element 2:3
Enter element 3:4
Enter element 4:2
Enter element 5:1
```



Question 13 :

Write a program that makes use of a function to display sine, cosine, polynomial and exponential curves.

Solution 13 :

```
# Created By : ANUJ KUMAR SEN

import matplotlib.pyplot as plt
import math

def sineCurve():
    '''
    To plot sine function
    '''
```



```
plt.subplot(2, 2, 1)
degrees = range(0, 360 + 1)
sinValues = [math.sin(math.radians(i)) for i in degrees]
plt.plot(sinValues)
plt.xlabel('Degrees')
plt.ylabel('Sin Values')
plt.title('Sine Curve')
plt.grid()
```

```
def cosineCurve():
    '''
    To plot cos function
    '''

    plt.subplot(2, 2, 3)
    degrees = range(0, 360 + 1)
    cosValues = [math.cos(math.radians(i)) for i in degrees]
    plt.plot(degrees, cosValues)
    plt.xlabel('Degrees')
    plt.ylabel('Cosine Values')
    plt.title('Cosine Curve')
    plt.grid()
```

```
def polynomialCurve():
    '''
    To plot a polynomial function
    '''

    def polynomial(x):
        return (8*x*x)

    plt.subplot(2, 2, 2)
    x = range(-51, 50 + 2)
    y = [polynomial(i) for i in x]
    plt.plot(x, y)
    plt.xlabel('x')
    plt.ylabel('y = 8x^2')
    plt.title('Polynomial Curve')
    plt.grid()
```

```
def expCurve():
    '''
    To plot sine function
    '''
```

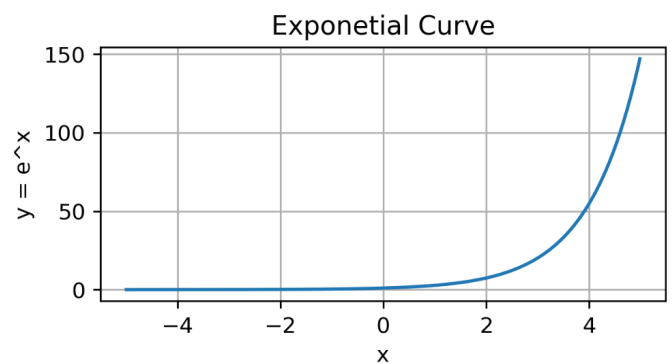
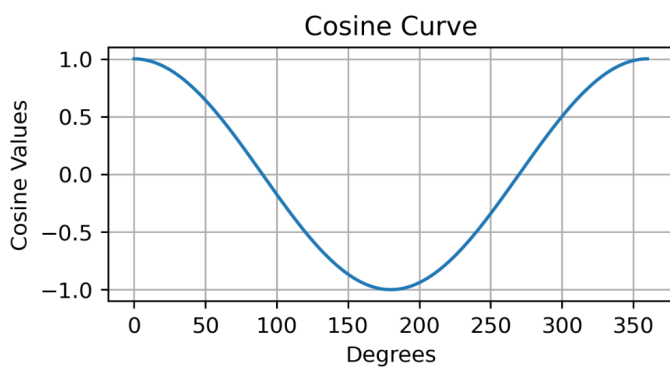
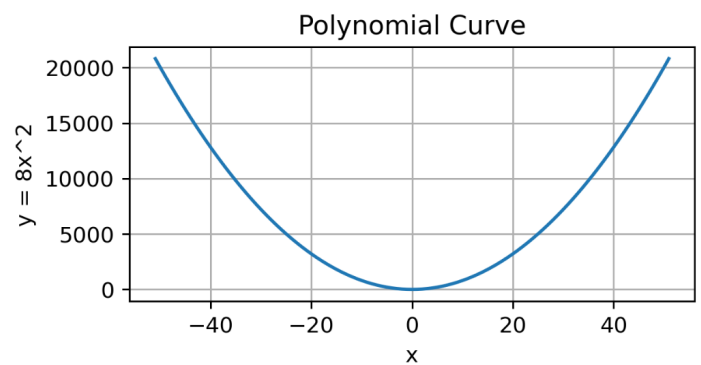
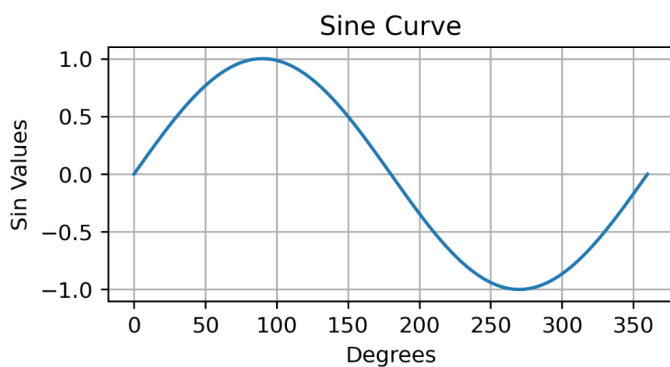
```

plt.subplot(2, 2, 4)
x = []
for i in range(0, 100*10):
    x.append((-5) + (0.01)*i)
y = [math.exp(i) for i in x]
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y = e^x')
plt.title('Exponetial Curve')
plt.grid()

if __name__ == "__main__":
    plt.figure(figsize=(9, 5)) # To set the figure size
    sineCurve()
    cosineCurve()
    polynomialCurve()
    expCurve()
    plt.tight_layout()
    plt.savefig('plot.png', dpi=275, bbox_inches='tight') # To save the plot
    plt.show()

```

Output 13 :



Question 14 :

Write a function that reads a file file1 and copies only alternative lines to another file file2. Alternative lines copied should be the odd numbered lines. Use Exception.

Solution 14 :

