|  |
| --- |
| **Faculty In charge: JANANI T** |
| **Class No.: VL2021220107101** |

Logo

Description automatically generated

**School of Computer Science and Engineering (SCOPE)**

**Vellore Institute of Technology**

**Vellore.**

|  |
| --- |
|  |
| **BCSE101E - Computer Programming: Python** |
|  |
| **Digital Footprint** |

BCSE101E - Computer Programming: Python

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| Ex.No. | Title | Date | Page No. |
| 1 | **M12\_CSQ1** | 19/10/2021 | 2 |
| 2 | **M12\_CSQ2** | 19/10/2021 | 3 |
| 3 | **M12\_CSQ3** | 19/10/2021 | 5 |
| 4 | **M12\_CSQ4** | 19/10/2021 | 8 |
| 5 | **M12\_CSQ5** | 19/10/2021 | 10 |
| 6 | **M12\_CSQ6** | 19/10/2021 | 12 |
| 7 | **M12\_CSQ7** | 19/10/2021 | 13 |
| 8 | **M12\_CSQ8** | 19/10/2021 | 14 |
| 9 | **M12\_CSQ9** | 19/10/2021 | 15 |
| 10 | **M12\_CSQ10** | 19/10/2021 | 16 |
| 11 | **M3\_CSQ1** | 19/10/2021 | 17 |
| 12 | **M3\_CSQ2** | 19/10/2021 | 18 |
| 13 | **M3\_CSQ3** | 19/10/2021 | 19 |
| 14 | **M3\_CSQ4** | 20/19/2021 | 21 |
| 15 | **M3\_CSQ5** | 20/10/2021 | 23 |
| 16 | **M4\_CSQ1** | 16/11/2021 | 24 |
| 17 | **M4\_CSQ2** | 16/11/2021 | 26 |
| 18 | **M4\_CSQ3** | 16/11/2021 | 28 |
| 19 | **M4\_CSQ4** | 17/11/2021 | 30 |
| 20 | **M4\_CSQ5** | 18/11/2021 | 32 |
| 21 | **M5\_CSQ1** | 23/11/2021 | 34 |
| 22 | **M5\_CSQ2** | 26/11/2021 | 35 |
| 23 | **M5\_CSQ3** | 26/11/2021 | 36 |
| 24 | **M5\_CSQ4** | 30/11/2021 | 38 |
| 25 | **M5\_CSQ5** | 30/11/2021 | 39 |
| 26 | **M6\_CSQ1** | 8/12/2021 | 40 |
| 27 | **M6\_CSQ2** | 10/12/2021 | 41 |
| 28 | **M6\_CSQ3** | 22/12/2021 | 42 |
| 29 | **M6\_CSQ4** | 31/12/2021 | 44 |
| 30 | **M6\_CSQ5** | 31/12/2021 | 46 |
| 31 | **M7\_CSQ1** | 4/1/2022 | 48 |
| 32 | **M7\_CSQ2** | 5/1/2022 | 49 |
| 33 | **M7\_CSQ3** | 5/1/2022 | 50 |
| 34 | **M7\_CSQ4** | 7/1/2022 | 51 |
| 35 | **M7\_CSQ5** | 7/1/2022 | 52 |
| 36 | **M7\_CSQ6** | 7/1/2022 | 54 |

|  |
| --- |
| **Signature of the student (Digital)** |
| **Abhinav Dinesh Srivatsa** |

## [Ex. No. M12\_CSQ1]

### AIM

To calculate the age of a housefly in seconds, given the number of days the housefly lived.

### Algorithm / Pseudocode

Read NoDays

Calculate NoSeconds = NoDays \* 24 \* 3600

Display NoSeconds

### Table Description automatically generated with low confidenceFlowchart

### Program Code

days = int(input("Enter days the fly has been alive: "))

seconds = days \* 24 \* 3600

print(f"The fly has lived for {seconds} seconds")

A picture containing text

Description automatically generated

**Output**

## [Ex. No. M12\_CSQ2]

### AIM

To calculate the total quantity of milk from ‘n’ farm.

