**RAJALAKSHMI ENGINEERING COLLEGE**

**RAJALAKSHMI NAGAR, THANDALAM – 602 105**



|  |
| --- |
| **CS23221**  **PYTHON PROGRAMMING LAB** |
| **Laboratory Observation Note Book** |

# Name: RENUGA DEVI.K

# Year/Branch/Section: 1-year/CSE(CS)

# Register No: 2116231901041

# Semester: 2nd Semester

# Academic Year: 2023 - 2024

# 

**INDEX**

**Reg. No: 2116231901041 Name: RENUGA DEVI.K**

**Year: 1 year Branch: CSE(CS) Sec: A**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **No.** | **Date** | **Title** | **Page**  **No.** | **Teacher’s Signature /**  **Remarks** |
| **Introduction to python-Variables-Datatypes-Input/Output-Formatting** | | | | |
| 1.1 | **12.03.24** | Converting Input Strings | **8** |  |
| 1.2 | **12.03.24** | Gross salary | **10** |  |
| 1.3 | **12.03.24** | Square Root | **11** |  |
| 1.4 | **12.03.24** | Gain percent | **12** |  |
| 1.5 | **12.03.24** | Deposits | **14** |  |
| 1.6 | **12.03.24** | Carpenter | **16** |  |
| **Operators in Python** | | | | |
| 2.1 | **06.04.24** | Widgets and Gizmos | **19** |  |
| 2.2 | **06.04.24** | Doll Sings | **20** |  |
| 2.3 | **06.04.24** | Birthday party | **22** |  |
| 2.4 | **06.04.24** | Hamming Weight | **24** |  |
| 2.5 | **06.04.24** | Compound Interest | **25** |  |
| 2.6 | **06.04.24** | Eligible to donate blood | **26** |  |
| 2.7 | **06.04.24** | C or D | **28** |  |
| 2.8 | **06.04.24** | Troy Battle | **30** |  |
| 2.9 | **06.04.24** | Tax and Tip | **32** |  |
| 2.10 | **06.04.24** | Return last digit of the given number | **34** |  |
| **Selection Structures in Python** | | | | |
| 3.1 | **12.04.24** | Admission eligibility | **37** |  |
| 3.2 | **12.04.24** | Classifying triangles | **40** |  |
| 3.3 | **12.04.24** | Electricity Bill | **42** |  |
| 3.4 | **12.04.24** | IN/OUT | **45** |  |
| 3.5 | **12.04.24** | Vowel or Constant | **47** |  |
| 3.6 | **12.04.24** | Leap Year | **49** |  |
| 3.7 | **12.04.24** | Month name to Days | **52** |  |
| 3.8 | **12.04.24** | Pythagorean triple | **54** |  |
| 3.9 | **12.04.24** | Second Last Digit | **56** |  |
| 3.10 | **12.04.24** | Chinese Zodiac | **58** |  |
| **Algorithmic Approach: Iteration Control Structures** | | | | |
| 4.1 | **13.04.24** | Factors of a Number | **61** |  |
| 4.2 | **13.04.24** | Non-Repeated Digits Count | **63** |  |
| 4.3 | **13.04.24** | Prime Checking | **64** |  |
| 4.4 | **13.04.24** | Next Perfect Square | **66** |  |
| 4.5 | **13.04.24** | Nth Fibonacci | **67** |  |
| 4.6 | **13.04.24** | Disarium Number | **69** |  |
| 4.7 | **13.04.24** | Sum of Series | **70** |  |
| 4.8 | **13.04.24** | Unique Digits Count | **72** |  |
| 4.9 | **13.04.24** | Product of single digits | **73** |  |
| 4.10 | **13.04.24** | Perfect Square After adding One | **75** |  |
| **Strings in Python** | | | | |
| 5.1 | **17.04.24** | Count chars | **78** |  |
| 5.2 | **17.04.24** | Decompress the String | **79** |  |
| 5.3 | **17.04.24** | First N Common Characters | **80** |  |
| 5.4 | **17.04.24** | Remove Characters | **82** |  |
| 5.5 | **17.04.24** | Remove Palindrome Words | **84** |  |
| 5.6 | **17.04.24** | Return Second Word in Uppercase | **85** |  |
| 5.7 | **17.04.24** | Reverse String | **87** |  |
| 5.8 | **17.04.24** | String characters balance Test | **88** |  |
| 5.9 | **17.04.24** | Unique Names | **89** |  |
| 5.10 | **17.04.24** | Username Domain Extension | **90** |  |
| **List in Python** | | | | |
| 6.1 | **04.05.24** | Monotonic array | **93** |  |
| 6.2 | **04.05.24** | Check pair with difference k . | **96** |  |
| 6.3 | **04.05.24** | Count Elements | **98** |  |
| 6.4 | **04.05.24** | Distinct Elements in an Array | **101** |  |
| 6.5 | **04.05.24** | Element Insertion | **104** |  |
| 6.6 | **04.05.24** | Find the Factor | **107** |  |
| 6.7 | **04.05.24** | Merge list | **111** |  |
| 6.8 | **04.05.24** | Merge Two Sorted Arrays Without Duplication | **114** |  |
| 6.9 | **04.05.24** | Print Element Location | **117** |  |
| 6.10 | **04.05.24** | Strictly increasing | **124** |  |
| **Tuples & Set** | | | | |
| 7.1 | **18.05.24** | Binary String | **126** |  |
| 7.2 | **18.05.24** | Check Pair | **131** |  |
| 7.3 | **18.05.24** | DNA Sequence | **133** |  |
| 7.4 | **18.05.24** | Print repeated no | **136** |  |
| 7.5 | **18.05.24** | Remove repeated | **138** |  |
| 7.6 | **18.05.24** | malfunctioning keyboard | **141** |  |
| 7.7 | **18.05.24** | American keyboard | **143** |  |
| **Dictionary** | | | | |
| 8.1 | **25.05.24** | Uncommon Words | **145** |  |
| 8.2 | **25.05.24** | Sort Dictionary By Values Summation | **149** |  |
| 8.3 | **25.05.24** | Winner Of Election | **151** |  |
| 8.4 | **25.05.24** | Student Record | **154** |  |
| 8.5 | **25.05.24** | Scramble Score | **156** |  |
| **Functions** | | | | |
| 9.1 | **01.06.24** | Abundant Number | **158** |  |
| 9.2 | **01.06.24** | Automorphic number or not | **161** |  |
| 9.3 | **01.06.24** | Check Product of Digits | **166** |  |
| 9.4 | **01.06.24** | Christmas Discount | **169** |  |
| 9.5 | **01.06.24** | Coin Change | **170** |  |
| 9.6 | **01.06.24** | Difference Sum | **171** |  |
| 9.7 | **01.06.24** | Ugly number | **172** |  |
| **Searching & Sorting** | | | | |
| 10.1 | **01.06.24** | Merge Sort | **173** |  |
| 10.2 | **01.06.24** | Bubble Sort | **174** |  |
| 10.3 | **01.06.24** | Peak Element | **175** |  |
| 10.4 | **01.06.24** | Binary Search | **176** |  |
| 10.5 | **01.06.24** | Frequency of Numbers | **177** |  |
| **Exception Handling** | | | | |
| 11.1 | **02.06.24** | Exception Handling-1 | **178** |  |
| 11.2 | **02.06.24** | Exception Handling-2 | **180** |  |
| 11.3 | **02.06.24** | Exception Handling-3 | **181** |  |
| 11.4 | **02.06.24** | Exception Handling-4 | **183** |  |
| 11.5 | **02.06.24** | Exception Handling-5 | **185** |  |
| **Modules** | | | | |
| 12.1 | **07.06.24** | Representing unique pairs | **187** |  |
| 12.2 | **07.06.24** | Calculating average | **190** |  |
| 12.3 | **07.06.24** | Using dictionary | **191** |  |
| 12.4 | **07.06.24** | Power of four | **193** |  |
| 12.5 | **07.06.24** | Determining the total revenue | **195** |  |

**01 - Introduction to Python-Variables-Datatypes**

**Input/Output-Formatting**

**Ex. No: 1.1 Date: 12.03.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Converting Input Strings**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=6373)

Write a program to convert strings to an integer and float and display its type.

*Sample Input:*

10

10.9

*Sample Output:*

10,<class 'int'>

10.9,<class 'float'>

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10  10.9 | 10,<class 'int'>  10.9,<class 'float'> |

**Program:**

a=int(input())

print(a,end=",")

print(type(a))

b=float(input())

print(round(b,1),end=",")

print(type(b)



**Ex. No: 1.2 Date: 12.03.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Gross Salary**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=6374)

Ramesh’s basic salary is input through the keyboard. His dearness allowance is 40% of his basic salary, and his house rent allowance is 20% of his basic salary. Write a program to calculate his gross salary.

*Sample Input:*

10000

*Sample Output:*

16000

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10000 | 16000 |

**Program:**

a=int(input())

b=a\*0.6

print(round(a+b))

**

**Ex. No. :1.3 Date: 12.03.24**

**Register No.: 231901039 Name: Ram Haygrev S**



**Square Root**

Write a simple python program to find the square root of a given floating point number. The output should be displayed with 3 decimal places.

Sample Input:

8.00

Sample Output:

2.828

**For example:**

| **Input** | **Result** |
| --- | --- |
| 14.00 | 3.742 |

**Program:**

import math

a=float(input())

b=math.sqrt(a)

print(format(b,".3f"))



**Ex. No. : 1.4 Date: 12.03.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Gain percent**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=6376)

Alfred buys an old scooter for Rs. X and spends Rs. Y on its repairs. If he sells the scooter for Rs. Z (Z>X+Y). Write a program to help Alfred to find his gain percent. Get all the above-mentioned values through the keyboard and find the gain percent.

Input Format:

The first line contains the Rs X

The second line contains Rs Y

The third line contains Rs Z

Sample Input:

10000

250

15000

Sample Output:

46.34 is the gain percent.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 45500  500  60000 | 30.43 is the gain percent. |

**Program:**

a=int(input())

b=int(input())

c=int(input())

d=a+b

e=c-d

f=(e/d)\*100

print(format(f,".2f"),"is the gain percent.")



**Ex. No.: 1.5 Date: 12.03.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Deposits**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=6378)

In many jurisdictions, a small deposit is added to drink containers to encourage people to recycle them. In one particular jurisdiction, drink containers holding one liter or less have a $0.10 deposit and drink containers holding more than one liter have a $0.25 deposit. Write a program that reads the number of containers of each size(less and more)  from the user. Your program should continue by computing and displaying the refund that will be received for returning those containers. Format the output so that it includes a dollar sign and always displays exactly two decimal places.

