**RAJALAKSHMI ENGINEERING COLLEGE**

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

1

**RAJALAKSHMI NAGAR, THANDALAM – 602 105**



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



.

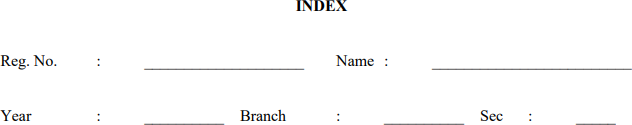
**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

2

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

3



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **No.** | **Date** | **Title** | **Page No.** | **Teacher’s Signature / Remarks** |
| **Introduction to python-Variables-Datatypes-Input/Output-Formatting** | | | | |
| 1.1 |  | Converting Input Strings |  |  |
| 1.2 |  | Gross salary |  |  |
| 1.3 |  | Square Root |  |  |
| 1.4 |  | Gain percent |  |  |
| 1.5 |  | Deposits |  |  |
| 1.6 |  | Carpenter |  |  |
| **Operators in Python** | | | | |
| 2.1 |  | Widgets and Gizmos |  |  |
| 2.2 |  | Doll Sings |  |  |
| 2.3 |  | Birthday party |  |  |
| 2.4 |  | Hamming Weight |  |  |
| 2.5 |  | Compound Interest |  |  |
| 2.6 |  | Eligible to donate blood |  |  |
| 2.7 |  | C or D |  |  |
| 2.8 |  | Troy Battle |  |  |
| 2.9 |  | Tax and Tip |  |  |
| 2.10 |  | Return last digit of the given number |  |  |
| **Selection Structures in Python** | | | | |
| 3.1 |  | Admission eligibility |  |  |
| 3.2 |  | Classifying triangles |  |  |
| 3.3 |  | Electricity Bill |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.4 |  | IN/OUT |  |  |
| 3.5 |  | Vowel or Constant |  |  |
| 3.6 |  | Leap Year |  |  |
| 3.7 |  | Month name to Days |  |  |
| 3.8 |  | Pythagorean triple |  |  |
| 3.9 |  | Second Last Digit |  |  |
| 3.10 |  | Chinese Zodiac |  |  |
| **Algorithmic Approach: Iteration Control Structures** | | | | |
| 4.1 |  | Factors of a Number |  |  |
| 4.2 |  | Non-Repeated Digits Count |  |  |
| 4.3 |  | Prime Checking |  |  |
| 4.4 |  | Next Perfect Square |  |  |
| 4.5 |  | Nth Fibonacci |  |  |
| 4.6 |  | Disarium Number |  |  |
| 4.7 |  | Sum of Series |  |  |
| 4.8 |  | Unique Digits Count |  |  |
| 4.9 |  | Product of single digits |  |  |
| 4.10 |  | Perfect Square After adding One |  |  |
| **Strings in Python** | | | | |
| 5.1 |  | Count chars |  |  |
| 5.2 |  | Decompress the String |  |  |
| 5.3 |  | First N Common Characters |  |  |
| 5.4 |  | Remove Characters |  |  |
| 5.5 |  | Remove Palindrome Words |  |  |
| 5.6 |  | Return Second Word in Uppercase |  |  |
| 5.7 |  | Reverse String |  |  |
| 5.8 |  | String characters balance Test |  |  |
| 5.9 |  | Unique Names |  |  |
| 5.10 |  | Username Domain Extension |  |  |
| **List in Python** | | | | |
| 6.1 |  | Monotonic array |  |  |
| 6.2 |  | Check pair with difference k . |  |  |
| 6.3 |  | Count Elements |  |  |
| 6.4 |  | Distinct Elements in an Array |  |  |

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6.5 |  | Element Insertion |  |  |
| 6.6 |  | Find the Factor |  |  |
| 6.7 |  | Merge list |  |  |
| 6.8 |  | Merge Two Sorted Arrays Without  Duplication |  |  |
| 6.9 |  | Print Element Location |  |  |
| 6.10 |  | Strictly increasing |  |  |
| **Tuples & Set** | | | | |
| 7.1 |  | Binary String |  |  |
| 7.2 |  | Check Pair |  |  |
| 7.3 |  | DNA Sequence |  |  |
| 7.4 |  | Print repeated no |  |  |
| 7.5 |  | Remove repeated |  |  |
| 7.6 |  | malfunctioning keyboard |  |  |
| 7.7 |  | American keyboard |  |  |
| **Dictionary** | | | | |
| 8.1 |  | Uncommon Words |  |  |
| 8.2 |  | Sort Dictionary By Values Summation |  |  |
| 8.3 |  | Winner Of Election |  |  |
| 8.4 |  | Student Record |  |  |
| 8.5 |  | Scramble Score |  |  |
| **Functions** | | | | |
| 9.1 |  | Abundant Number |  |  |
| 9.2 |  | Automorphic number or not |  |  |
| 9.3 |  | Check Product of Digits |  |  |
| 9.4 |  | Christmas Discount |  |  |
| 9.5 |  | Coin Change |  |  |
| 9.6 |  | Difference Sum |  |  |
| 9.7 |  | Ugly number |  |  |
|  | | | | |