### Algorithm / Pseudocode

Read NoFarms

Read L1, L2, …

Read M1, M2, …

Calculate TotalM = M1 + M2 + …

Calculate TotalL = L1 + L2 + …

Calculate TotalL = TotalL + TotalM / 1000

Calculate TotalM = TotalM % 1000

Display TotalL

Display TotalM

### Diagram, schematic Description automatically generatedFlowchart

### Program Code

farms = int(input("Enter number of farms: "))

litre\_sum = 0

milli\_sum = 0

for i in range(farms):

    milk = input(f"Farm {i + 1}: ").split(" ")

    litre\_sum += int(milk[0])

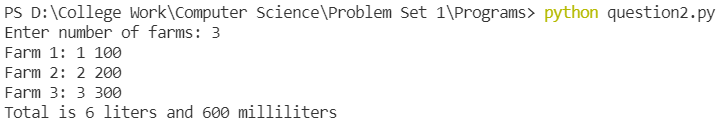
    milli\_sum += int(milk[1])

# carrying over the extra milliliters to liters

litre\_sum += int(milli\_sum / 1000)

milli\_sum %= 1000

print(f"Total is {litre\_sum} liters and {milli\_sum} milliliters")



### Output

## [Ex. No. M12\_CSQ3]

### AIM

To convert a two-digit number to its corresponding Roman numeral.

### Algorithm / Pseudocode

Read Number

Tens = Number // 10

Units = Number % 10

If Tens is between 0 and 4, then concatenate Tens amount of ‘X’

Else if Tens is 4, concatenate ‘XL’

Else if Tens is between 4 and 9, then concatenate ‘L’ + (Tens – 5) amount of ‘X’

Else if Tens is 9, then concatenate ‘XC’

Else if Tens is 10, then concatenate ‘C’

If Units is between 0 and 4, then concatenate Units amount of ‘I’

Else if Units is 4, concatenate ‘IV’

Else if Units is between 4 and 9, then concatenate ‘V’ + (Tens – 5) amount of ‘I’

Else if Units is 9, then concatenate ‘IX’

Display Roman

### Diagram Description automatically generatedFlowchart

### Program Code

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

roman = ""

tens = number // 10

units = number % 10

if tens > 0 and tens < 4:

    roman += (tens \* 'X')

elif number / 10 == 4:

    roman += 'XL'

elif tens > 4 and tens < 9:

    roman += 'L' + ((tens - 5) \* 'X')

elif tens == 9:

    roman += 'XC'

elif tens == 10:

    roman += 'C'

if units > 0 and units < 4:

    roman += (units \* 'I')

elif units == 4:

    roman += 'IV'

elif units > 4 and units < 9:

    roman += 'V' + ((units - 5) \* 'I')

elif units == 9:

    roman += 'IX'

print(f"{number} in Roman numerals is {roman}")



### Output

## [Ex. No. M12\_CSQ4]

### AIM

To find the sum of the real and imaginary parts of two complex numbers.

### Algorithm / Pseudocode

Read Vector1, Vector2

Calculate RealSum = Get real parts of the vectors and sum

Calculate ImaginarySum = Get imaginary parts of the vectors and sum

Display 'Real part is ' RealSum

Diagram

Description automatically generatedDisplay 'Imaginary part is ' ImaginarySum

**Flowchart**

### Program Code

vector1 = complex(input("Enter vector 1: "))

vector2 = complex(input("Enter vector 2: "))

real\_sum = vector1.real + vector2.real

imaginary\_sum = vector1.imag + vector2.imag

print(f"Real part is: {int(real\_sum)}")

print(f"Imaginary part is: {int(imaginary\_sum)}")

Text

Description automatically generated with low confidence

**Output**

## [Ex. No. M12\_CSQ5]

### AIM

To convert a given integer to binary, octal and hexadecimal.

### Algorithm / Pseudocode

Read Number

Calculate Bin = bin(Number)

Calculate Oct = oct(Number)

Calculate Hex = hex(number)

Chart, box and whisker chart

Description automatically generatedDisplay Bin, Oct, Hex

**Flowchart**

### Program Code

num = int(input())

print(bin(num), oct(num), hex(num))

****

**Output**

## [Ex. No. M12\_CSQ6]

### AIM

To calculate the area of a triangle given its three sides.