Sample Input

10

20

Sample Output

Your total refund will be $6.00.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 20  20 | Your total refund will be $7.00. |

**Program:**

a=int(input())

b=int(input())

c=a\*0.1+b\*0.25

print("Your total refund will be $",format(c,".2f"),".",sep="")



**Ex. No.: 1.6 Date: 12.03.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Carpenter**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=6379)

Justin is a carpenter who works on an hourly basis. He works in a company where he is paid Rs 50 for an hour on weekdays and Rs 80 for an hour on weekends. He works 10 hrs more on weekdays than weekends. If the salary paid for him is given, write a program to find the number of hours he has worked on weekdays and weekends.

**Hint:**

If the final result(hrs) are in -ve convert that to +ve using abs() function

The abs() function returns the absolute value of the given number.

number = -20

absolute\_number = abs(number)

print(absolute\_number)

# Output: 20

**Sample Input:**

450

**Sample Output:**

weekdays 10.38

weekend 0.38

**For** exam**ple:**

| **Input** | **Result** |
| --- | --- |
| 450 | weekdays 10.38  weekend 0.38 |

**Program:**

a=int(input())

b=((500-a)/130)

b=abs(b)

c=b+10

print("weekdays",format(c,".2f"))

print("weekend",format(b,".2f"))



### [**02- Operators in Python**](https://www.rajalakshmicolleges.net/moodle/course/view.php?id=84#section-2)

**Ex. No. : 2.1 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Widgets and Gizmos**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5695)

An online retailer sells two products: widgets and gizmos. Each widget weighs 75 grams. Each gizmo weighs 112 grams. Write a program that reads the number of widgets and the number of gizmos from the user. Then your program should compute and display the total weight of the parts.

Sample Input

10

20

Sample Output

The total weight of all these widgets and gizmos is 2990 grams.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10  20 | The total weight of all these widgets and gizmos is 2990 grams. |

**Program:**

a=int(input())

b=int(input())

t=a\*75+b\*112

print("The total weight of all these widgets and gizmos is",t,"grams.")

****

**Ex. No. : 2.2 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Hamming Weight**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5698)

Write a python program that takes a integer between 0 and 15 as input and displays the number of '1' s in its binary form.(Hint:use python bitwise operator.

Sample Input

3

Sample Output:

2

Explanation:

The binary representation of 3 is 011, hence there are 2 ones in it. so the output is 2.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3 | 2 |

**Program:**

a=int(input())

binary=bin(a)[2:]

c=0

for char in binary:

if char=='1':

c=c+1

print(c)

****

**Ex. No. : 2.3 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Doll Sings**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5696)

In London, every year during Dasara there will be a very grand doll show. People try to invent new dolls of different varieties. The best-sold doll's creator will be awarded with a cash prize. So people broke their heads to create dolls innovatively. Knowing this competition, Mr.Lokpaul tried to create a doll that sings only when an even number is pressed and the number should not be zero and greater than 100.

 IF Lokpaul wins print true, otherwise false.

Sample Input

10

Sample Output

True

Explanation:

Since 10 is an even number and a number between 0 and 100, True is printed

**For example:**

| **Input** | **Result** |
| --- | --- |
| 101 | False |

**Program:**

a=int(input())

if(a>0 and a<101):

if(a%2==0):

print("True")

else:

print("False")



**Ex. No. : 2.4 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Return last digit of the given number**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=6031)

Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number.

The last digit should be returned as a positive number.

For example,

if the given number is 197, the last digit is 7

if the given number is -197, the last digit is 7

**For example:**

| **Input** | **Result** |
| --- | --- |
| 197 | 7 |
| -197 | 7 |

**Program:**

a=int(input())

b=abs(a)

c=b%10

print(c)

****

**Ex. No. : 2.5 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Tax and Tip**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=6030)

The program that you create for this exercise will begin by reading the cost of a meal ordered at a restaurant from the user. Then your program will compute the tax and tip for the meal. Use your local tax rate (5 percent) when computing the amount of tax owing. Compute the tip as 18 percent of the meal amount (without the tax). The output from your program should include the tax amount, the tip amount, and the grand total for the meal including both the tax and the tip. Format the output so that all of the values are displayed using two decimal places.

Sample Input

100

Sample Output

The tax is 5.00 and the tip is 18.00, making the total 123.00

**For example:**

| **Input** | **Result** |
| --- | --- |
| 100 | The tax is 5.00 and the tip is 18.00, making the total 123.00 |

**Program:**

a=int(input())

tax=a\*0.05

tip=a\*0.18

total=a+tax+tip

print("The tax is ",format(tax,".2f")," and the tip is ",format(tip,".2f"),","," making the total ",format(total,".2f"),sep="")

****

**Ex. No. : 2.6 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Eligible to donate blood**

A team from the Rotract club had planned to conduct a rally to create awareness among the Coimbatore people to donate blood. They conducted the rally successfully. Many of the Coimbatore people realized it and came forward to donate their blood to nearby blood banks. The eligibility criteria for donating blood are people should be above or equal to 18 and his/ her weight should be above 40. There was a huge crowd and staff in the blood bank found it difficult to manage the crowd. So they decided to keep a system and ask the people to enter their age and weight in the system. If a person is eligible he/she will be allowed inside.

 Write a program and feed it to the system to find whether a person is eligible or not.

Input Format:

 Input consists of two integers that correspond to the age and weight of a person respectively.

 Output Format:

 Display True(IF ELIGIBLE)

Display False (if not eligible)

Sample Input

19

45

Sample Output

True

**For example:**

| **Input** | **Result** |
| --- | --- |
| 18  40 | False |

**Program:**

a=int(input())

b=int(input())

if(a>=18 and b>40):

print("True")

else:

print("False")

****

**Ex. No. : 2.7 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Birthday Party**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5697)

Mr. X's birthday is in next month. This time he is planning to invite N of his friends. He wants to distribute some chocolates to all of his friends after the party. He went to a shop to buy a packet of chocolates. At the chocolate shop, 4 packets are there with different numbers of chocolates. He wants to buy such a packet which contains a number of chocolates, which can be distributed equally among all of his friends. Help Mr. X to buy such a packet.

 Input Given:

N-No of friends

P1,P2,P3 AND P4-No of chocolates

OUTPUT:

 "True" if he can buy that packet and "False" if he can't buy that packet.

SAMPLE INPUT AND OUTPUT:

5

25

12

10

9

OUTPUT

True False True False

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  25  23  20  10 | True False True True |

**Program:**

a=int(input())

b=[]

for i in range(4):

c=int(input())

b.append(c)

for i in range(4):

if(b[i]%a==0):

print("True",end=" ")

else:

print("False",end=" ")



**Ex. No. : 2.8 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Troy Battle**

In the 1800s, the battle of Troy was led by Hercules. He was a superstitious person. He believed that his crew can win the battle only if the total count of the weapons in hand is in multiple of 3 and the soldiers are in an even number of count. Given the total number of weapons and the soldier's count, Find whether the battle can be won or not according to Hercules's belief. If the battle can be won print True otherwise print False.

**Input format:**

Line 1 has the total number of weapons

Line 2 has the total number of Soldiers.

**Output  Format:**

If the battle can be won print True otherwise print False.  
Sample Input:

32

43

Sample Output:'

False

**For example:**

| **Input** | **Result** |
| --- | --- |
| 32  43 | False |

**Program:**

a=int(input())

b=int(input())

if(a%3==0 and b%2==0):

print("True")

else:

print("False")



**Ex. No. : 2.9 Date: 06.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Compound Interest**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5699)

Pretend that you have just opened a new savings account that earns 4 percent interest per year. The interest that you earn is paid at the end of the year, and is added to the balance of the savings account. Write a program that begins by reading the amount of money deposited into the account from the user. Then your program should compute and display the amount in the savings account after 1, 2, and 3 years. Display each amount so that it is rounded to 2 decimal places.

Sample Input:

10000

Sample Output:

Balance as of end of Year 1: $10400.00.

Balance as of end of Year 2: $10816.00.

Balance as of end of Year 3: $11248.64

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10000 | Balance as of end of Year 1: $10400.00.  Balance as of end of Year 2: $10816.00.  Balance as of end of Year 3: $11248.64. |

**Program:**

a=int(input())

b=a+a\*0.04

c=b+b\*0.04

d=c+c\*0.04

print("Balance as of end of Year 1: ","$",format(b,".2f"),".",sep="")

print("Balance as of end of Year 2: ","$",format(c,".2f"),".",sep="")

print("Balance as of end of Year 3: ","$",format(d,".2f"),".",sep="")



**Ex. No. : 2.10 Date: 06.04.24**

**Register No.:231901041 Name: RENUGA DEVI**



**C or D**

Mr.Ram has been given a problem kindly help him to solve it. The input of the program is either 0 or 1. IF 0 is the input he should display "C" if 1 is the input it should display "D".There is a constraint that Mr. Ram should use either logical operators or arithmetic operators to solve the problem, not anything else.

Hint:

Use ASCII values of C and D.

**Input Format:**

An integer x, 0<=x<=1. .

**Output Format:**

output a single character "C" or "D"depending on the value of x.

**Input 1:**

0

**Output 1:**

C

**Input 2:**

1

**Output 1:**

D

**For example:**

| **Input** | **Result** |
| --- | --- |
| 0 | C |

**Program:**

a=int(input())

if(a==0):

print("C")

elif(a==1):

print("D")



### [**03 - Selection Structures in Python**](https://www.rajalakshmicolleges.net/moodle/course/view.php?id=84#section-3)

**Ex. No. : 3.1 Date: 12.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Classifying Triangles**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5707)

A triangle can be classified based on the lengths of its sides as equilateral, isosceles or scalene. All three sides of an equilateral triangle have the same length. An isosceles triangle has two sides that are the same length, and a third side that is a different length. If all of the sides have different lengths then the triangle is scalene.

Write a program that reads the lengths of the three sides of a triangle from the user. Then display a message that states the triangle’s type.

