MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY

«KHARKIV POLYTECHNIC INSTITUTE»

Department of Software Engineering and Management Information Technologies

Report from lab № 6

discipline «Fundamentals of python»

Executed by: Student of group 1.КН.201.8e1

CHUKWU IRELE AMORIN

Checked by: Senior lecturer

Kharkiv

2019

***Laboratory work 6***

***Loop “while”***

*1. List of squares*

For a given integer *N*, print all squares of natural numbers not exceeding *N* in ascending order.

*2. "Minimum divisor"*

Given an integer not less than 2. Print its smallest natural divisor other than 1.

*3. "Power of two"*

For a given natural number *N*, find the largest integer power of two not exceeding *N*. Derive the exponent and the power itself.

The operation of raising to a power cannot be used!

*4. "Morning running"*

On the first day, the athlete ran *x* kilometers, and then every day he increased the mileage by 10% from the previous value. Given the number *y*, determine the number of the day on which the athlete’s mileage will be at least *y* kilometers.

The program receives real numbers *x* and *y* as input and must output a single positive integer.

*5. "Sequence Length"*

The program receives a sequence of non-negative integers as an input, each number is written on a separate line. The sequence ends with the number 0, upon reading which the program should finish its work and output the number of members of the sequence (not counting the final number 0). The numbers following the number 0 do not need to be read.

*6. "Sum of sequence"*

Determine the sum of all elements of the sequence ending with the number 0. In this and in all the following problems, the numbers following the first zero do not need to be taken into account.

*7. "The average value of the sequence"*

Determine the average value of all elements of the sequence ending in number 0.

*8. “Maximum Sequence”*

The sequence consists of natural numbers and ends with the number 0. Determine the value of the largest element of the sequence.

*9. "Sequence Maximum Index"*

The sequence consists of natural numbers and ends with the number 0. Determine the index of the largest element of the sequence. If there are several largest elements, print the index of the first of them. Numbering of elements starts from zero.

*10. "The number of even elements of the sequence"*

Determine the number of even elements in the sequence ending with the number 0.

*11. "The number of elements that are larger than the previous"*

The sequence consists of natural numbers and ends with the number 0. Determine how many elements of this sequence are larger than the previous element.

*12. "Second Maximum"*

The sequence consists of various natural numbers and ends with the number 0. Determine the value of the second largest element in this sequence. It is guaranteed that there are at least two elements in the sequence.

*13. "The number of elements equal to the maximum"*

The sequence consists of natural numbers and ends with the number 0. Determine how many elements of this sequence are equal to its largest element.

*14. Fibonacci numbers*

The Fibonacci sequence is defined as follows:

*φ0* = 0, *φ1* = 1, *φn* = *φn* − 1 + *φn* − 2.

Given the number *n*, determine the nth Fibonacci number *φn*.

This problem can also be solved with the for loop.

*15. Number of Fibonacci Number*

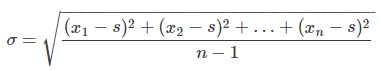
Given a positive integer *A*. Determine what the Fibonacci number is, that is, output a number *n* such that *φn = A*. If *A* is not a Fibonacci number, print the number -1.

*16. "The maximum number of consecutive equal elements"*

Given a sequence of natural numbers, ending with the number 0. Determine the largest number of consecutive elements of this sequence are equal to each other.

*17. "Standard deviation"*

A sequence of positive integers *x1*, *x2*, ..., *xn* is given. The standard deviation is called the value



where is the arithmetic mean of the sequence.

Determine the standard deviation for a given sequence of natural numbers ending in 0.

import math as m

#1

def squares():

    n = int(input("enter the value of N :"))

    sq = 0

    i = 1

    while n >= sq:

        sq = i\*\*2

        if sq <= n:

            print(sq)

        i += 1

#2

def smallestDivisor(n):

    # if divisible by 2

    if (n % 2 == 0):

        return 2;

    # iterate from 3 to sqrt(n)

    i = 3;

    while(i \* i <= n):

        if (n % i == 0):

            return i;

        i += 2;

    return n;

def divi():

    print("the smallest divider is : " ,smallestDivisor(int(input("enter the number : "))))