### Algorithm / Pseudocode

Read S1, S2, S3

Calculate Semiper = (S1 + S2 + S3) / 2

Calculate Area = (Semiper \* (Semi per – S1) \* (Semiper – S2) \* (Semiper – S3)) ^ 0.5

Display Area

### Program Code

s1 = int(input())

s2 = int(input())

s3 = int(input())

semi\_per = (s1 + s2 + s3) / 2

area = (semi\_per \* (semi\_per - s1) \* (semi\_per - s2) \* (semi\_per - s3)) \*\* 0.5

print("{:.2f}".format(area))

A picture containing text

Description automatically generated

**Output**

## [Ex. No. M12\_CSQ7]

### AIM

To convert a string to an integer.

### Algorithm / Pseudocode

Read String

Calculate Integer = int(Number)

Display Integer

### Program Code

string = input()

integer = int(string)

print(integer)



**Output**

## [Ex. No. M12\_CSQ8]

### AIM

To convert an integer to a string.

### Algorithm / Pseudocode

Read Integer

Calculate String = str(Integer)

Display String

### Program Code

integer = int(input())

string = str(integer)

print(string)

****

**Output**

## [Ex. No. M12\_CSQ9]

### AIM

To convert Fahrenheit to Celsius using a lambda function.

### Algorithm / Pseudocode

Celsius = lambda function of Fahrenheit

Read Fahrenheit

Display Celsius(Fahrenheit)

### Program Code

c = lambda f: (f - 32) \* 5 / 9

f = float(input())

print("{:.2f}".format(c(f)))



**Output**

## [Ex. No. M12\_CSQ10]

### AIM

To swap the values of two given variables.

### Algorithm / Pseudocode

Read Number1

Read Number2

Number1, Number2 = Number2, Number1

Display Number1, Number2

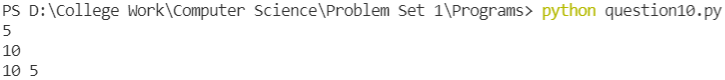
### Program Code

number1 = input()

number2 = input()

number1, number2 = number2, number1

print(number1, number2)



**Output**

## [Ex. No. M3\_CSQ1]

### AIM

To calculate the amount of tax given a yearly salary.

### Algorithm / Pseudocode

Read Income

If Income > 1000000, then display Income \* 0.04

Else if Income > 500000, then display Income \* 0.02

Else display 0

### Program Code

income = float(input())

if income > 1000000:

    print(int(income \* 0.04))

elif income > 500000:

    print(int(income \* 0.02))

else:

    print(0)



**Output**

## [Ex. No. M3\_CSQ2]

### AIM

To calculate the price of a taxi ride based on the amount of distance driven.

### Algorithm / Pseudocode

Read Distance

If Distance < 0, then display ‘Enter Positive Number Only’

Else,

If Distance <= 10, then display Distance \* 15

Else if Distance <= 90, then display 150 + Distance \* 8

Else display 150 + 720 + Distance \* 6

### Program Code

s = int(input())

if s < 0:

    print('Enter Positive Number Only')

else:

    if s <= 10:

        print(s \* 15)

    elif s <= 90:

        print(150 + (s - 10) \* 8)

    else:

        print(150 + 720 + (s - 100) \* 6)



**Output**

## [Ex. No. M3\_CSQ3]

### AIM

To calculate the amount of marks a student scores based on their original marks.

### Algorithm / Pseudocode

Read Marks

If Marks <= 0, then display ‘Enter appropriate Mark’

Else,

Read Session

If marks >= 80, then

If Session is ‘Theory’, then display Marks \* 1.08

If Session is ‘Lab’, then display Marks \* 1.06

If marks >= 60, then

If Session is ‘Theory’, then display Marks \* 1.06

If Session is ‘Lab’, then display Marks \* 1.04

If marks >= 40, then

If Session is ‘Theory’, then display Marks \* 1.04

If Session is ‘Lab’, then display Marks \* 1.02

Else,

If Session is ‘Theory’, then display Marks

If Session is ‘Lab’, then display Marks

### Program Code

marks = float(input())

if marks <= 0:

    print('Enter appropriate Mark')

else:

    session = input()

    if marks >= 80:

        if session == 'T':

            print('{:.2f}'.format(marks \* 1.08))

        if session == 'L':

            print('{:.2f}'.format(marks \* 1.06))

    elif marks >= 60:

        if session == 'T':

            print('{:.2f}'.format(marks \* 1.06))

        if session == 'L':

            print('{:.2f}'.format(marks \* 1.04))

    elif marks >= 40:

        if session == 'T':

            print('{:.2f}'.format(marks \* 1.04))

        if session == 'L':

            print('{:.2f}'.format(marks \* 1.02))

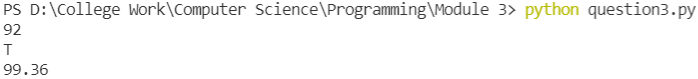
    else:

        if session == 'T':

            print('{:.2f}'.format(marks))

        if session == 'L':

            print('{:.2f}'.format(marks))



**Output**

## [Ex. No. M3\_CSQ4]

### AIM

To display all alternate number factorials below the number given.