Sample Input 1

60

60

60

Sample Output 1

That's a equilateral triangle

Sample Input 2

40

40

80

Sample Output 2

That's a isosceles triangle

Sample Input 3

50

60

70

Sample Output 3

That's a scalene triangle

**For example:**

| **Input** | **Result** |
| --- | --- |
| 60  60  60 | That's a equilateral triangle |
| 40  40  80 | That's a isosceles triangle |

**Program:**

a=int(input())

b=int(input())

c=int(input())

if(a==b and b==c and c==a):

print("That's","a","equilateral triangle")

elif(a!=b and a!=c and b!=c):

print("That's","a","scalene triangle")

else:

print("That's","a","isosceles triangle")

****

**Ex. No. : 3.2 Date: 12.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Vowel or Consonant**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5710)

In this exercise you will create a program that reads a letter of the alphabet from the user. If the user enters a, e, i, o or u then your program should display a message indicating that the entered letter is a vowel. If the user enters 'y' then your program should display a message indicating that sometimes y is a vowel, and sometimes y is a consonant. Otherwise your program should display a message indicating that the letter is a consonant.

Sample Input 1

i

Sample Output 1

It's a vowel.

Sample Input 2

y

Sample Output 2

Sometimes it's a vowel... Sometimes it's a consonant.

Sample Input3

c

Sample Output 3

It's a consonant.

**For example:**

| **Input** | **Result** |
| --- | --- |
| y | Sometimes it's a vowel... Sometimes it's a consonant. |
| u | It's a vowel. |
| p | It's a consonant. |

**Program:**

x=input()

if(x=='y'):

print("Sometimes it's a vowel... Sometimes it's a consonant.")

elif(x=='a' or x=='e' or x=='i' or x=='o' or x=='u'):

print("It's a vowel.")

else:

print("It's a consonant.")

****

**Ex. No. : 3.3 Date: 12.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Electricity Bill**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5708)

Write a program to calculate and print the Electricity bill where the unit consumed by the user is given from test case. It prints the total amount the customer has to pay. The charge are as follows:

Unit                                                     Charge / Unit

Upto 199                                             @1.20

200 and above but less than 400        @1.50

400 and above but less than 600        @1.80

600 and above                                    @2.00

If bill exceeds Rs.400 then a surcharge of 15% will be charged and the minimum bill should be of Rs.100/-

Sample Test Cases

Test Case 1

Input

50

Output

100.00

Test Case 2

Input

300

Output

517.50

**For example:**

| **Input** | **Result** |
| --- | --- |
| 100.00 | 120.00 |

**Program:**

a=float(input())

if(a<200):

b=a\*1.20

if(b<=100):

b=100

print(format(b,".2f"))

else:

print(format(b,".2f"))

elif(a>=200 and a<400):

c=a\*1.50

if(c>=400):

print(format(c\*0.15+c,".2f"))

else:

print(format(c,".2f"))

elif(a>=400 and a<600):

d=a\*1.80

print(format(d\*0.15+d,".2f"))

elif(a>600):

e=a\*2.00

print(format(e\*0.15+e,".2f"))



**Ex. No. : 3.4 Date: 12.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**IN/OUT**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5709)

Ms. Sita, the faculty handling programming lab for you is very strict. Your seniors have told you that she will not allow you to enter the week's lab if you have not completed atleast half the number of problems given last week. Many of you didn't understand this statement and so they requested the good programmers from your batch to write a program to find whether a student will be allowed into a week's lab given the number of problems given last week and the number of problems solved by the student in that week.

Input Format:

Input consists of 2 integers.

The first integer corresponds to the number of problems given and the second integer corresponds to the number of problems solved.

 Output Format:

Output consists of the string “IN” or “OUT”.

 Sample Input and Output:

Input

 8

 3

 Output

OUT

**For example:**

| **Input** | **Result** |
| --- | --- |
| 8  3 | OUT |

**Program:**

a=int(input())

b=int(input())

c=a//2

if(b>=c):

print("IN")

else:

print("OUT")

****

**Ex. No. : 3.5 Date: 12.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Second last digit**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5714)

Write a program that returns the second last digit of the given number. Second last digit is being referred 10the digit in the tens place in the given number.

For example, if the given number is 197, the second last digit is 9.

Note1 - The second last digit should be returned as a positive number. i.e. if the given number is -197, the second last digit is 9.

Note2 - If the given number is a single digit number, then the second last digit does not exist. In such cases, the program should return -1. i.e. if the given number is 5, the second last digit should be returned as -1.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 197 | 9 |

**Program:**

a=int(input())

b=abs(a)

if(b>=10):

c=b//10

d=c%10

print(d)

else:

print(-1)



**Ex. No. : 3.6 Date: 12.04.24**

**Register No.:231901041 Name: RENUGA DEVI**



**Chinese Zodiac**

The Chinese zodiac assigns animals to years in a 12 year cycle. One 12 year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the dragon, and 1999 being another year of the hare.

Year Animal

2000 Dragon

2001 Snake

2002 Horse

2003 Sheep

2004 Monkey

2005 Rooster

2006 Dog

2007 Pig

2008 Rat

2009 Ox

2010 Tiger

2011 Hare

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

Sample Input 1

2010

Sample Output 1

2010 is the year of the Tiger.

Sample Input 2

2020

Sample Output 2

2020 is the year of the Rat.

**Program:**

a=int(input())

b=a%12

if(b==0):

print(a,"is the year of the Monkey.")

elif(b==1):

print(a,"is the year of the Rooster.")

elif(b==2):

print(a,"is the year of the Dog.")

elif(b==3):

print(a,"is the year of the Pig.")

elif(b==4):

print(a,"is the year of the Rat.")

elif(b==5):

print(a,"is the year of the Ox.")

elif(b==6):

print(a,"is the year of the Tiger.")

elif(b==7):

print(a,"is the year of the Hare.")

elif(b==8):

print(a,"is the year of the Dragon.")

elif(b==9):

print(a,"is the yaer of the Snake.")

elif(b==10):

print(a,"is the year of the Horse.")

elif(b==11):

print(a,"is the year of the Sheep.")



**Ex. No. : 3.7 Date: 12.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Pythagorean triple**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5713)

Three numbers form a Pythagorean triple if the [sum of squares](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5759) of two numbers is equal to the square of the third.

For example, 3, 5 and 4 form a Pythagorean triple, since 3\*3 + 4\*4 = 25 = 5\*5   
You are given three integers, a, b, and c. They need not be given in increasing order. If they form a Pythagorean triple, then print "Yes", otherwise, print "No".

Sample Input

3

5

4

Sample Output

yes

Sample Test Cases

Test Case 1

Input

3

5

4

Output

yes

Test Case 2

Input

5

8

2

Output

no

**Program:**

a=int(input())

b=int(input())

c=int(input())

if(a\*a+b\*b==c\*c or b\*b+c\*c==a\*a or c\*c+a\*a==b\*b):

print("yes")

else:

print("no")



**Ex. No. : 3.8 Date: 12.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Leap Year**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5711)

Most years have 365 days. However, the time required for the Earth to orbit the Sun is actually slightly more than that. As a result, an [extra](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5753) day, February 29, is included in some years to correct for this difference. Such years are referred to as leap years. The rules for determining whether or not a year is a leap year follow:

• Any year that is divisible by 400 is a leap year.

• Of the remaining years, any year that is divisible by 100 is not a leap year.

• Of the remaining years, any year that is divisible by 4 is a leap year.

• All other years are not leap years.

Write a program that reads a year from the user and displays a message indicating whether or not it is a leap year.

Sample Input 1

1900

Sample Output 1

1900 is not a leap year.

Sample Input 2

2000

Sample Output 2

2000 is a leap year.

**Program:**

a=int(input())

b=a%100

c=a%400

if(b==0):

if(c==0):

print(a,"is a leap year.")

else:

print(a,"is not a leap year.")

elif(a%4==0):

print(a,"is a leap year.")

****

**Ex. No. : 3.9 Date: 12.04.24**

**Register No.: 231901041 Name : RENUGA DEVI**



[**Month name to days**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5712)

The length of a month varies from 28 to 31 days. In this exercise you will create a program that reads the name of a month from the user as a string. Then your program should display the number of days in that month. Display “28 or 29 days” for February so that leap years are addressed.

Sample Input 1

February

Sample Output 1

February has 28 or 29 days in it.

Sample Input 2

March

Sample Output 2

March has 31 days in it.

Sample Input 3

April

Sample Output 3

April has 30 days in it.

**For example:**

| **Input** | **Result** |
| --- | --- |
| February | February has 28 or 29 days in it. |

**Program:**

odd=['January','March','May','July','August','October','December']

even=['April','June','September','November']

a=input()

if(a=='February'):

print("February has 28 or 29 days in it.")

if a in odd:

print(a,"has 31 days in it.")

if a in even:

print(a,"has 30 days in it.")

****

**Ex. No. : 3.10 Date: 12.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Admission Eligibility**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5706)

Write a program to find the eligibility of admission for a professional course based on the following criteria:

Marks in Maths >= 65

Marks in Physics >= 55

Marks in Chemistry >= 50

Or

Total in all three subjects >= 180

Sample Test Cases

Test Case 1

Input

70

60

80

Output

The candidate is eligible

Test Case 2

Input

50

80

80

Output

The candidate is eligible

Test Case 3

Input

50

60

40

Output

The candidate is not eligible

**For example:**

| **Input** | **Result** |
| --- | --- |
| 50  80  80 | The candidate is eligible |

**Program:**

a=int(input())

b=int(input())

c=int(input())

if(a>=65 and b>=55 and c>=50):

print("The candidate is eligible")

elif(a+b+c>=180):

print("The candidate is eligible")

else:

print("The candidate is not eligible")



[**04 - Iteration Control Structures**](https://www.rajalakshmicolleges.net/moodle/course/view.php?id=84#section-4)

**Ex. No. : 4.1 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Nth Fibonacci**

Write a [program](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=3478) to return the nth number in the fibonacci series. The value of N will be passed to the [program](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=3478) as input.

NOTE: Fibonacci series looks like –

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, . . . and so on.

i.e. Fibonacci series starts with 0 and 1, and continues generating the next number as the sum of the previous two numbers.

• first Fibonacci number is 0,

• second Fibonacci number is 1,

• third Fibonacci number is 1,

• fourth Fibonacci number is 2,

• fifth Fibonacci number is 3,

• sixth Fibonacci number is 5,

• seventh Fibonacci number is 8, and so on.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1  4  7 | 0  2  8 |

**Program:**

a=int(input())

b=0

c=1

if(a==1):

print("0")

elif(a==2):

print("1")

else:

for i in range (3,a+1):

d=b+c

b=c

c=d

print(d)



**Ex. No. : 4.2 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Factors of a number**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5720)

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number).