```
# Created By : ANUJ KUMAR SEN

#Don't forget to create file1.txt and file2.txt

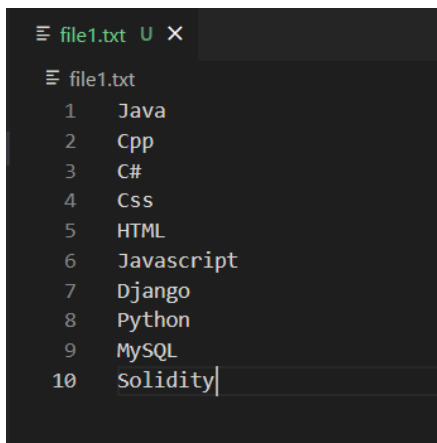
def copyOddNumberedLines(file_in, file_out):
    print("\nCopying alternate lines from '{}' to '{}...'".format(
        file_in, file_out))
    try:
        fh_in = open(file_in, 'r')
        fh_out = open(file_out, 'w')
        lines = fh_in.readlines()
        for i in range(0, len(lines), 2):
            fh_out.write(lines[i])
        print('Done!\n')
    except FileNotFoundError:
        print('Error: {} not found.\n'.format(file_in))

if __name__ == "__main__":

    copyOddNumberedLines('file1.txt', 'file2.txt')
```

Output 14 :

File 1 :



```
file1.txt U X
file1.txt
1  Java
2  Cpp
3  C#
4  Css
5  HTML
6  Javascript
7  Django
8  Python
9  MySQL
10 Solidity
```

```
E:\Semester 3\Python\Codes>python Practical14.py

Copying alternate lines from 'file1.txt' to 'file2.txt'...
Done!
```

File 2 :

```
file2.txt
1   Java
2   C#
3   HTML
4   Django
5   MySQL
6   |
```

Question 15 :

Define a class Student to store his/ her name and marks in three subjects.
Use a class variable to store the maximum average marks of the class.
Use constructor and destructor to initialize and destroy the objects.

Solution 15 :

```
# Created By : ANUJ KUMAR SEN

class Student:
    # Class Variable
    max_avg = 0

    # Constructor
    def __init__(self, name='', marks=[0, 0, 0]):
        self.name = name
        self.marks = marks

    # Destructor
    def __del__(self):
        del self.name
        del self.marks
        del self

    def __str__(self):
        str = '\nMarks: {} \n'.format(self.name)
        str += '-----\n'
```

```

        str += 'Subject 1: {}'.format(self.marks[0])
        str += 'Subject 2: {}'.format(self.marks[1])
        str += 'Subject 3: {}'.format(self.marks[2])
        str += 'Average: {:.2f}'.format(self.calcAvg())
        return str

def calcAvg(self):
    total = 0.0
    for i in range(0, 3, 1):
        total += self.marks[i]
    return total/3

if __name__ == "__main__":

    students = []
    num = int(input('\nEnter the no. of students: '))
    for i in range(1, num+1, 1):
        name = input('\nEnter the name of Student {}: '.format(i))
        marks = []
        for j in range(1, 4, 1):
            marks.append(float(input('Marks in Subject {}: '.format(j))))

        student = Student(name, marks)
        if Student.max_avg < student.calcAvg():
            Student.max_avg = student.calcAvg()
        students.append(student)

    for i in range(0, len(students), 1):
        print(students[i])

    print('\nMaximum average marks of class: {:.2f}\n'.format(Student.max_avg))

```

Output 15 :

```
E:\Semester 3\Pyhton\Codes>python Practical15.py

Enter the no. of students: 3

Enter the name of Student 1: Hardik
Marks in Subject 1: 87
Marks in Subject 2: 90
Marks in Subject 3: 91

Enter the name of Student 2: Rohit
Marks in Subject 1: 88
Marks in Subject 2: 90
Marks in Subject 3: 93

Enter the name of Student 3: Ishan
Marks in Subject 1: 88
Marks in Subject 2: 76
Marks in Subject 3: 81

Marks: Hardik
-----
Subject 1: 87.0
Subject 2: 90.0
Subject 3: 91.0
Average: 89.33

Marks: Rohit
-----
Subject 1: 88.0
Subject 2: 90.0
Subject 3: 93.0
Average: 90.33

Marks: Ishan
-----
Subject 1: 88.0
Subject 2: 76.0
Subject 3: 81.0
Average: 81.67

Maximum average marks of class: 90.33

E:\Semester 3\Pyhton\Codes>
```

END OF ASSIGNMENT