### Algorithm / Pseudocode

Read Number

If Number <= 0, then display ‘Enter only positive number’

Else,

If Number is even, then Start = 2

Else Start = 1

While Start <= Number

Find factorial of Start and assign to Fact

Increment Start by 2

Display Start, Fact

### Program Code

num = int(input())

if num < 0:

    print("Enter only positive number")

else:

    if num % 2 == 0:

        i = 2

    else:

        i = 1

    while i <= num:

        fact = 1

        j = 2

        while j <= i:

            fact \*= j

            j += 1

        print(i, fact)

        i += 2

**A picture containing text

Description automatically generatedOutput**

## [Ex. No. M3\_CSQ5]

### AIM

To find the sum of the digits of a given number as a single digit.

### Algorithm / Pseudocode

Read Number

While Number % 10 != 0

Tens = Number // 10

Units = Number % 10

Number = Tens + Units

Display Number

### Program Code

num = int(input())

while num // 10 != 0:

    num = num // 10 + num % 10

****print(num)

**Output**

## [Ex. No. M4\_CSQ1]

### AIM

To find the point in a list where the right elements and left elements sums are equal.

### Algorithm / Pseudocode

Read Number

Create list Numbers

Read inputs in for loop and assign to Numbers

Loop I from 1 to Number – 1

Initialise Left as the left side in Numbers

Initialise Right as the right side in Numbers

If the sum of Left = sum of Right

Display I (Loop Variable)

If nothing has been displayed

Display 0

### Program Code

n = int(input())

numbers = []

for i in range(n):

    numbers.append(int(input()))

flag = False

for i in range(1, n - 1):

    left = numbers[:i]

    right = numbers[i + 1:]

    if sum(left) == sum(right):

        flag = True

        print(i)

if(not flag):

    print(0)

**Background pattern

Description automatically generated with low confidenceOutput**

## [Ex. No. M4\_CSQ2]

### AIM

To find the answer of an RPN (postfix) notation expression.

### Algorithm / Pseudocode

Read Number

Create list RPN

Read inputs in for loop and assign to RPN, as integer if number and string if character

Loop while RPN’s length is not 1

Initialise Triple as the i­­­­­­th­­, i+1th and i+2th elements in RPN

If the third element is an integer, then take the next triplet and continue

Else,

Initialise Op = the third element in Triple

If Op = ‘/’, then divide the first and second elements of Triple

If Op = ‘\*’, then multiply the first and second elements of Triple

If Op = ‘+’, then add the first and second elements of Triple

If Op = ‘-’, then subtract the first and second elements of Triple

Replace the corresponding triple in with the calculated number

Start from the first index again and continue

Print the first index of RPN

### Program Code

n = int(input())

rpn = []

for i in range(n):

    inp = input()

    try:

        inp = int(inp)

    except:

        pass

    finally:

        rpn.append(inp)

i = 0

while len(rpn) != 1:

    triple = rpn[i: i + 3]

    if type(triple[2]) is int:

        i += 1

        continue

    else:

        op = triple[2]

        if op == '/':

            sum = triple[0] / triple[1]

        elif op == '\*':

            sum = triple[0] \* triple[1]

        elif op == '+':

            sum = triple[0] + triple[1]

        elif op == '-':

            sum = triple[0] - triple[1]

        rpn[i: i+3] = [sum]

        i = 0

**Background pattern

Description automatically generated with low confidence**print(int(rpn[0]))

**Output**

## [Ex. No. M4\_CSQ3]

### AIM

To display a list with the primary key and the corresponding matching value from a list of lists.

### Algorithm / Pseudocode

Read Number

Create list Matrix

Read inputs in for loop and assign to a list New\_Mat, as integer if number and string if character

Add this New\_Mat to Matrix

Read Primary\_Index

Read Primary\_Value

Try converting Primary\_Value to string, if not continue

Loop through matrix with indices as i being row elements and j being column indices in i

If j = Primary\_Index and i[ j ] = Primary\_Value, then display i

### Program Code

n = int(input())

m = int(input())

matrix = []

for i in range(n):

    new\_mat = []

    for j in range(m):

        inp = input()

        try:

            inp = int(inp)

        except:

            pass

        finally:

            new\_mat.append(inp)

    matrix.append(new\_mat)

primary\_index = int(input())

primary\_value = input()

try:

    primary\_value = int(primary\_value)

except:

    pass

for i in matrix:

    for j in range(len(i)):

        if j == primary\_index and i[j] == primary\_value:

Text

Description automatically generated with low confidence            print(i)

**Output**

## [Ex. No. M4\_CSQ4]

### AIM

To display a tuple summarizing the costs per department entered.

### Algorithm / Pseudocode

Read Number

Initialise Costs as ()

Loop i from 0 to Number

Read No\_Items

Initialise Info as ()

Loop j from 0 to No\_Items and record values in Info

Initialise Sum as 0

Loop through Items and add all integers to sum

Add the department and cost to Costs

Display Costs

### Program Code

n = int(input())

costs = ()

for i in range(n):

    items = int(input())

    info = ()

    for j in range(items):

        info += (input(), )

    sum = 0

    for j in info:

        try:

            sum += int(j)

        except:

            pass

    if i == n - 1:

        costs += ((info[0], sum))

    else:

        costs += ((info[0], sum), )

Text

Description automatically generated with low confidenceprint(costs)

**Output**

## [Ex. No. M4\_CSQ5]

### AIM

To find which students failed in a particular subject.

### Algorithm / Pseudocode

Read No\_Students

Initialise Students as list

Loop i from 0 to No\_Students

Initialise Marks as dictionary

Read Subject and Marks and assign to dictionary Marks

Append the Marks to Students list

Display Students

Initialise Failed as dictionary

Initialise Total to calculate total failures

Loop through Students and check if any score is less than 50

If true, then add the student and subject to Failed and increment Total

Display Failed and Total

### Program Code

no\_students = int(input())

students = []

for i in range(no\_students):

    marks = {}

    no\_marks = int(input())

    for j in range(no\_marks):

        m = input()

        marks[m] = int(input())

    students.append(marks)

print(students)

failed = {}

total = 0

#getting fails

for i in students:

    flag = True

    for j in i:

        if j not in failed.keys():

            failed[j] = 0

        if i[j] < 50:

            if(flag):

                total += 1

                flag = False

            failed[j] += 1

for i in failed:

    print(i)

    print(failed[i])

print(total)

**Output**

Graphical user interface, text, application

Description automatically generated

## [Ex. No. M5\_CSQ1]

### AIM

To swap a string uppercase letters to lower case and vice versa.

### Algorithm / Pseudocode

Read Sentence

Loop through characters in Sentence as i

If i is lower case, then display it’s upper case

Else if i is upper case, then display it’s lower case

Else, display i

### Program Code

sen = input()

for i in sen:

    if i.islower():

        print(i.upper(), end = '')

    elif i.upper():

        print(i.lower(), end = '')

    else:

        print(i, end = '')

**Output**

## [Ex. No. M5\_CSQ2]

### AIM

To check if a string is a good word or a bad word. Good words have no repeat characters, while bad words do.

### Algorithm / Pseudocode

Read Word and convert it to lower case

Initialise Letters as the characters of Word

Initialise Verdict as ‘GOOD’

Loop from 0 to length of Letters as i

Initialize Letter1 as Letters[i]

Loop Letter2 as Letters past Letter1

If Letter1 equals Letter2, the assign verdict as’BAD’

Break

Display Verdict

### Program Code

word = input().lower()

letters = list(word)

verdict = 'GOOD'

for i in range(len(letters)):

    letter1 = letters[i]

    for letter2 in letters[i + 1:]:

        if letter1 == letter2:

            verdict = 'BAD'

            break

print(verdict)

**Output**

## [Ex. No. M5\_CSQ3]

### AIM

To check if a word exists in a sentence.

### Algorithm / Pseudocode

Read Sentence and convert it to lower case

Read Word and convert it to lower case

Initialise Length as length of Word

Initialise Found as False

Loop through 0 to length of Sentence as i

Initialise Replace as replacing Sentence of i to i + Length with Word

If Replace and Sentence are equal, then

Assign Found as True

Display i

Display i + Length

Break

If Found is False, the display ‘Not Found’

### Program Code

sen = input().lower()

word = input().lower()

length = len(word)

found = False

for i in range(len(sen)):

    if sen.replace(sen[i:i + length], word) == sen:

        found = True

        print(i)

        print(i + length)

        break

if not found:

    print('Not Found')A picture containing text

Description automatically generated

**Output**

## [Ex. No. M5\_CSQ4]

### AIM

To find all the students names and marks from an input string.

### Algorithm / Pseudocode

Read Sentence

Initialise Words by taking out commas and splitting Sentence by spaces

Initialise Students and Marks as a list

Loop through Word in Words

If Word starts with a capital letter, then append Word to Students

Else if Word is a number, then append Word to Marks

Display Marks

Display Students

### Program Code

import re

sen = input()

words = sen.replace(',', '').split(' ')

students = []

marks = []

for word in words:

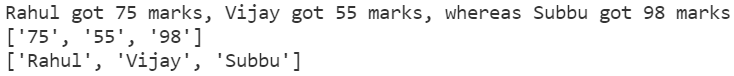
    if re.match('^[A-Z]', word):

        students.append(word)

    elif re.match('[0-9]', word):

        marks.append(word)

print(marks)

print(students)

**Output**

## [Ex. No. M5\_CSQ5]

### AIM

To find all the words starting with vowels in a sentence.

### Algorithm / Pseudocode

Read Sentence

Initialise Words by splitting Sentence by spaces

Loop through Word in Words

If Word starts with a vowel (a, e, i, o, u, A, E, I, O, U), then display Word

### Program Code

import re

sen = input()

words = sen.split(' ')

for word in words:

    if re.match('^[aeiouAEIOU]', word):

        print(word)

**Output**

A picture containing letter

Description automatically generated

## [Ex. No. M6\_CSQ1]

### AIM

To find the substring of a word relative to a position given.

### Algorithm / Pseudocode

Substring\_Position(String S1, String S2, Integer N)

Loop I from 0 to length of S1

If slice of S1 of I to I + length of S2 equals S2, then return I – N

Return ‘NotAvailable’

Read S1

Read S2

Read position N

Display Substring\_Position(S1, S2, N)

### Program Code

def substring\_position(s1, s2, n):

    for i in range(len(s1)):

        if s1[i : i + len(s2)] == s2:

            return i - n

    return 'NotAvailable'

s1 = input()

s2 = input()

n = int(input())

A picture containing text

Description automatically generatedprint(substring\_position(s1, s2, n))

**Output**

## [Ex. No. M6\_CSQ2]

### AIM

To find the number of chicke3ns and rabbits being given the number of heads and tails.

### Algorithm / Pseudocode

Get\_Animals(Integer M, Integer N)

Return N/2 – M, 2M – N/2

Read Count as a tuple

Initialise M = Count[0]

Initialise N = Count[1]

Display Get\_Animals(M, N)

### Program Code

def get\_animals(m, n):

    # equations:

    #    x  +  y = m

    #    4x + 2y = n

    #

    # =>  x      = n/2 - m

    # =>  y      = m - x = 2m - n/2

    return n//2 - m, 2\*m - n//2

count = eval(input())

m = count[0]

n = count[1]

Text

Description automatically generatedprint(get\_animals(m, n))

**Output**

## [Ex. No. M6\_CSQ3]

### AIM

To find the mean marks of students and categorise them into groups of scoring more, equal, or less to the average.

### Algorithm / Pseudocode

Scores(Dictionary Student\_Marks)

Calculate mean marks of Student\_Marks and assign to Mean\_Marks

Initialise lists Above, Mean, Below

Loop through items in Student\_Marks with I as students and J as marks

If J > Mean\_Marks, then append I to Above

Else if J = Mean\_Marks, append I to Mean

Else, append I to Below

Return Mean, Above, Below

Read Student\_Marks

Mean, Above, Below = Scores(Student\_Marks)