**For example:**

| **Input** | **Result** |  |
| --- | --- | --- |
| 20 | 1 2 4 5 10 20 |  |

**Program:**

a=int(input())

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

if(a%i==0):

print(i,end=" ")



**Ex. No. : 4.3 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Product of single digit**

Given a positive integer N, check whether it can be represented as a product of single digit numbers.

Input Format:

Single Integer input.

Output Format:

Output displays Yes if condition satisfies else prints No.

Example Input:

14

Output:

Yes

Example Input:

13

Output:

No

**Program:**

a=int(input())

c=0

for i in range(1,10): for j in range(1,10):

if i\*j==a:

c=1

if(c==1):

print("Yes")

▾else:

print("No")

****

**Ex. No. : 4.4 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Unique Digit Count**

Write a program to find the count of unique digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number >= 1 and <= 25000.

For e.g.

If the given number is 292, the program should return 2 because there are only 2 unique digits '2' and '9' in this number

If the given number is 1015, the program should return 3 because there are 3 unique digits in this number, '1', '0', and '5'.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 292 | 2 |
| 1015 | 3 |

**Program:**

a=input()

b=len(set(a))

print(b)

****

**Ex. No. : 4.5 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Non**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5717) **Repeated Digit Count**

Write a program to find the count of non-repeated digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number >= 1 and <= 25000.

Some examples are as below.

If the given number is 292, the program should return 1 because there is only 1 non-repeated digit '9' in this number

If the given number is 1015, the program should return 2 because there are 2 non-repeated digits in this number, '0', and '5'.

If the given number is 108, the program should return 3 because there are 3 non-repeated digits in this number, '1', '0', and '8'.

If the given number is 22, the function should return 0 because there are NO non-repeated digits in this number.

| **Input** | **Result** |
| --- | --- |
| 292 | 1 |
| 1015 | 2 |
| 108 | 3 |
| 22 | 0 |

**For example:**

**Program:**

a={}

for i in input:

if i in a:a[i]+=1

else:a[i]=1

print(sum([1 for i in a if a[i]==1]))

****

**Ex. No. : 4.6 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Next Perfect Square**

Given a number N, find the next perfect square greater than N.

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

10

Output:

16

**Program:**

import math

a=int(input())

b = a + 1

while b > 0 :

m=math.sqrt(b)

if(m==int(m)):

print(b)

break

else:

b = b + 1

****

**Ex. No. : 4.7 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Sum of Series**

Write a program to find the sum of the series 1 +11 + 111 + 1111 + . . . + n terms (n will be given as input from the user and sum will be the output)

Sample Test Cases

Test Case 1

Input

4

Output

1234

Explanation:

as input is 4, have to take 4 terms.

1 + 11 + 111 + 1111

Test Case 2

Input

6

Output

123456

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3 | 123 |

**Program:**

a=int(input())

t=1

s=0

for i in range(a)

s+=t

t=t\*10+1

print(s)

****

**Ex. No. : 4.8 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Prime Checking**

Write a program that finds whether the given number N is Prime or not. If the number is prime, the program should return 2 else it must return 1.

Assumption: 2 <= N <=5000, where N is the given number.

Example1: if the given number N is 7, the method must return 2

Example2: if the given number N is 10, the method must return 1

**For example:**

| **Input** | **Result** |
| --- | --- |
| 7 | 2 |
| 10 | 1 |

**Program:**

a=int(input())

c=0

for i in range(2,a):

if(a%i==0):

c=1

if(c==1):

print("1")

elif(c==0): print("2”)

****

**Ex. No. : 4.9 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Disarium Number**

A Number is said to be Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself. Write a [program](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=3478) to print number is Disarium or not.

Input Format:

Single Integer Input from stdin.

Output Format:

Yes or No.

Example Input:

175

Output:

Yes

Explanation

1^1 + 7^2 +5^3 = 175

Example Input:

123

Output:

No

**For example:**

| **Input** | **Result** |
| --- | --- |
| 175 | Yes |
| 123 | No |

**Program:**

a=input()

n=len(a)

r=0

for i,d in enumerate(a):

r+=int(d)\*\*(i+1)

if r==int(a):

print("Yes")

else:

print("No")

****

**Ex. No. : 4.10 Date: 13.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Perfect Square After adding One**

Given an integer N, check whether N the given number can be made a perfect square after adding 1 to it.

Input Format:

Single integer input.

Output Format:

Yes or No.

Example Input:

24

Output:

Yes

Example Input:

26

Output:

No

**For example:**

| **Input** | **Result** |
| --- | --- |
| 24 | Yes |

**Program:**

import math

a=int(input())

b=a+1

c=math.sqrt(b)

if(c==int(c)):

print("Yes")

else:

print("No")

****

### [**05 -**](https://www.rajalakshmicolleges.net/moodle/course/view.php?id=84#section-5) **Strings in Python**

**Ex. No. : 5.1 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**String characters balance Test**

Write a program to check if two strings are balanced. For example, strings s1 and s2 are balanced if all the characters in the s1 are present in s2. The character’s position doesn’t matter. If balanced display as "true" ,otherwise "false".

**For example:**

| **Input** | **Result** |
| --- | --- |
| Yn  PYnative | True |

**Program:**

a=input()

b=input()

if a in b or b in a:

print("True")

else:

print("False")

****

**Ex. No. : 5.2 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Decompress the String**

Assume that the given string has enough memory. Don't use any extra space(IN-PLACE)

Sample Input 1

a2b4c6

Sample Output 1

aabbbbcccccc

**Program:**

s=input()

r=""

i=0

while i< len(s):

char=s[i]

i+=1

num=""

while i<len(s) and s[i].isdigit():

num+=s[i]

i+=1

r+=char\*int(num)

print(r)

****

**Ex. No. : 5.3 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**First N Common Chars**

Two string values S1, S2 are passed as the input. The program must print first N characters present in S1 which are also present in S2.

Input Format:

The first line contains S1.

The second line contains S2.

The third line contains N.

Output Format:

The first line contains the N characters present in S1 which are also present in S2.

Boundary Conditions:

2 <= N <= 10

2 <= Length of S1, S2 <= 1000

Example Input/Output 1:

Input:

abcbde

cdefghbb

3

Output:

bcd

Note:

b occurs twice in common but must be printed only once.

**Program:**

a=input()

b=input()

n=int(input())

bset=set(b)

cc=[]

c=0

for i in a:

if i in bset and i not in cc:

cc.append(i)

c=c+1

if(c==n):

break

s=''.join(cc)

print(s)

****

**Ex. No. : 5.4 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Username Domain Extension**

Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

**Input Format**:

The first line contains S.

**Output Format**:

The first line contains EXTENSION.

The second line contains DOMAIN.

The third line contains USERNAME.

**Boundary Condition:**

1 <= Length of S <= 100

Example Input/Output 1:

**Input**:

vijayakumar.r@rajalakshmi.edu.in

**Output**:

edu.in

rajalakshmi

vijayakumar.r

**Program:**

s=input()

at=s.index('@')

dot=s.index('.')

username=s[:at]

domain=s[at+1:dot]

exten=s[dot+1:]

print(exten)

print(domain)

print(username)

****

**Ex. No. : 5.5 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Count Chars**

Write a python program to count all letters, digits, and special symbols respectively from a given string

**For example:**

| **Input** | **Result** |
| --- | --- |
| rec@123 | 3  3  1 |

**Program:**

x=input()

a,b,c=0,0,0

for i in x:

if(i.isalpha()):

a+=1

elif(i.isalnum()):

b+=1

else:

c+=1

print(a,b,c,sep="\n")



**Ex. No. : 5.6 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Reverse String**

Reverse a string without affecting special characters. Given a string S, containing special characters and all the alphabets, reverse the string without affecting the positions of the special characters.

Input:

A&B

Output:

B&A

Explanation: As we ignore '&' and

As we ignore '&' and then reverse, so answer is "B&A".

For example:

Input Result

A&x#

x&A#

**Program:**

s=input()

l=[]

for i in s:

if(i.isalpha()):

l.append(i)

l.reverse()

r=''

index=0

for i in s:

if(i.isalpha()):

r+=l[index]

index+=1

else:

r+=i

print(r)

****

**Ex. No. : 5.7 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Longest Word**

Write a python to read a sentence and print its longest word and its length

**For example:**

| **Input** | **Result** |
| --- | --- |
| This is a sample text to test | sample  6 |

**Program:**

sen=input()

words=sen.split()

l=""

maxi=0

for word in words:

if(len(word)>maxi):

l=word

maxi=len(word)

print(l,maxi,sep="\n")



**Ex. No. : 5.8 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Remove Palindrome Words**

String should contain only the words are not palindrome.

Sample Input 1

Malayalam is my mother tongue

Sample Output 1

is my mother tongue

**Program:**

s=input()

words=s.split()

x=''

for word in words:

word=word.lower()

if (word!=word[::-1]):

print(word,end=" ")

****

**Ex. No. : 5.9 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Remove Characters**

Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Constraints

1<= string length <= 200

Sample Input 1

experience

enc

Sample Output 1

xpri

**Program:**

s1=input()

s2=input()

x=''.join(char for char in s1 if char not in s2)

print(x)

****

**Ex. No. : 5.10 Date: 17.04.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Unique Names**

In this exercise, you will create a program that reads words from the user until the user enters a blank line. After the user enters a blank line your program should display each word entered by the user exactly once. The words should be displayed in the same order that they were first entered. For example, if the user enters:

**Input:**

first

second

first

third

second

then your program should display:

**Output:**

first

second

third

**Program:**

l=[]

while(True):

a=input()

if a!=" ":

l.append(a)

else:

break

l=dict.fromkeys(l)

for i in l:

print(i)

****

### **[06 - List in Python](https://www.rajalakshmicolleges.net/moodle/course/view.php?id=84" \l "section-5)**

**Ex. No. : 6.1 Date: 04.05.24**

**Register No.:231901041 Name: RENUGA DEVI**



**Element Insertion**

Consider a program to insert an element / item in the sorted array. Complete the logic by filling up required code in editable section. Consider an array of size 10. The eleventh item is the data is to be inserted.

