**Set-1**

**Question – 1**

Problem Statement:

Write a prime generator using only primes and using python loops.

Source Code:

import math

def isPrime(n):

for i in range(2, n):

if(n % i == 0):

return 0

return 1

def getPrimes(limit):

for i in range(2, limit+1):

if(isPrime(i)):

yield i

limit=int(input("Enter the max limit: "))

for i in getPrimes(limit):

print(i)

Output:

SEM\_2\PYTHON>py q\_1.py

Enter the max limit: 20

2

3

5

7

11

13

17

19

**Question – 2**

Problem Statement:

Write a discount coupon code using dictionary in Python with different rate coupons for each day of the week.

Source Code:

dict = {}

dict["Monday"] = ("cp\_m123", "5%")

dict["Tuesday"] = ("cp\_yu253", "6%")

dict["Wednesday"] = ("cp\_we564", "2%")

dict["Thurseday"] = ("cp\_ts89", "3%")

dict["Friday"] = ("cp\_fd990", "10%")

dict["Saturday"] = ("cp\_sx343", "4%")

dict["Sunday"] = ("cp\_snd893", "8%")

while True:

day=input("Find coupon for day: ")

print(dict.get(day, "Invalid day"))

Output:

\SEM\_2\PYTHON>py q\_2.py

Find coupon for day: Monday

('cp\_m123', '5%')

Find coupon for day: Sunday

('cp\_snd893', '8%')

Find coupon for day: hello

Invalid day

Find coupon for day: Friday

('cp\_fd990', '10%')

Find coupon for day: exit

**Question – 3**

Problem Statement:

Print first 10 odd and even numbers using iterators and compress. You can use duck typing.

Source Code:

from itertools import compress

num\_list = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]

list\_Even = [True if x%2 == 0 else False for x in num\_list]

list\_Odd = [True if x%2 != 0 else False for x in num\_list]

print("Even numbers are: ")

for i in compress(num\_list, list\_Even):

print(i, end=", ")

print("\nOdd numbers are: ")

for i in compress(num\_list, list\_Odd):

print(i, end=", ")

Output:

SEM\_2\PYTHON>py q\_3.py

Even numbers are:

2, 4, 6, 8, 10, 12, 14, 16, 18, 20,

Odd numbers are:

1, 3, 5, 7, 9, 11, 13, 15, 17, 19,

**Question – 4**

Problem Statement:

Write a regular expression to validate a phone number.

Source Code:

import re

def isValidNum(phoneNum):

regex = "^[6-9][0-9]{9}"

valid = re.search(regex, phoneNum)

if valid is not None:

print("This is a Valid phone number")

else:

print("This is NOT a Valid phone number")

phoneNum = input("Enter a phone number: ")

isValidNum(phoneNum)

Output:

SEM\_2\PYTHON>py q\_4.py

Enter a phone number: 1234567892

This is NOT a Valid phone number

SEM\_2\PYTHON>py q\_4.py

Enter a phone number: 9833056002

This is a Valid phone number

**Question – 5**

Problem Statement:

Write first seven Fibonacci numbers using generator next function / yield function in Python. Trace and memorize the function.

Source Code:

def fibo(count):

a,b,c=0,1,0

while count>0:

yield c

a=b

b=c

c=a+b

fiboSeries = iter(fibo(7));

for i in range(7):

print(fiboSeries.\_\_next\_\_())

Output:

SEM\_2\PYTHON>py q\_5.py

0

1

1

2

3

5

8

**Question – 8**

Problem Statement:

Create a list of all the numbers up to N = 50 which are multiples of five using anonymous function.

Source Code:

foo = lambda value: True if value%5 == 0 else False

numList = list()

for i in range(51):

if foo(i):

numList.append(i)

print("Numbers multiple of 5 are: ")

for i in numList:

print(i, end=", ")

Output:

SEM\_2\PYTHON>py q\_8.py

Numbers multiple of 5 are:

0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50,

**Question – 10**

Problem Statement:

Filter out the odd squares using map, filter, list.