If Mean contains more than 0 elements, then print Mean joined by commas

If Above contains more than 0 elements, then print Above joined by commas

If Below contains more than 0 elements, then print Below joined by commas

### Program Code

def scores(student\_marks):

    marks = student\_marks.values()

    mean\_marks = sum(marks) // len(marks)

    above = []

    mean = []

    below = []

    for i, j in student\_marks.items():

        if j > mean\_marks:

            above.append(i)

        elif j == mean\_marks:

            mean.append(i)

        else:

            below.append(i)

    return mean, above, below

student\_marks = eval(input())

mean, above, below = scores(student\_marks)

if len(mean) != 0:

    print(', '.join(mean))

if len(above) != 0:

    print(', '.join(above))

if len(below) != 0:

A picture containing text

Description automatically generated    print(', '.join(below))

**Output**

## [Ex. No. M6\_CSQ4]

### AIM

To create a quiz with two files – Questions.txt and Answers.txt

### Algorithm / Pseudocode

Initialise Questions by reading Questions.txt

Initialise Answers by reading Answers.txt

Display Questions

Read User\_Answers as input

Initialise I as 0

Initialise Sum as 0

Loop through lines in Answer as Line

If the corresponding User\_Answer and Line are the same, then add 50 to Sum

Increment I by 1

Display Sum

### Program Code

questions = open('Questions.txt', 'r')

answers = open('Answers.txt', 'r')

print(questions.read())

user\_ans = input().split(' ')

i = 0

sum = 0

for line in answers.readlines():

    if user\_ans[i] == line.strip()[-1].lower():

        sum += 50

    i += 1

print(sum)

Text

Description automatically generated**Output**

## [Ex. No. M6\_CSQ5]

### AIM

To find the number of unique words in a sentence

### Algorithm / Pseudocode

Duplicate\_Characters(String Sen):

Initialise Words by splitting Sen by spaces

Initialise Unique\_Words as list

Loop through Words with Word

If Word is not in Unique\_Words, then add Word to Unique\_Words

Initialise Word\_Count as dict

Loop through a sorted Unique\_Words as Unique

Initialise Count as 0

Loop through Words as Word

If Word = Unique, then increment count by 1

Word\_Count of Unique = Count

Return Word\_Count

Read Sen as input

Initialise Dict as Duplicate\_Characters(Sen)

Loop through Dict and print the word counts

### Program Code

def duplicate\_characters(sen):

    words = sen.split(' ')

    unique\_words = []

    for word in words:

        if word not in unique\_words:

            unique\_words.append(word)

    word\_count = {}

    for unique in sorted(unique\_words):

        count = 0

        for word in words:

            if unique == word:

                count += 1

        word\_count[unique] = count

        count = 0

    return word\_count

sen = input()

dict = duplicate\_characters(sen)

for i, j in dict.items():

Text

Description automatically generated    print(f'{i}:{j}')

**Output**

## [Ex. No. M7\_CSQ1]

### AIM

To find the mean, standard deviation and variance of a list given

### Algorithm / Pseudocode

Import numpy as NP

Initialise Array as NP array of evaluated input

Calculate Mean as NP mean of Array

Calculate Standard\_Deviation as NP std of Array

Calculate Variance as NP var of Array

Display Mean

Display Standard\_Deviation

Display Variance

### Program Code

import numpy as np

array = np.array(eval(input()))

mean = np.mean(array)

std\_dev = np.std(array)

var = np.var(array)

print(mean)

print(std\_dev)

A picture containing calendar

Description automatically generatedprint(var)

**Output**

## [Ex. No. M7\_CSQ2]

### AIM

To sort an array by class then height for given data

### Algorithm / Pseudocode

Import numpy as NP

Initialise Students as given data

Initialise Data\_Types as the data types of students

Reinitialise Students as a structured array of students with datatypes as Data\_Types

Reinitialise Students as NP sort of Students, while ordering by class and then height

Display Students

### Program Code

import numpy as np

students = [('john', 164.5, 'XA'), ('mark', 188.3, 'XB'), ('zack', 178.7, 'XB')]

data\_types = [('name', 'S10'), ('height', float), ('class', 'S4')]

students = np.array(students, dtype = data\_types)

students = np.sort(students, order = ['class', 'height'])

print(students)

**Output**

## [Ex. No. M7\_CSQ3]

### AIM

To develop a NumPy program to sort the student id with increasing height of the students from given students id and height. Print the integer indices that describes the sort order by multiple columns and the sorted data

### Algorithm / Pseudocode

Import numpy as NP

Initialise Names as NP array of random names

Initialise Heights as NP array of random heights with same length as Names

Initialise Sort as NP lexsort of (Names, Heights)