Sample Test Cases

Test Case 1

Input

1

3

4

5

6

7

8

9

10

11

2

Output

ITEM to be inserted:2

After insertion array is:

1

2

3

4

5

6

7

8

9

10

11

Test Case 2

Input

11

22

33

55

66

77

88

99

110

120

44

Output

ITEM to be inserted:44

After insertion array is:

11

22

33

44

55

66

77

88

99

110

120

**Program:**

x=[]

for i in range(0,11):

b=int(input())

x.append(b)

#a.sort()

print("ITEM to be inserted:",x[-1],sep='')

x.sort()

print("After insertion array is:")

for i in x:

print(i)



**Ex. No. : 6.2 Date: 04.05.24**

**Register No.:231901041 Name: RENUGA DEVI**



**Anagram**

Given two lists A and B, and B is an anagram of A. B is an anagram of A means B is made by randomizing the order of the elements in A.

We want to find an *index mapping* P, from A to B. A mapping P[i] = j means the ith element in A appears in B at index j.

These lists A and B may contain duplicates. If there are multiple answers, output any of them.

For example, given

**Input**

5

12 28 46 32 50

50 12 32 46 28

**Output**

1 4 3 2 0

**Explanation**

A = [12, 28, 46, 32, 50]

B = [50, 12, 32, 46, 28]

We should return

[1, 4, 3, 2, 0]

as P[0] = 1 because the 0th element of A appears at B[1], and P[1] = 4 because the 1st element of A appears at B[4], and so on.

**Note:**

1. A, B have equal lengths in range [1, 100].
2. A[i], B[i] are integers in range [0, 10^5].

**Program:**

def index\_mapping(A, B):

index\_map = {num: i for i, num in enumerate(B)}

return ' '.join(str(index\_map[num]) for num in A)

n=int(input())

A = list(map(int, input().split()))

B = list(map(int, input().split()))

print(index\_mapping(A, B))

****

**Ex. No. : 6.3 Date: 04.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Merge Two Sorted Arrays Without Duplication**

Output is a merged array without duplicates.

Input Format

N1 - no of elements in array 1

Array elements for array 1

N2 - no of elements in array 2

Array elements for array2

Output Format

Display the merged array

**Sample Input 1**

5

1

2

3

6

9

4

2

4

5

10

**Sample Output 1**

1 2 3 4 5 6 9 10

**Program:**

n1=int(input())

l1=[]

for i in range(0,n1):

a=int(input())

l1.append(a)

n2=int(input())

l2=[]

for i in range(0,n2):

a=int(input())

l2.append(a)

l3=[]

l3.extend(l1)

l3.extend(l2)

a=list(set(l3))

a.sort()

for i in a:

print(i,end=' ') n1=int(input())

l1=[]

for i in range(0,n1):

a=int(input())

l1.append(a)

n2=int(input())

l2=[]

for i in range(0,n2):

a=int(input())

l2.append(a)

l3=[]

l3.extend(l1)

l3.extend(l2)

a=list(set(l3))

a.sort()

for i in a:

print(i,end=' ')

****

**Ex. No. : 6.4 Date: 04.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Distinct Elements in an Array**

Program to print all the distinct elements in an array. Distinct elements are nothing but the unique (non-duplicate) elements present in the given array.

Input Format:

First line take an Integer input from stdin which is array length n.

Second line take n Integers which is inputs of array.

Output Format:

Print the Distinct Elements in Array in single line which is space Separated

Example Input:

5

1

2

2

3

4

Output:

1 2 3 4

Example Input:

6

1

1

2

2

3

3

Output:

1 2 3

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  1  2  2  3  4 | 1 2 3 4 |
| 6  1  1  2  2  3  3 | 1 2 3 |

**Program:**

n = int(input())

arr = []

for \_ in range(n):

arr.append(int(input()))

distinct\_elements = set(arr)

print(\*distinct\_elements)

****

**Ex. No. : 6.5 Date: 04.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**The Pivot**

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

Example

arr=[1,2,3,4,6]

·         the sum of the first three elements, 1+2+3=6. The value of the last element is 6.

·         Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.

·         The index of the pivot is 3.

Constraints

·         3 ≤ n ≤ 105

·         1 ≤ arr[i] ≤ 2 × 104, where 0 ≤ i < n

·         It is guaranteed that a solution always exists.

The first line contains an integer n, the size of the array arr.

Each of the next n lines contains an integer, arr[i], where 0 ≤ i < n.

Sample Case 0

Sample Input 0

4

1

2

3

3

Sample Output 0

2

Explanation 0

·         The sum of the first two elements, 1+2=3. The value of the last element is 3.

·         Using zero based indexing, arr[2]=3 is the pivot between the two subarrays.

·         The index of the pivot is 2.

Sample Case 1

Sample Input 1

3

1

2

1

Sample Output 1

1

Explanation 1

·         The first and last elements are equal to 1.

·         Using zero based indexing, arr[1]=2 is the pivot between the two subarrays.

·         The index of the pivot is 1.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  1  2  3  3 | 2 |
| 3  1  2  1 | 1 |

**Program:**

a = int(input())

b= []

for i in range(a):

element = int(input())

b.append(element)

total= sum(b)

left= 0

right = total- b[0]

if left== right:

print(0)

exit()

for i in range(1, a):

left+= b[i - 1]

right-= b[i]

if left== right:

print(i)

break

****

**Ex. No. : 6.6 Date: 04.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Intersection of array**

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

The first line contains T, the number of test cases. Following T lines contain:

1.      Line 1 contains N1, followed by N1 integers of the first array

2.      Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

7

1

2

3

3

4

5

6

2

1

6

Output:

1 6

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1  3  10  17  57  6  2  7  10  15  57  246 | 10 57 |
| 1  7  1  2  3  3  4  5  6  2  1  6 | 1 6 |

**Program:**

t=int(input())

l1=list()

while(t!=0):

n1=int(input())

l1=[]

l2=[]

for i in range(0,n1):

a=int(input())

l1.append(a)

n2=int(input())

for i in range(0,n2):

a=int(input())

l2.append(a)

t=t-1

c=set(l1)

d=set(l2)

e=list(c.intersection(d))

e.sort()

for i in e:

print(i,end=' ')

print('\n')

****

**Ex. No. : 6.7 Date: 04.05.24**

**Register No.:231901041 Name: RENUGA DEVI**



**Location**

Write a program to print all the locations at which a particular element (taken as input) is found in a list and also print the total number of times it occurs in the list. The location starts from 1.

For example, if there are 4 elements in the array:

5

6

5

7

If the element to search is 5 then the output will be:

5 is present at location 1

5 is present at location 3

5 is present 2 times in the array.

Sample Test Cases

Test Case 1

Input

4

5

6

5

7

5

Output

5 is present at location 1.

5 is present at location 3.

5 is present 2 times in the array.

Test Case 2

Input

5

67

80

45

97

100

50

Output

50 is not present in the array.

**Program:**

n = int(input())

arr = [int(input()) for \_ in range(n)]

element\_to\_search = int(input())

locations = []

occurrences = 0

for i in range(len(arr)):

if arr[i] == element\_to\_search:

locations.append(i + 1)

occurrences +=1

if occurrences == 0:

print(f"{element\_to\_search} is not present in the array.")

else:

for loc in locations:

print(f"{element\_to\_search} is present at location {loc}.")

print(f"{element\_to\_search} is present {occurrences} times in the array.")

****

**Ex. No. : 6.8 Date: 04.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Strictly increasing**

Write a Python program to check if a given list is strictly increasing or not. Moreover, If removing only one element from the list results in a strictly increasing list, we still consider the list true

Input:

n : Number of elements

List1: List of values

Output

Print "True" if list is strictly increasing or decreasing else print "False"

Sample Test Case

Input

7

1

2

3

0

4

5

6

Output

True

**Program:**

def check\_increasing\_or\_decreasing(lst):

increasing = True

decreasing = True

for i in range(1, len(lst)):

if lst[i] > lst[i - 1]:

decreasing = False

elif lst[i] < lst[i - 1]:

increasing = False

return increasing or decreasing

def check\_strictly\_increasing\_with\_removal(lst):

for i in range(len(lst)):

temp\_lst = lst[:i] + lst[i+1:]

if check\_increasing\_or\_decreasing(temp\_lst):

return True

return False

n = int(input())

lst = []

for \_ in range(n):

lst.append(int(input())

if check\_increasing\_or\_decreasing(lst) or check\_strictly\_increasing\_with\_removal(lst):

print("True")

else:

print("False")

****

**Ex. No. : 6.9 Date: 04.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Merge List**

Write a Python program to Zip two given lists of lists.

Input:

m : row size

n: column size

list1 and list 2 : Two lists

Output

Zipped List : List which combined both list1 and list2

Sample test case

Sample input

2

2

1

3

5

7

2

4

6

8

Sample Output

[[1, 3, 2, 4], [5, 7, 6, 8]]

**Program:**

m=int(input())

n=int(input())

l1=[]

l2=[]

c=1

for i in range(0,m\*n\*2,2):

a=int(input())

b=int(input())

if c%2!=0:

l1.append(a)

l1.append(b)

else:

l2.append(a)

l2.append(b)

c=c+1

l3=[]

l3.append(l1)

l3.append(l2)

print(l3)

****

**Ex. No. : 6.10 Date: 04.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Check pair with difference k**

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[i] - A[j] = k, i != j.

Input Format

1.      First line is number of test cases T. Following T lines contain:

2.      N, followed by N integers of the array

3.      The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn’t.

Input

1

3

1

3

5

4

Output:

1

Input

1

3

1

3

5

99

Output

0

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1  3  1  3  5  4 | 1 |
| 1  3  1  3  5  99 | 0 |

**Program:**

t=int(input())

for i in range(0,t):

n=int(input())

l=[]

for j in range(0,n):

a=int(input())

l.append(a)

p=int(input())

for k in range(0,n):

c=0

for m in range(i+1,n):

if l[m]-l[k]==p:

c=1

print('1')

break

if c==1:

break

if c==0:

print('0')



**07 – Tuple/Set**

**Ex. No. : 7.1 Date: 18.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Binary String**

Coders here is a simple task for you, Given string str. Your task is to check whether it is a binary string or not by using python set.

Examples:

Input: str = "01010101010"

Output: Yes

Input: str = "REC101"

Output: No

**For example:**

| **Input** | **Result** |
| --- | --- |
| 01010101010 | Yes |
| 010101 10101 | No |

**Program:**

a = input()

try:

c = int(a)

print("Yes")

except:

print("No")

****

**Ex. No. : 7.2 Date: 18.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**DNA Sequence**

The **DNA sequence** is composed of a series of nucleotides abbreviated as 'A', 'C', 'G', and 'T'.

For example, "ACGAATTCCG" is a **DNA sequence**.