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Searching & Sorting** | | | | |
| 10.1 |  | Merge Sort |  |  |
| 10.2 |  | Bubble Sort |  |  |
| 10.3 |  | Peak Element |  |  |
| 10.4 |  | Binary Search |  |  |
| 10.5 |  | Frequency of Numbers |  |  |

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

6

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

7

**01 - Introduction to Python-Variables-Datatypes Input/Output-Formatting**

**Ex. No. : 1.1 Date: 14/03/2024**

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

8

**Register No.: 231401059 Name: LEKHA L**

[**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'> |

**Answer:**

a=int(input()) b=float(input()) print(a,type(a),sep=",")

print(round(b,1),type(b),sep=",")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 10  10.9 | 10,<class 'int'> 10.9,<class 'float'> | 10,<class 'int'> 10.9,<class 'float'> |  |
|  | 12  12.5 | 12,<class 'int'> 12.5,<class 'float'> | 12,<class 'int'> 12.5,<class 'float'> |  |
|  | 89  7.56 | 89,<class 'int'>  7.6,<class 'float'> | 89,<class 'int'>  7.6,<class 'float'> |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  | |
|  | 55000  56.2 | 55000,<class 'int'> 56.2,<class 'float'> | 55000,<class 'int'> 56.2,<class 'float'> |  |  |
|  | 2541  2541.679 | 2541,<class 'int'> 2541.7,<class 'float'> | 2541,<class 'int'> 2541.7,<class 'float'> |  | |

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

9

**Ex. No. : 1.2 Date: 14/03/2024**

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

10

**Register No.: 231401059 Name: LEKHA L**

[**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 |

**Answer:** a=int(input()) b=40/100\*a c=20/100\*a d=a+b+c print(round(d))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 10000 | 16000 | 16000 |  |
|  | 20000 | 32000 | 32000 |  |
|  | 28000 | 44800 | 44800 |  |
|  | 5000 | 8000 | 8000 |  |

**Ex. No. : 1.3 Date: 14/03/2024**

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

11

**Register No.: 231401059 Name: LEKHA L**

**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 |

Answer:

import math a=float(input()) b=math.sqrt(a) print(format(b,'.3f'))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 8.00 | 2.828 | 2.828 |  |
|  | 14.00 | 3.742 | 3.742 |  |
|  | 4.00 | 2.000 | 2.000 |  |
|  | 487 | 22.068 | 22.068 |  |

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

12

**Ex. No. : 1.4 Date: 14/03/2024**

**Register No.: 231401059 Name: LEKHA L**

[**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. |

**Answer:**

a=int(input())

.

b=int(input()) c=int(input()) g=c-(a+b) gp=g/(a+b)\*100

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

13

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 10000  250  15000 | 46.34 is the gain percent. | 46.34 is the gain percent. |  |
|  | 45500  500  60000 | 30.43 is the gain percent. | 30.43 is the gain percent. |  |
|  | 5000  0  7000 | 40.00 is the gain percent. | 40.00 is the gain percent. |  |
|  | 12500  5000  18000 | 2.86 is the gain percent. | 2.86 is the gain percent. |  |

**Ex. No. : 1.5 Date: 14/03/2024**

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

14

**Register No.: 231401059 Name: LEKHA L**

[**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. |

**Answer:**

x=int(input()) y=int(input()) a=x\*0.10 b=y\*0.25 c=a+b

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

.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 20  20 | Your total refund will be $7.00. | Your total refund will be $7.00. |  |
|  | 11  22 | Your total refund will be $6.60. | Your total refund will be $6.60. |  |
|  | 123  200 | Your total refund will be $62.30. | Your total refund will be $62.30. |  |
|  | 76  38 | Your total refund will be $17.10. | Your total refund will be $17.10. |  |

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

15

**Ex. No. : 1.6 Date: 14/03/2024**

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

16

**Register No.: 231401059 Name: LEKHA L**

[**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 example:**

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

**Answer:**

.

x=int(input()) a=(x-(50\*10))

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

17

b=a/(130) b=abs(b)

print("weekdays",(format(b+10,'.2f')))

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 450 | weekdays 10.38  weekend 0.38 | weekdays 10.38  weekend 0.38 |  |
|  | 500 | weekdays 10.00  weekend 0.00 | weekdays 10.00  weekend 0.00 |  |
|  | 10000 | weekdays 83.08  weekend 73.08 | weekdays 83.08  weekend 73.08 |  |
|  | 6789 | weekdays 58.38  weekend 48.38 | weekdays 58.38  weekend 48.38 |  |