Loop through Sort as I

Display Names of I, Heights of I

### Program Code

import numpy as np

names = np.array(['Mark', 'Sam', 'Henry', 'John', 'Xander', 'Hornato', 'Wade', 'Xenos', 'Barry', 'Rich'])

heights = np.array([145.5, 193.2, 178.4, 135.9, 169.8, 159.3, 178.4, 183.2, 176.7, 165.3])

sort = np.lexsort((names, heights))

for i in sort:

    print(f'{names[i]}, {heights[i]}')

**Text

Description automatically generatedOutput**

## [Ex. No. M7\_CSQ4]

### AIM

To design a Pandas program to join the two given data frames along rows and assign all data

### Algorithm / Pseudocode

Import pandas as PD

Initialise Data1 as dictionary of random data

Initialise Data2 as dictionary of random data

Initialise DF1 as PD dataframe of Data1

Initialise DF2 as PD dataframe of Data2

Initialise DF3 as PD concat of [DF1, DF2]

Display DF3

### Program Code

import pandas as pd

data1 = {'regno': ['21BDS0342', '21BDS0343'], 'studname': ['ghi', 'jkl'], 'cgpa': [5.6, 10]}

data2 = {'regno': ['21BDS0340', '21BDS0341'], 'studname': ['abc', 'def'], 'cgpa': [8.5, 9.2]}

DF1 = pd.DataFrame(data1)

DF2 = pd.DataFrame(data2)

DF3 = pd.concat([DF1, DF2])

Text

Description automatically generated with medium confidenceprint(DF3)

**Output**

## [Ex. No. M7\_CSQ5]

### AIM

To write a Pandas program to convert all the string values to upper, lower cases in each pandas series. Also find the length of the string values

### Algorithm / Pseudocode

Import pandas as PD

Read Strings as input

Reinitialise Strings as Strings split by space

Reinitialise Strings as PD series of Strings

Set Strings’ name as ‘Text’

Initialise Upper as Strings’ strings as upper case

Set Upper’s name as ‘Upper Case’

Initialise Lower as Strings’ strings as lower case

Set Lower’s name as ‘Lower Case’

Initialise Length as Strings’ strings length

Set Length’s name as ‘Length’

Initialise DF as PD concat of [Strings, Upper, Lower, Length] concatenated horizontally

Display DF

### Program Code

import pandas as pd

strings = input()

strings = strings.split(' ')

strings = pd.Series(strings)

strings.name = 'Text'

upper = strings.str.upper()

upper.name = 'Upper Case'

lower = strings.str.lower()

lower.name ='Lower Case'

length = strings.str.len()

length.name = 'Length'

DF = pd.concat([strings, upper, lower, length], axis = 1)

Text

Description automatically generatedprint(DF)

Table

Description automatically generated**Output**

## [Ex. No. M7\_CSQ6]

### AIM

To create a pandas program to find

* Datetime object for Jan 12, 2022
* Specific date and time of 10:00 pm
* Local date and time
* A date without time
* Current date
* Time from a datetime
* Current local time

### Algorithm / Pseudocode

Import pandas as PD

Initialise Timestamp as list

Append a PD timestamp of “12/1/2022 00:00:00.00” to Timestamp

Append a PD timestamp of “7/1/2022 22:00:00.00” to Timestamp

Append a PD timestamp of now to Timestamp

Append a PD timestamp of “30/12/2022 00:00:00.00” date to Timestamp

Append a PD timestamp of current date to Timestamp

Append a PD timestamp of “12/1/2022 12:34:56.00” time to Timestamp

Append a PD timestamp of current time to Timestamp

Loop through Timestamp as I

Display I

### Program Code

import pandas as pd

timestamp = []

timestamp.append(pd.Timestamp(year = 2022, month = 1, day = 12))

timestamp.append(pd.Timestamp(year = 2022, month = 1, day = 7, hour = 22))

timestamp.append(pd.Timestamp.now())

timestamp.append(pd.Timestamp(year = 2022, month = 12, day = 30).date())

timestamp.append(pd.Timestamp.now().date())

timestamp.append(pd.Timestamp(year = 2022, month = 1, day = 12, hour = 12, minute = 34, second = 56).time())

timestamp.append(pd.Timestamp.now().time())

for i in timestamp:

Text

Description automatically generated    print(i)

**Output**