When studying **DNA**, it is useful to identify repeated sequences within the DNA.

Given a string s that represents a **DNA sequence**, return all the **10-letter-long** sequences (substrings) that occur more than once in a DNA molecule. You may return the answer in **any order**.

**Example 1:**

**Input:** s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"

**Output:** ["AAAAACCCCC","CCCCCAAAAA"]

**Example 2:**

**Input:** s = "AAAAAAAAAAAAA"

**Output:** ["AAAAAAAAAA"]

**For example:**

| **Input** | **Result** |
| --- | --- |
| AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT | AAAAACCCCC  CCCCCAAAAA |

**Program:**

def findRepeatedSequences(s):

sequences = {}

result = []

for i in range(len(s) - 9):

seq = s[i:i+10]

sequences[seq] = sequences.get(seq, 0) + 1

if sequences[seq] == 2:

result.append(seq)

return result

s1 = input()

for i in findRepeatedSequences(s1):

print(i)



**Ex. No. : 7.3 Date: 18.05.24**

**Register No.:231901041 Name: RENUGA DEVI**



**American keyboard**

Given an array of strings words, return *the words that can be typed using letters of the alphabet on only one row of American keyboard like the image below*.

In the **American keyboard**:

* the first row consists of the characters "qwertyuiop",
* the second row consists of the characters "asdfghjkl", and
* the third row consists of the characters "zxcvbnm".
* 
* **Example 1:**
* **Input:** words = ["Hello","Alaska","Dad","Peace"]
* **Output:** ["Alaska","Dad"]
* **Example 2:**
* **Input:** words = ["omk"]
* **Output:** []
* **Example 3:**
* **Input:** words = ["adsdf","sfd"]
* **Output:** ["adsdf","sfd"]
* **For example:**

| **Input** | **Result** |
| --- | --- |
| 4  Hello  Alaska  Dad  Peace | Alaska  Dad |

**Program:**

def findWords(words):

row1 = set('qwertyuiop')

row2 = set('asdfghjkl')

row3 = set('zxcvbnm')

result = []

for word in words:

w = set(word.lower())

if w.issubset(row1) or w.issubset(row2) or w.issubset(row3):

result.append(word)

if len(result) == 0:

print("No words")

else:

for i in result:

print(i)

a = int(input())

arr = [input() for i in range(a)]

findWords(arr)



**Ex. No. : 7.4 Date: 18.05.24**

**Register No.: 231901041 Name RENUGA DEVI**



**Print repeated no**

Given an array of integers nums containing n + 1 integers where each integer is in the range [1, n] inclusive.There is only **one repeated number** in nums, return *this repeated number*. Solve the problem using [set](http://118.185.187.137/moodle/mod/resource/view.php?id=734).

**Example 1:**

**Input:** nums = [1,3,4,2,2]

**Output:** 2

**Example 2:**

**Input:** nums = [3,1,3,4,2]

**Output:** 3

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1 3 4 4 2 | 4 |

**Program:**

n =input().split(" ")

n = list(n)

for i in range(len(n)):

for j in range(i+1,len(n)):

if n[i] == n[j]:

print(n[i])

exit(0)

****

**Ex. No. : 7.5 Date: 18.05.24**

**Register No.:231901041 Name: RENUGA DEVI**



**Check Pair**

Given a tuple and a positive integer k, the task is to find the count of distinct pairs in the tuple whose sum is equal to **K**.

**Examples:**

**Input**: t = (5, 6, 5, 7, 7, 8 ), K = 13   
**Output**: 2   
Explanation:   
Pairs with sum K( = 13) are  {(5, 8), (6, 7), (6, 7)}.   
Therefore, distinct pairs with sum K( = 13) are { (5, 8), (6, 7) }.   
Therefore, the required output is 2.

| Input | Result |
| --- | --- |
| 1,2,1,2,5  3 | 1 |
| 1,2  0 | 0 |

**For example:**

**Program:**

def count\_distinct\_pairs(t, K):

distinct\_pairs = set()

for i in range(len(t)):

for j in range(i + 1, len(t)):

if t[i] + t[j] == K:

distinct\_pairs.add((min(t[i], t[j]), max(t[i], t[j])))

return len(distinct\_pairs)

t\_input = input()

t = tuple(map(int, t\_input.split(',')))

K = int(input())

print(count\_distinct\_pairs(t, K))



**08 – Dictionary**

**Ex. No. : 8.1 Date: 25.05.24**

**Register No.:231901041 Name: RENUGA DEVI**



**Sort Dictionary by Values Summation**

Give a dictionary with value lists, sort the keys by summation of values in value list.

**Input** : test\_dict = {‘Gfg’ : [6, 7, 4], ‘best’ : [7, 6, 5]}

**Output** : {‘Gfg’: 17, ‘best’: 18}

**Explanation** : Sorted by sum, and replaced.

**Input** : test\_dict = {‘Gfg’ : [8,8], ‘best’ : [5,5]}

**Output** : {‘best’: 10, ‘Gfg’: 16}

**Explanation** : Sorted by sum, and replaced.

 Sample Input:

2

Gfg 6 7 4

Best 7 6 5

Sample Output

Gfg 17

Best 18

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2  Gfg 6 7 4  Best 7 6 5 | Gfg 17  Best 18 |

**Program:**

n = int(input())

d = {}

for i in range(n):

s = input().split()

d[s[0]] = list(map(int, s[1:]))

d1 = {k: sum(v) for k, v in d.items()}

sorted\_d = dict(sorted(d1.items(), key=lambda x: x[1]))

for k, v in sorted\_d.items():

print(k, v)

****

**Ex. No. : 8.2 . Date: 25.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Student Record**

Create a student dictionary  for n students with the student name as key and their test mark assignment mark and lab mark as values. Do the following computations and display the result.

1.Identify the student with the  highest [average](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5717) score

2.Identify the student who as the highest Assignment marks

3.Identify the student with the Lowest lab marks

4.Identify the student with the lowest [average](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5717) score

Note:

If more than one student has the same score display all the student names

Sample input:

4

James 67 89 56

Lalith 89 45 45

Ram 89 89 89

Sita 70 70 70

Sample Output:

Ram

James Ram

Lalith

Lalith

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  James 67 89 56  Lalith 89 45 45  Ram 89 89 89  Sita 70 70 70 | Ram  James Ram  Lalith  Lalith |

**Program:**

n=int(input())

d={}

for i in range(n):

na=input().split()

d[na[0]]=[int(na[1]),int(na[2]),int(na[3])]

l=int(na[3])

h=0

for i in d:

if h< sum(d[i]):

h=sum(d[i])

j=i

h1=sum(d[i])

print(j)

h=0

for i in d:

if(h<d[i][1]):

h=d[i][1]

j=i

for i in d:

if(h==d[i][1]):

print(i,end=" ")

l1=[]

k=[]

print()

for i in d:

if(l>d[i][2]):

l=d[i][2]

j=i

for i in d:

if(l==d[i][2]):

l1.append(i)

for i in range(-1,-len(l1)-1,-1):

print(l1[i],end=" ")

print()

for i in d:

if h1> sum(d[i]):

h1=sum(d[i])

j=i

print(j)

****

**Ex. No. : 8.3 Date: 25.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



[**Scramble Score**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5780)

In the game of Scrabble™, each letter has points associated with it. The total score of a word is the sum of the scores of its letters. More common letters are worth fewer points while less common letters are worth more points.

Write a program that computes and displays the Scrabble™ score for a word. Create a dictionary that maps from letters to point values. Then use the dictionary to compute the score.

A Scrabble™ board includes some squares that multiply the value of a letter or the value of an entire word. We will ignore these squares in this exercise.

The points associated with each letter are shown below:

Points Letters

1 A, E, I, L, N, O, R, S, T and U

2 D and G

3 B, C, M and P

4 F, H, V, W and Y

5 K

8 J and X

10 Q and Z

[Sample](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5127) Input

REC

[Sample](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5127) Output

REC is worth 5 points.

**For example:**

| **Input** | **Result** |
| --- | --- |
| REC | REC is worth 5 points. |

**Program:**

def calculate\_scrabble\_score(word):

# Dictionary mapping letters to points

letter\_points = {

'A': 1, 'B': 3, 'C': 3, 'D': 2, 'E': 1, 'F': 4, 'G': 2, 'H': 4,

'I': 1, 'J': 8, 'K': 5, 'L': 1, 'M': 3, 'N': 1, 'O': 1, 'P': 3,

'Q': 10, 'R': 1, 'S': 1, 'T': 1, 'U': 1, 'V': 4, 'W': 4, 'X': 8,

'Y': 4, 'Z': 10

}

score = 0

for letter in word:

letter = letter.upper()

score += letter\_points.get(letter, 0) # Add the points for each letter, defaulting to 0 if not found

return score

word=input()

score = calculate\_scrabble\_score(word)

print(f"{word} is worth {score} points.")

****

**Ex. No. : 8.4 Date: 25.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Uncommon words**

A sentence is a string of single-space separated words where each word consists only of lowercase letters.A word is uncommon if it appears exactly once in one of the sentences, and does not appear in the other sentence.

Given two sentences s1 and s2, return a list of all the uncommon words. You may return the answer in any order.

Example 1:

Input: s1 = "this apple is sweet", s2 = "this apple is sour"

Output: ["sweet","sour"]

Example 2:

Input: s1 = "apple apple", s2 = "banana"

Output: ["banana"]

 Constraints:

1 <= s1.length, s2.length <= 200

s1 and s2 consist of lowercase English letters and spaces.

s1 and s2 do not have leading or trailing spaces.

All the words in s1 and s2 are separated by a single space.

Note:

Use dictionary to solve the problem

**For example:**

| **Input** | **Result** |
| --- | --- |
| this apple is sweet  this apple is sour | sweet sour |

**Program:**

s1 = input().split()

s2 = input().split()

d = {}

for i in s1:

if i not in d:

d[i] = 1

else:

d[i] += 1

for i in s2:

if i not in d:

d[i] = 1

else:

d[i] += 1

for i in d:

if d[i] == 1:

print(i, end=" ")

****

**Ex. No. : 8.5 Date: 25.05.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Winner of Election**

Given an array of names of candidates in an election. A candidate name in the array represents a vote cast to the candidate. Print the name of candidates received Max vote. If there is tie, print a lexicographically smaller name.