**02-Operations in python**

**Ex. No. : 2.1 Date:29/03/24**

**Register No: 231401059 Name: LEKHA L**

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

10

20

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

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

**For example:**



**Program:**

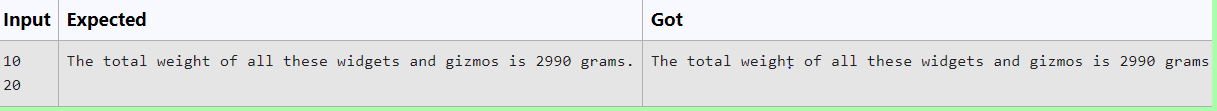
widget=75

gizmo=112

x=eval(input())

y=eval(input())

print("The total weight of all these widgets and gizmos is", x\*widget+y\*gizmo, "grams.")



**Ex. No. : 2.2 Date: 29/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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:**



**Program:**

x=int(input())

if x!=0 and x<100:

if x%2==0:

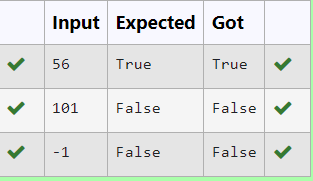
print("True")

else:

print("False")

else:

print("False")



**Ex. No. : 2.3 Date:29/03/24**

**Register No.: 231401059 Name:LEKHA L**

[**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**:



**Program:**

N=int(input())

for i in range(4):

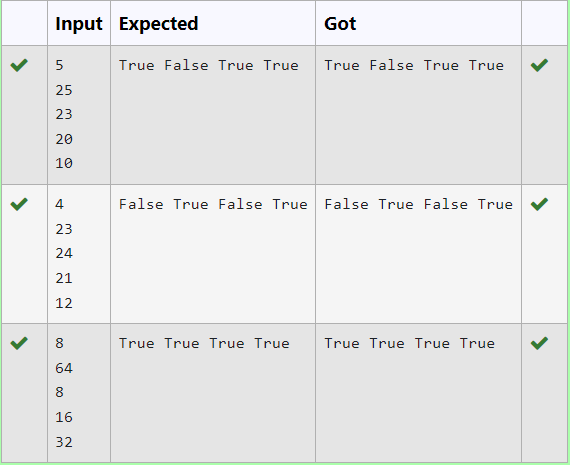
x=int(input())

if x%N==0:

print("True", end=" ")

else:

print("False", end=" ")



**Ex. No. : 2.4 Date:29/03/24**

**Register No.:231401059 Name:LEKHA L**

**Binary form**

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:**



**Program**:

num = int(input())

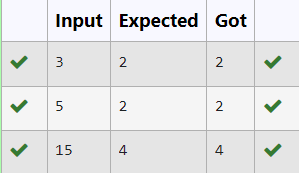
count = 0

while num:

num &= num - 1

count += 1

print(count)



**Ex. No. : 2.5 Date:29/03/24**

**Register No.: 231401059 Name:LEKHA L**

[**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.

**For example:**



**Program:**

x=int(input())

a=4/100\*x

b=x+a

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

c=4/100\*b

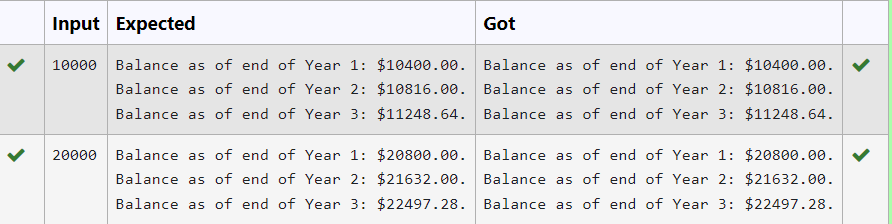
d=c+b

print("Balance as of end of Year 2:", "${:.2f}.".format(d))

e=4/100\*d

f=e+d

print("Balance as of end of Year 3:", "${:.2f}.".format(f))



**Ex. No. : 2.6 Date:29/03/24**

**Register No.:231401059 Name:LEKHA L**

**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:**

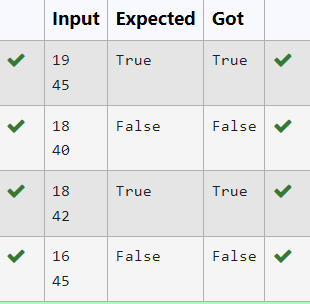


**Program:**

age=int(input())

weight=eval(input())

print(age>=18 and weight>40)



**Ex. No. : 2.7 Date:29/03/24**

**Register No.: 231401059 Name:LEKHA L**

**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:**



**Program:**

x=int(input())

if 0<=x<=1:

if x==0:

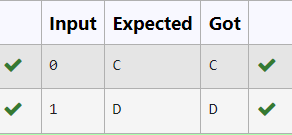
print(chr(67))

else:

print(chr(68))

else:

print("Invalid")



**Ex. No. : 2.8 Date:29/03/24**

**Register No.: 231401059 Name: LEKHA L**

**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:**



**Program:**

weapons=int(input())

soldiers=int(input())

x=weapons%3

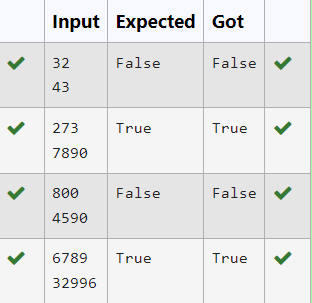
y=soldiers%2

if x==0 and y==0:

print("True")

else:

print("False")



**Ex. No. : 2.9 Date:29/03/24**

**Register No.: 231401059 Name:LEKHA L**

[**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:**



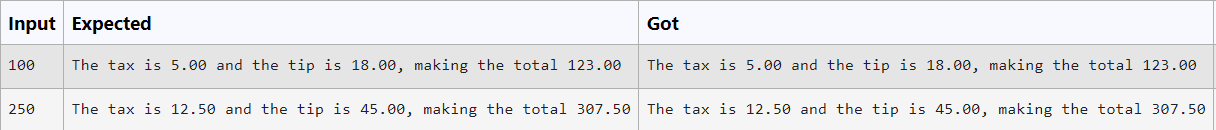
**Program:**

x=int(input())

tax=5/100\*x

tip=18/100\*x

print("The tax is", "{:.2f}".format(tax), "and the tip is", "{:.2f},".format(tip), "making the total", "{:.2f}".format(tip+tax+x))



**Ex. No. : 2.10 Date:29/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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 :



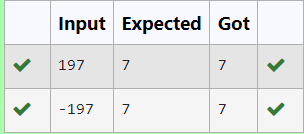
Program:

x=int(input())

y=abs(x)

z=y%10

print(z)



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

**Ex. No. : 3.1 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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())

d=a+b+c

if a>=65 and b>=55 and c>=50 or d>=180:

print("The candidate is eligible")

else:

print("The candidate is not eligible")

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 70  60  80 | The candidate is eligible | The candidate is eligible |  |
|  | 50  80  80 | The candidate is eligible | The candidate is eligible |  |
|  | 50  60  40 | The candidate is not eligible | The candidate is not eligible |  |
|  | 20  10  25 | The candidate is not eligible | The candidate is not eligible |  |

**Ex. No. : 3.2 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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

**For example:**

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

**PROGRAM:**

a=int(input())

b=int(input())

c=int(input())

if a==b==c:

print("That's a equilateral triangle")

elif a==b!=c or a==c!=b or c==b!=a:

print("That's a isosceles triangle")

else:

print("That's a scalene triangle")

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 60  60  60 | That's a equilateral triangle | That's a equilateral triangle |  |
|  | 40  40  80 | That's a isosceles triangle | That's a isosceles triangle |  |
|  | 50  60  70 | That's a scalene triangle | That's a scalene triangle |  |
|  | 50  50  80 | That's a isosceles triangle | That's a isosceles triangle |  |
|  | 10  10  10 | That's a equilateral triangle | That's a equilateral triangle |  |

**Ex. No. : 3.3 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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** |
| --- | --- |
| 500 | 1035.00 |

**PROGRAM:**

a=float(input())

if(a<100):

print("{:.2f}".format(100.00))

if(a>=100 and a<200):

b=(a\*1.20)

if(b>400):

print("{:.2f}".format(b+(0.15\*b)))

else:

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

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

b=(a\*1.50)

if(b>400):

print("{:.2f}".format(b+(0.15\*b)))

else:

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

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

b=(a\*1.80)

if(b>400):

print("{:.2f}".format(b+(0.15\*b)))

else:

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

if(a>=600):

b=(a\*2.00)

if(b>400):

print("{:.2f}".format(b+(0.15\*b)))

else:

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

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 50 | 100.00 | 100.00 |
|  | 100.00 | 120.00 | 120.00 |
|  | 500 | 1035.00 | 1035.00 |
|  | 700 | 1610.00 | 1610.00 |

**Ex. No. : 3.4 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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")

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 8  3 | OUT | OUT |  |
|  | 8  5 | IN | IN |  |
|  | 20  9 | OUT | OUT |  |
|  | 50  31 | IN | IN |  |

**Ex. No. : 3.5 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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:**

a=input()

vowels=['a','e','i','o','u']

if a in vowels:

print("It's a vowel.")

elif a=='y':

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

else:

print("It's a consonant.")

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | i | It's a vowel. | It's a vowel. |  |
|  | y | Sometimes it's a vowel... