PYTHON PRACTICALS

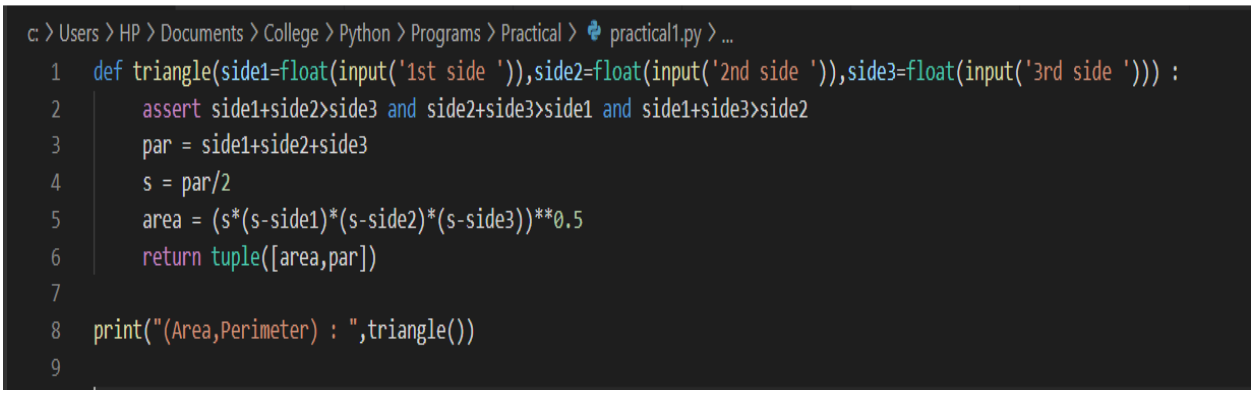
PYUSH DEEP

19/78096

B.Sc.(H) C.S. – 3rd Sem

Practical 1

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



Practical 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 that calculates 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 the salesmen.

Assign remarks according to the following criteria:

- Excellent: Sales >= 80000

- Good: 60000 <= Sales < 80000
- Average: 40000 <= Sales < 60000
- Work Hard: Sales < 40000

```python
def sales():
    week1 = int(input('enter sales in first week '))
    week2 = int(input('enter sales in second week '))
    week3 = int(input('enter sales in third week '))
    week4 = int(input('enter sales in fourth week '))
    monthlysale = week1+week2+week3+week4
    commission = "No Commission"
    if monthlysale >50000:
        commission = float(monthlysale*0.05)
    if monthlysale >= 80000:
        remark = 'Excellent'
    elif monthlysale >= 60000:
        remark = 'Good'
    elif monthlysale >= 40000:
        remark = 'Average'
    else :
        remark = 'work hard'
    salesman = (monthlysale,commission,remark)
    print('Monthly sale =',monthlysale,'Commission =',commission,'Remark =',remark)

sales()
```

Practical 3

Write a Python function to find the n^{th} term of Fibonacci sequence and it's factorial. Return the result as a list.

```python
def factorial(n):
    if n == 0:
        return 1
    return n*factorial(n-1)

def fib(n):
    if n == 1 or n == 2:
        return 1
    else:
        return fib(n-1)+fib(n-2)

def fib_factorial(n):
    return [fib(n),factorial(fib(n))]

print(fib_factorial(1))
print(fib_factorial(2))
print(fib_factorial(3))
print(fib_factorial(4))
print(fib_factorial(5))
print(fib_factorial(6))
print(fib_factorial(7))
```

Practical 4

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

```python
def digits():
    num = input('Enter a number >=10 :')
    s = []
    for i in num:
        s.append(i)
    return set(s)

print("Set : ", digits())
```

Practical 5

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

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \frac{x^n}{n!}$$

```python
def factorial(n):
    if n == 0 or n == 1:
        return 1
    return n*factorial(n-1)

def series():
    x = int(input("Enter the value of x"))
    n = int(input("enter the length of series"))
    sum = 1.0
    for i in range(2,n+1,1):
        if i%2 == 0:
            sum -= x**i/factorial(i)
        else:
            sum += (x**i)/factorial(i)
    return sum

print(series())
```

Practical 6

Consider a tuple t1 = {1,2,5,7,9,2,4,6,8,10}. Write a program to perform the 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 the tuple.

```python
def  retrunEven(tpl):
    evenTuple = []
    for i in tpl:
        if i % 2 == 0:
            evenTuple.append(i)
    return tuple(evenTuple)

def addTuple(tpl_1,tpl_2):
    tpl_1 = list(tpl_1)
    tpl_2 = list(tpl_2)
    tpl = tpl_1 + tpl_2
    return tuple(tpl)

def max_min(tpl):
    max = tpl[0]
    min = tpl[0]
    for i in tpl:
        if i > max:
            max = i
        if i < min:
            min = i
    return (max,min)

def main():
    t1 = (1,2,5,7,9,2,4,6,8,10)
    flag = True
    while flag:
        print("Tuple t1: ", t1)
        print("1. Return a Tuple that contains \"EVEN VALUES\" of given Tuple t1 ")
        print("2. Concatenate Tuple t2 with t1")
        print("3. Return Maximum and Minimum Values of the Tuple t1")
        print("4. Exit")
        choice = int(input("What do you want to perform ? "))
        if choice == 1:
            print(retrunEven(t1))
        elif choice == 2:
            elif choice == 2:
                t2 = (11,13,15)
                t1 = addTuple(t1,t2)
                print(t1)
            elif choice == 3:
                print(max_min(t1))
            else:
                flag = False
            print("==========================================\n")

if __name__ == "__main__":
    main()