#3

def ptwo():

    N = int(input("input the value of n : "))

    n =1

    count = 0

    while N > n:

        if (n \* 2) < N :

            n = n \* 2

            count += 1

        else:

            break

    print("the exponet : ", count)

    print("the power is : ",n)

#4

def kilo():

    x = int(input("enter the value of x : "))

    y = int(input("enter the value of y kilometers : "))

    days = 1

    covered = x

    while  covered <= y:

        x =  x + (x \* (10/100))

        covered += x

        days += 1

    print("the athlete is on is {0} day" .format(days))

#5

def sequence1():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    count =0

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            count +=1

        i += 1

    print("count of members of the sequence is : ",count)

#6

def sumseq():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    count =0

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            count += int(seq[i])

        i += 1

    print("sum of members of the sequence is : ",count)

#7

def ave():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    count =0

    sumt =0

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            sumt += int(seq[i])

            count +=1

        i += 1

    ave = sumt /count

    print("average of members of the sequence is : ",ave)

#8

def maxi():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    maxm =int(seq[i])

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            if maxm < int(seq[i]):

                maxm = int(seq[i])

        i += 1

    print("largest of members of the sequence is : ",maxm)

#9

def maxIN():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    maxm =i

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            if int(seq[maxm]) < int(seq[i]):

                maxm = i

        i += 1

    print("index largest of members of the sequence is : ",maxm)

#10

def even():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    count =0

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            if int(seq[i]) % 2 == 0:

                count += 1

        i += 1

    print("sum of even  members of the sequence is : ",count)

#11

def prvmax():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 1

    j = 0

    count =0

    while seq[i] != '0' and i < length:

        prev =int(seq[j])

        if seq[i] != '0':

            if prev < int(seq[i]):

                count += 1

        i += 1

        j += 1

    print("the count of members larger than previous : ",count )

#12

def secmax():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    maxm =int(seq[i])

    secmaxm =int(seq[i])

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            if maxm < int(seq[i]):

                maxm = int(seq[i])

            if secmaxm < int(seq[i])  < maxm :

                secmaxm = int(seq[i])

        i += 1

    print("second largest of members of the sequence is : ",secmaxm)

#13

def equamax():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    maxm =int(seq[i])

    count = 0

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            if maxm <= int(seq[i]):

                if maxm < int(seq[i]):

                    count = 0

                count += 1

                maxm = int(seq[i])

        i += 1

    print("largest of members of the sequence is : ",count)

#14

def Fibonacci(n):

    if n<0:

        print("Incorrect input")

    # First Fibonacci number is 0

    elif n==1:

        return 0

    # Second Fibonacci number is 1

    elif n==2:

        return 1

    else:

        return Fibonacci(n-1)+Fibonacci(n-2)

def fib():

    n = int(input("enter n :"))

    print(Fibonacci(n))

#15

def  frt():

    # Program to display the Fibonacci sequence up to n-th term

    val = int(input("what is the value  "))

    # first two terms

    n1, n2 = 0, 1

    count = 0

    nterms = 1

    while n1 <= val:

        # check if the number of terms is valid

        if nterms <= 0:

            print("Please enter a positive integer")

        elif nterms == 1:

            #print("Fibonacci sequence upto",nterms,":")

            #print(n1)

            pass

        else:

            #print("Fibonacci sequence:")

            pass

        while count < nterms and n1 <= val:

            # print(n1)

            nth = n1 + n2

            # update values

            n1 = n2

            n2 = nth

            count += 1

            nterms += 1

    if Fibonacci(nterms -1) == val:

        print("the Fibonacci number is : ", nterms-1)

    else:

        print(-1)

#16

def maxcon():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    curr =int(seq[i])

    prev = int(seq[i])

    count = 0

    maxcount =0

    while seq[i] != '0' and i < length:

        if seq[i] != '0':

            if curr != int(seq[i]):

                count  = 0

            count += 1

            if curr == int(seq[i]):

                if count >= maxcount:

                    maxcount = count

            curr =  int(seq[i])

        i += 1

    print("most number consecutive of members of the sequence is : ",count)