**Examples:**

Input :  votes[] = {"john", "johnny", "jackie",

                    "johnny", "john", "jackie",

                    "jamie", "jamie", "john",

                    "johnny", "jamie", "johnny",

                    "john"};

Output : John

We have four Candidates with name as 'John', 'Johnny', 'jamie', 'jackie'. The candidates John and Johny get maximum votes. Since John is alphabetically smaller, we print it. Use dictionary to solve the above problem

**Sample Input:**

10

John

John

Johny

Jamie

Jamie

Johny

Jack

Johny

Johny

Jackie

**Sample Output:**

Johny

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10  John  John  Johny  Jamie  Jamie  Johny  Jack  Johny  Johny  Jackie | Johny |

**Program:**

n=int(input())

d={}

for i in range(n):

s=input()

if s not in d:

d[s]=1

else:

d[s]+=1

h=0

for i in d:

if h<d[i]:

h=d[i]

j=i

print(j)

****

**09- Functions**

**Ex. No. : 9.1 Date: 01.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Christmas Discount**

An e-commerce company plans to give their customers a special discount for Christmas.

They are planning to offer a flat discount. The discount value is calculated as the sum of all the prime digits in the total bill amount.

Write an python code to find the discount value for the given total bill amount.

**Constraints**

1 <= orderValue< 10e100000

Input

The input consists of an integer orderValue, representing the total bill amount.

Output

Print an integer representing the discount value for the given total bill amount.

Example Input

578

Output

12

**For example:**

| **Test** | **Result** |
| --- | --- |
| print(christmasDiscount(578)) | 12 |

**Program:**

def is\_prime\_digit(digit):

return digit in [2,3,5,7]

def christmasDiscount(n):

s=discount=0

prime\_digitis=[2,3,5,7]

for digit in str(n):

digit=int(digit)

if is\_prime\_digit(digit):

discount+=digit

return discount

****

**Ex. No. : 9.2 Date: 01.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Check Product of Digits**

Write a code to check whether product of digits at even places is divisible by sum of digits at odd place of a positive integer.

Input Format:

Take an input integer from stdin.

Output Format:

Print TRUE or FALSE.

Example Input:

1256

Output:

TRUE

Example Input:

1595

Output:

FALSE

For example:

| **Test** | **Result** |
| --- | --- |
| print(productDigits(1256)) | True |
| print(productDigits(1595)) | False |

**Program:**

def productDigits(n):

a=n

temp=[]

list1=[]

list2=[]

rem=0

while a!=0:

rem=a%10

temp.append(rem)

a=a//10

for i in range(len(temp)):

if(i+1)%2==0:

list1.append(temp[i])

else:

list2.append(temp[i])

pro=1

sum=0

for i in list1:

sum+=i

for i in list2:

pro\*=i

if pro%sum==0:

return True

else:

return False



**Ex. No. : 9.3 Date: 01.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Abundant Number**

An abundant number is a number for which the sum of its proper divisors is greater than the number itself. Proper divisors of the number are those that are strictly lesser than the number.

**Input Format**:

Take input an integer from stdin

**Output Format:**

Return Yes if given number is Abundant. Otherwise, print No

**Example input:**

12

**Output**:

Yes

Explanation

The proper divisors of 12 are: 1, 2, 3, 4, 6, whose sum is 1 + 2 + 3 + 4 + 6 = 16. Since sum of proper divisors is greater than the given number, 12 is an abundant number.

**Example input:**

13

**Output**:

No

**Explanation**

The proper divisors of 13 is: 1, whose sum is 1. Since sum of proper divisors is not greater than the given number, 13 is not an abundant number.

For example:

Test Result

print(abundant(12)) Yes

print(abundant(13)) No

**Program:**

def abundant(number):

d\_s=sum([divisor for divisor in range(1,number) if number % divisor == 0])

if d\_s>number:

return"Yes"

else:

return "No"

****

**Ex. No. : 9.4 Date: 01.06.24**

**Register No.: 231901041 Name RENUGA DEVI**



**Ugly number**

A number is considered to be ugly if its only prime factors are 2, 3 or 5.

[1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, …] is the sequence of ugly numbers.

Task:

complete the function which takes a number n as input and checks if it's an ugly number. return ugly if it is ugly, else return not ugly

Hint:

An ugly number U can be expressed as: U = 2^a \* 3^b \* 5^c, where a, b and c are nonnegative integers.

**For example:**

| **Test** | **Result** |
| --- | --- |
| print(checkUgly(6)) | ugly |
| print(checkUgly(21)) | not ugly |

**Program:**

def checkUgly(n):

if n <= 0:

return "not ugly”

while n % 2 == 0:

n //= 2

while n % 3 == 0:

n //= 3

while n % 5 == 0:

n //=5

return "ugly" if n == 1 else "not ugly"



**Ex. No. : 9.5 Date: 01.06.24**

**Register No.: 231901041 Name RENUGA DEVI**



**Automorphic number or not**

An automorphic number is a number whose square ends with the number itself. For example, 5 is an automorphic number because 5\*5 =25. The last digit is 5 which same as the given number.

If the number is not valid, it should display “Invalid input”.

If it is an automorphic number display “Automorphic” else display “Not Automorphic”.

Input Format:

Take a Integer from Stdin

Output Format:

Print Automorphic if given number is Automorphic number, otherwise Not Automorphic

Example input: 5 Output: Automorphic Example input: 25 Output: Automorphic Example input: 7 Output: Not Automorphic

**For example:**

Test Result

print(automorphic(5)) Automorphic

**Program:**

def automorphic(n):

if(n<0):

return "Invalid input"

square = n \* n

n\_s=str(n)

s\_s=str(square)

if s\_s.endswith(n\_s):

return "Automorphic"

else:

return "Not Automorphic"

****

**10 - Searching & Sorting**

**Ex. No. : 10.1 Date: 01.06.24**

**Register No.: 231901041 Name RENUGA DEVI**



**Bubble Sort**

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. You read an list of numbers. You need to arrange the elements in ascending order and print the result. The sorting should be done using bubble sort.

Input Format: The first line reads the number of elements in the array. The second line reads the array elements one by one.  
  
Output Format: The output should be a sorted list.

**For example**:

| Input | Result |
| --- | --- |
| 6  3 4 8 7 1 2 | 1 2 3 4 7 8 |
| 5  4 5 2 3 1 | 1 2 3 4 5 |

**Program:**

n=int(input())

k=[int(x) for x in input().split()]

k.sort()

for i in k:

print(i,end=' ')

****

**Ex. No. : 10.2 Date: 01.06.24**

**Register No.: 231901041 Name RENUGA DEVI**



**Peak Element**

Given an [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068), find peak element in it. A peak element is an element that is greater than its neighbors.

An element a[i] is a peak element if

A[i-1] <= A[i] >=a[i+1] for middle elements. [0<i<n-1]

A[i-1] <= A[i] for last element [i=n-1]

A[i]>=A[i+1] for first element [i=0]

**Input Format**

The first line contains a single integer n , the length of A .  
The second line contains n space-separated integers,A[i].

**Output Format**

**Print** peak numbers separated by space.

**Sample Input**

5

8 9 10 2 6

**Sample Output**

10 6

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4  12 3 6 8 | 12 8 |

**Program:**

a=int(input())

lst1=[str(x) for x in input().split(" ")]

lst2=[]

lst=[]

g=0

for i in lst1:

if i.isdigit():

g=int(i)

lst.append(g)

for i in range(0,a):

if(i==0):

if(lst[i]>=lst[i+1]):

lst2.append(lst[i])

elif(i>0 and i<a-2):

if(lst[i]>=lst[i-1] and lst[i]>=lst[i+1]):

lst2.append(lst[i])

elif(i==a-1):

if(lst[i]>=lst[i-1]):

lst2.append(lst[i])

for i in lst2:

print(i,end=" ")



**Ex. No. : 10.3 Date: 01.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Merge Sort**

Write a Python program to sort a list of elements using the merge sort algorithm.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  6 5 4 3 8 | 3 4 5 6 8 |

**Program:**

def merge\_sort(arr):

if len(arr) > 1:

mid = len(arr) // 2

left\_half = arr[:mid]

right\_half = arr[mid:]

merge\_sort(left\_half)

merge\_sort(right\_half)

i = j = k = 0

while i < len(left\_half) and j < len(right\_half):

if left\_half[i] < right\_half[j]:

arr[k] = left\_half[i]

i += 1

else:

arr[k] = right\_half[j]

j += 1

k += 1

while i < len(left\_half):

arr[k] = left\_half[i]

i += 1

k += 1

while j < len(right\_half):

arr[k] = right\_half[j]

j += 1

k += 1

def main():

n = int(input())

arr = list(map(int, input().split()))

merge\_sort(arr)

for num in arr:

print(num, end=" ")

if \_\_name\_\_ == "\_\_main\_\_":

main()



**Ex. No. : 10.4 Date: 01.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Sum of Two numbers**

An list contains N numbers and you want to determine whether two of the numbers sum to a given number K. For example, if the input is 8, 4, 1, 6 and K is 10, the answer is yes (4 and 6). A number may be used twice.

**Input Format**

The first line contains a single integer n , the length of list

The second line contains n space-separated integers, list[i].

The third line contains integer k.

**Output Format**

Print Yes or No.

**Sample Input**

7

0 1 2 4 6 5 3

1

**Sample Output**

Yes

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  8 9 12 15 3  11 | Yes |
| 6  2 9 21 32 43 43 1  4 | No |

**Program:**

n=int(input())

a=[int(x) for x in input().split()]

k=int(input())

flag=0

if len(a)!=n:

print("No")

flag=1

for i in a:

for j in a:

if i+j==k and flag==0:

flag=1

print("Yes")

break

if flag==0:

print("No")



**Ex. No. : 10.5 Date: 01.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**Frequency of Elements**

To find the frequency of numbers in a list and display in sorted order.

**Constraints:**

1<=n, arr[i]<=100

**Input:**

1 68 79 4 90 68 1 4 5

**output:**

 1 2

 4 2

 5 1

 68 2

 79 1

90 1

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4 3 5 3 4 5 | 3 2  4 2  5 2 |
|  |  |

**Program:**

lst5=[int(x) for x in input().split(" ")]

lst=sorted(list(set(lst5)))

c=0

for i in lst:

c=0

for j in lst5:

if(i==j):

c=c+1

print("%d %d"%(i,c))

****

**11**- **EXCEPTION HANDLING**

**Ex. No. : 11.1 Date: 02.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**EXCEPTION HANDLING**

To find whether a digit lies in the specified range(1-100). Handling exceptions for invalid inputs and out-of-range numbers .