Source Code:

import math

odd = lambda x:True if x%2 != 0 else False

limit = int(input("Enter the range: "))

squres = []

for i in range(1, limit + 1):

squres.append(int(math.pow(i,2)))

oddSqures = []

oddSqures = filter(odd, squres)

for i in oddSqures:

print(i)

Output:

SEM\_2\PYTHON>py q\_10.py

Enter the range: 5

1

9

25

SEM\_2\PYTHON>py q\_10.py

Enter the range: 20

1

9

25

49

81

121

169

225

289

361

**Question – 13**

Problem Statement:

Write a code which yileds all terms of the geometric progression a, aq, aq­­2, aq3, ….

Source Code:

import time

import math

def getGPTerms(a,q):

x = 0

while(True):

temp = a\*(math.pow(x, q))

if (temp > 100000):

return False

yield temp

x += 1

start = time.time()

a = int(input("Enter the initial term :"))

q = int(input("Enter the common difference :"))

c = 0

n = int(input("Enter limit :"))

checkpoint = time.time()

terms = getGPTerms(a,q)

for i in terms:

if i and c <= n:

print(i)

c += 1

end = time.time()

print("Time taken for execution is: ",(end-start))

print("Time taken in loop :", (end-checkpoint))

Output:

SEM\_2\PYTHON>py q\_13.py

Enter the initial term :5

Enter the common difference :2

Enter limit :10

5.0

20.0

45.0

80.0

125.0

180.0

245.0

320.0

405.0

500.0

605.0

Time taken for execution is: 3.131488800048828

Time taken in loop : 0.0

SEM\_2\PYTHON>py q\_13.py

Enter the initial term :10

Enter the common difference :2

Enter limit :40

10.0

40.0

90.0

160.0

250.0

360.0

490.0

640.0

810.0

1000.0

1210.0

1440.0

1690.0

1960.0

2250.0

2560.0

2890.0

3240.0

3610.0

4000.0

4410.0

4840.0

5290.0

5760.0

6250.0

6760.0

7290.0

7840.0

8410.0

9000.0

9610.0

10240.0

10890.0

11560.0

12250.0

12960.0

13690.0

14440.0

15210.0

16000.0

16810.0

Time taken for execution is: 9.147000551223755

Time taken in loop : 0.06202530860900879

**Question – 14**

Problem Statement:

Search for palindrome and unique words in a text using class method and string methods.

Source Code:

class MyString:

userInput = ""

count = {}

def \_\_init\_\_(self, str):

self.userInput = str

def display(self):

print(self.userInput)

def isPalindrome(self, str):

if str == str[::-1]: return True

else: return False

def findAllUniquePalindor(self):

print("Palindrome words are: ")

words = self.userInput.split(" ")

for word in words:

if word in self.count:

self.count[word] += 1

else:

self.count[word] = 1

for i in self.count:

if self.count[i] == 1:

if self.isPalindrome(i):

print(i)

inputString = input("Enter a stirng: ")

#str = MyString("dad mad nayan hooh dad noice madam nayan hello")

str = MyString(inputString)

str.findAllUniquePalindor()

Output:

SEM\_2\PYTHON>py q\_14.py

Enter a stirng: hello world this is a test string

Palindrome words are:

a

SEM\_2\PYTHON>py q\_14.py

Enter a stirng: dad hello nayan madam world

Palindrome words are:

dad

nayan

madam

**Question – 18**

Problem Statement:

Make a list of the largest or smallest of N items in a collection.

Source Code:

def find\_K\_Largest(numList, k):

items = []

items = numList.copy()

while k > 0:

m = max(items)

yield m

items.remove(m)

k -= 1

listSize = int(input("Enter the size of the list: "))

numList = []

print("Enter elements: ")

for i in range(1, listSize + 1):

numList.append(int(input("")))

k = int(input("Enter k: "))

maxList = iter(find\_K\_Largest(numList, k))

print("Max K'th elemets are: ")

for i in range(k):

print(maxList.\_\_next\_\_(), end=", ")

Output:

SEM\_2\PYTHON>py q\_18.py

Enter the size of the list: 5

Enter elements:

1

5

3

4

2

Enter k: 2

Max K'th elemets are:

5, 4,

SEM\_2\PYTHON>py q\_18.py

Enter the size of the list: 8

Enter elements:

10

54

34

67

23

98

45

34

Enter k: 4

Max K'th elemets are:

98, 67, 54, 45,