#17

def drivative():

    seq = []

    value = 0

    while value != ('y' or 'Y'):

        value = input("enter a value  , enter y to stop : ")

        if value != ('y' or 'Y'):

            seq.append(value)

    length =  len(seq)

    i = 0

    s= 0

    count = 0

    su = 0

    T = 0;

    while seq[i] != '0' and i < length :

        su += int(seq[i])

        i += 1

        count += 1

    s = su / count

    i = 0

    while seq[i] != '0' and i < length:

        T += (int(seq[i]) - s)\*\*2

        i += 1

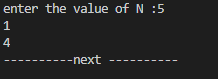
    total = m.sqrt( T / count - 1)

    print("this is the total of drivaton : ", total)

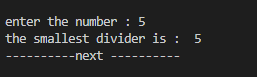
if \_\_name\_\_ == "\_\_main\_\_":

    drivative()

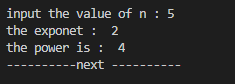
No1:



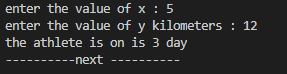
No2:



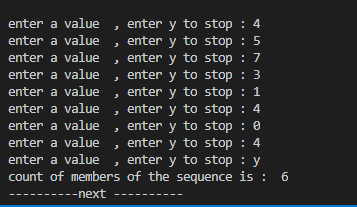
No3:



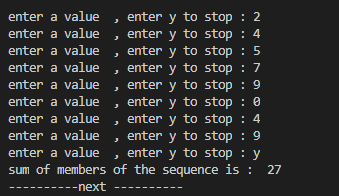
No4:



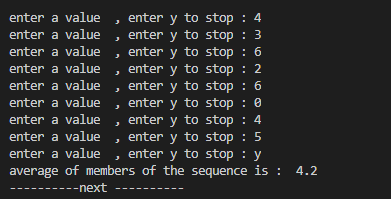
No5:



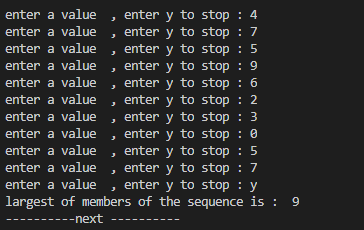
No6:



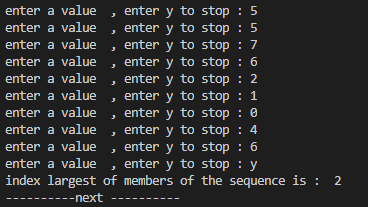
No7:



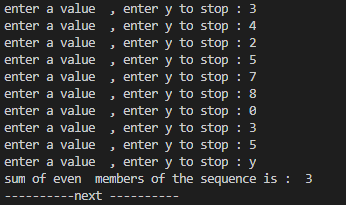
No8:



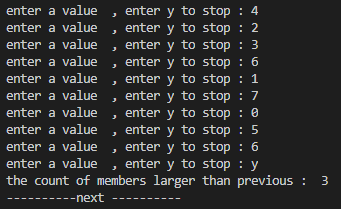
No9:



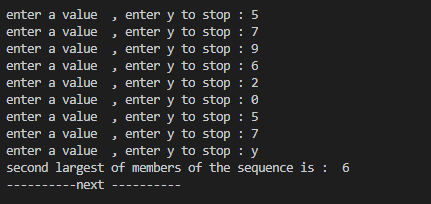
No10:



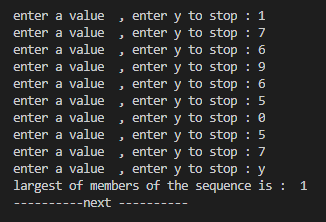
No11:



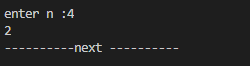
No12:



No13:



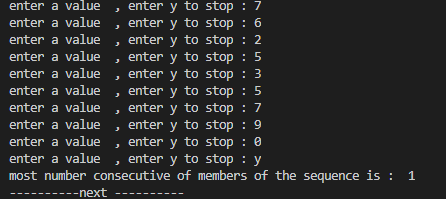
No14:



No15:



No16:



No17