```

Practical 7

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

a) Find the length of string.
b) Return maximum of three strings.
c) Accept a string and replace all the vowels with “#”.
d) Find the number of words in the given string.
e) Check whether the string is a pallindrome or not

```python
def strlen(string):
    length = 0
    for s in string:
        length = length + 1
    return length

def maxOfThree(str1,str2,str3):
    string = ""
    if str1 >= str2 and str1 >= str3:
        string = str1
    elif str2 >= str1 and str2 >= str3:
        string = str2
    else:
        string = str3
    return string

def changeVowel(string):
    vowels = "aeiou"
    newString = ""
    for i in string:
        if i.lower() in vowels:
            newString += "#"
        else:
            newString += i
    return newString

def wordCount(string):
    string = string.split()
    return len(string)

def pallindrome(string):
    flag = True
    for i in range(0,strlen(string)//2,1):
        if string[i] != string[strlen(string) - 1 - i]:
            flag = False
            break
    return flag
```

```python
def main():
    flag = True
    while flag:
        print(
        '''
STRING OPERATIONS
1. Find the Length of a String.
2. Find the Maximum Of Three Strings.
3. Replace all the vowels in a String with '#'.
4. Find the Number of Words in a String.
5. Check whether a String is a 'Palindrome' or not
6. Exit.
        '''
        )
        choice = int(input("Enter Choice : "))
        if choice == 1:
            string = input("Enter String : ")
            print(strlen(string))
        elif choice == 2:
            strings = []
            for i in range(3):
                strings.append(input("String "+str(i+1)+" : "))
            print(maxOfThree(strings[0],strings[1],strings[2]))
        elif choice == 3:
            string = input("Enter String : ")
            print(changeVowel(string))
        elif choice == 4:
            string = input("Enter String : ")
            print(wordCount(string))
        elif choice == 5:
            string = input("Enter String : ")
            print(pallindrome(string))
        else:
            flag = False
        print()
if __name__ == "__main__":
    main()
```

Practical 8

Write a Python program to perform the following using ‘list’:

a) Check if all the elements in the list are numbers or not.
b) If it is a numeric list, then count the number of odd values in it.
c) If the list contains all String , then display largest String in the list.
d) Display the 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.

```python
def isNum(l):
    flag = True
    for i in l:
        if not i.isnumeric():
            flag = False
            break
    return flag

def countOdd(l):
    if isNum(l):
        counter = 0
        for i in l:
            if int(i) % 2 != 0:
                counter += 1
        return counter
    return "List is not Numeric."

def maxString(l):
    flag = True
    for i in l:
        if i.isnumeric():
            flag = False
            break
    if flag:
        max = l[0]
        for i in l:
            if max < i:
                max = i
        return max
    return "List contains other datatypes than String."

def reverseList(l):
    for i in range(len(l)-1,-1,-1):
        print(l[i],end=' ')

def search(element,l):
    if element in l:
```

```
        return "Element is in List"
    return "Element is not in the List"

def sort(l):
    for i in range(1,len(l),1):
        key = l[i]
        j = i-1
        while j >= 0 and key > l[j]:
            l[j+1] = l[j]
            j = j - 1
        l[j+1] = key

def common(l1,l2):
    commmonList = []
    for i in l1:
        if i in l2:
            commmonList.append(i)
    return commmonList

def main():
    flag = True
    numberOfElements = int(input("Number of Elements in List : "))
    list1 = []
    for i in range(numberOfElements):
        list1.append(input())
    while flag:
        print("LIST :- ",list1)
        print(
'''
LIST OPERATIONS
1. Check if List is numeric or not.
2. Number of Odd Elments in List (if List is numeric)
3. Display the Largest String of the List (if all List Elements are of String Type).
4. Display the List in reverse.
5. Find a specified Element in List.
6. Remove a specified Element from List..
```

```
6. Remove a specified Element from List..
7. Sort the List in Descending Order.
8. Display Common Elements of List and another List from User.
9. Exit
''')
        choice = int(input("Enter Choice : "))
        if choice == 1:
            print(isNum(list1))
        elif choice == 2:
            print(countOdd(list1))
        elif choice == 3:
            print(maxString(list1))
        elif choice == 4:
            reverseList(list1)
        elif choice == 5:
            ele = input("Enter the Element : ")
            print(search(ele,list1))
        elif choice == 6:
            ele = input("Enter the Element to be removed : ")
            list1.remove(ele)
        elif choice == 7:
            sort(list1)
        elif choice == 8:
            numberOfElements = int(input("How many elements you want in List 2 :"))
            list2 = []
            for i in range(numberOfElements):
                list2.append(input())
            print("Common Elements : ",common(list1,list2))
        else:
            flag = False
        print()

if __name__ == "__main__":
    main()

```