Input Format:

User inputs a number.

Output Format:

Confirm the input or print an error message if it's invalid or out of range.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1 | Valid input. |
| 101 | Error: Number out of allowed range |
| rec | Error: invalid literal for int() |

**Program:**

try:

a=input()

if(int(a)>0 and int(a)<101):

print("Valid input.")

else:

print("Error: Number out of allowed range")

except:

print("Error: invalid literal for int()")



**Ex. No. : 11.2 Date: 02.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**EXCEPTION HANDLING**

Write a Python program that performs division and modulo operations on two numbers provided by the user. Handle division by zero and non-numeric inputs.

Input Format:

Two lines of input, each containing a number.

Output Format:

Print the result of division and modulo operation, or an error message if an exception occurs.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10  2 | Division result: 5.0  Modulo result: 0 |
| 7  3 | Division result: 2.3333333333333335  Modulo result: 1 |
| 8  0 | Error: Cannot divide or modulo by zero. |

**Program:**

try:

a=input()

b=input()

c=int(a)/int(b)

d=int(a)%int(b)

except ZeroDivisionError:

print("Error: Cannot divide or modulo by zero.")

except:

print("Error: Non-numeric input provided.")

else:

print("Division result:",c)

print("Modulo result:",d)

****

**Ex. No. : 11.3 Date: 02.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**EXCEPTION HANDLING**

Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

**Input Format:** A single line input representing the user's age.

**Output Format:** Print a message based on the age or an error if the input is invalid.

**For example:**

| **Input** | **Result** |
| --- | --- |
| twenty | Error: Please enter a valid age. |
| 25 | You are 25 years old. |
| -1 | Error: Please enter a valid age. |

**Program:**

try:

a=input()

if int(a)>=0:

print("You are",a,"years old.")

else:

print("Error: Please enter a valid age.")

except:

print("Error: Please enter a valid age.")



**Ex. No. : 11.4 Date: 02.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**EXCEPTION HANDLING**

Develop a Python program that safely calculates the square root of a number provided by the user. Handle exceptions for negative inputs and non-numeric inputs.

Input Format:

User inputs a number.

Output Format:

Print the square root of the number or an error message if an exception occurs.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 16 | The square root of 16.0 is 4.00 |
| -4 | Error: Cannot calculate the square root of a negative number. |
| rec | Error: could not convert string to float |

**Program:**

import math

try:

n=input()

n=float(n)

if n < 0:

print("Error: Cannot calculate the square root of a negative number.")

else:

r= math.sqrt(n)

print("The square root of {} is {:.2f}".format(n, r))

except ValueError:

print("Error: could not convert string to float")



**Ex. No. : 11.5 Date: 02.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**EXCEPTION HANDLING**

Develop a Python program that safely performs division between two numbers provided by the user. Handle exceptions like division by zero and non-numeric inputs.

**Input Format:** Two lines of input, each containing a number.

**Output Format:** Print the result of the division or an error message if an exception occurs.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10  2 | 5.0 |
| 10  0 | Error: Cannot divide or modulo by zero. |
| ten  5 | Error: Non-numeric input provided. |

**Program:**

try:

a=input()

b=input()

c=float(a)/float(b)

except ZeroDivisionError:

print("Error: Cannot divide or modulo by zero.")

except:

print("Error: Non-numeric input provided.")

else:

print(c)



**12-MODULES**

**Ex. No. : 12.1 Date: 07.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**MODULES - REPRESENTING UNIQUE PAIRS**

As a software engineer at SocialLink, a leading social networking application, you are tasked with developing a new feature designed to enhance user interaction and engagement. The company aims to introduce a system where users can form connections based on shared interests and activities. One of the feature's components involves analyzing pairs of users based on the activities they've participated in, specifically looking at the numerical difference in the number of activities each user has participated in.

Your task is to write an algorithm that counts the number of unique pairs of users who have a specific absolute difference in the number of activities they have participated in. This algorithm will serve as the backbone for a larger feature that recommends user connections based on shared participation patterns.

Problem Statement

Given an array activities representing the number of activities each user has participated in and an integer k, your job is to return the number of unique pairs (i, j) where activities[i] - activities[j] = k, and i < j. The absolute difference between the activities should be exactly k.

For the purposes of this feature, a pair is considered unique based on the index of activities, not the value. That is, if there are two users with the same number of activities, they are considered distinct entities.

Input Format

The first line contains an integer, n, the size of the array nums.

The second line contains n space-separated integers, nums[i].

The third line contains an integer, k.

Output Format

Return a single integer representing the number of unique pairs (i, j)

where | nums[i] - nums[j] | = k and i < j.

Constraints:

1 ≤ n ≤ 105

-104 ≤ nums[i] ≤ 104

0 ≤ k ≤ 104

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5  1 3 1 5 4  0 | 1 |
| 4  1 2 2 1  1 | 4 |

**Program:**

a=int(input())

b=input().split()

c=int(input())

co=0

l=[int(b) for b in b]

for i in range(0,a):

for j in range(0,a):

if abs(l[i]-l[j])==c and i<j:

co+=1

print(co)



**Ex. No. : 12.2 Date: 07.06.24**

**Register No.: 231901041 Name RENUGA DEVI**



**MODULES-CALCULATING AVERAGE**

Dr. John Wesley maintains a spreadsheet with student records for academic evaluation. The spreadsheet contains various data fields including student IDs, marks, class names, and student names. The goal is to develop a system that can calculate the average marks of all students listed in the spreadsheet.

Problem Statement:

Create a Python-based solution that can parse input data representing a list of students with their respective marks and other details, and compute the average marks. The input may present these details in any order, so the solution must be adaptable to this variability.

Input Format:

The first line contains an integer N, the total number of students.

The second line lists column names in any order (ID, NAME, MARKS, CLASS).

The next N lines provide student data corresponding to the column headers.

Output Format:

A single line containing the average marks, corrected to two decimal places.

Constraints:

1≤N≤100

Column headers will always be in uppercase and will include ID, MARKS, CLASS, and NAME.

Marks will be non-negative integers.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3  ID NAME MARKS CLASS  101 John 78 Science  102 Doe 85 Math  103 Smith 90 History | 84.33 |
| 3  MARKS CLASS NAME ID  78 Science John 101  85 Math Doe 102  90 History Smith 103 | 84.33 |

**Program:**

import math

a = int(input())

n=a

b = input().split()

s = 0

p = b.index("MARKS")

while a!=0:

c = input().split()

s += int(c[p])/n

a-=1

print(f"{s:.2f}")



**Ex. No. : 12.3 Date: 07.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**MODULES-USING DICTIONARY**

Rose manages a personal library with a diverse collection of books. To streamline her library management, she needs a program that can categorize books based on their genres, making it easier to find and organize her collection.

Problem Statement:

Develop a Python program that reads a series of book titles and their corresponding genres from user input, categorizes the books by genre using a dictionary, and outputs the list of books under each genre in a formatted manner.

Input Format:

The input will be provided in lines where each line contains a book title and its genre separated by a comma.

Input terminates with a blank line.

Output Format:

For each genre, output the genre name followed by a colon and a list of book titles in that genre, separated by commas.

Constraints:

Book titles and genres are strings.

Book titles can vary in length but will not exceed 100 characters.

Genres will not exceed 50 characters.

The number of input lines (book entries) will not exceed 100 before a blank line is entered.

**For example:**

| **Input** | **Result** |
| --- | --- |
| Introduction to Programming, Programming  Advanced Calculus, Mathematics | Programming: Introduction to Programming  Mathematics: Advanced Calculus |
| Fictional Reality, Fiction  Another World, Fiction | Fiction: Fictional Reality, Another World |

**Program:**

d={}

d = {}

while True:

try:

book = input().split(',')

if len(book) < 2:

continue

book\_name = book[0].strip()

category = book[1].strip()

if category in d:

d[category].append(book\_name)

else:

d[category] = [book\_name]

except EOFError:

break

for k, v in d.items():

print(f"{k}: ", end='')

print(', '.join(v))



**Ex. No. : 12.4 Date: 07.06.24**

**Register No.: 231901041 Name: RENUGA DEVI**



**MODULE-POWER OF FOUR**

Given an integer n, print *true if it is a power of four. Otherwise, print false*.

An integer n is a power of four, if there exists an integer x such that n == 4x.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 16 | True |
| 5 | False |

**Program:**

a=int(input())

c=0

for i in range(a):

if a==2\*\*i:

c+=1

if c==1:

print("True")

else:

print("False")



**Ex. No. : 12.5 Date: 07.06.24**

**Register No.: 231901041 Name : RENUGA DEVI**



**MODULES-DETERMINING THE TOTAL REVENUE**

Raghu owns a shoe shop with a varying inventory of shoe sizes. The shop caters to multiple customers who have specific size requirements and are willing to pay a designated amount for their desired shoe size. Raghu needs an efficient system to manage his inventory and calculate the total revenue generated from sales based on customer demands.

Problem Statement:

Develop a Python program that manages shoe inventory and processes sales transactions to determine the total revenue generated. The program should handle inputs of shoe sizes available in the shop, track the number of each size, and match these with customer purchase requests. Each transaction should only proceed if the desired shoe size is in stock, and the inventory should update accordingly after each sale.

Input Format:

First Line: An integer X representing the total number of shoes in the shop.

Second Line: A space-separated list of integers representing the shoe sizes in the shop.

Third Line: An integer N representing the number of customer requests.

Next N Lines: Each line contains a pair of space-separated values:

The first value is an integer representing the shoe size a customer desires.

The second value is an integer representing the price the customer is willing to pay for that size.

Output Format:

Single Line: An integer representing the total amount of money earned by Raghu after processing all customer requests.

Constraints:

1≤X≤1000 — Raghu's shop can hold between 1 and 1000 shoes.

Shoe sizes will be positive integers typically ranging between 1 and 30.

1≤N≤1000 — There can be up to 1000 customer requests in a single batch.

The price offered by customers will be a positive integer, typically ranging from $5 to $100 per shoe.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10  2 3 4 5 6 8 7 6 5 18  6  6 55  6 45  6 55  4 40  18 60  10 50 | 200 |
| 5  5 5 5 5 5  5  5 10  5 10  5 10  5 10  5 10 | 50 |

**Program:**

a=int(input())

b=input().split()

c=int(input())

s=0

for i in range(c):

l1=[]

l1=input().split()

if l1[0] in b:

s+=int(l1[1])

b.remove(l1[0])

print(s)

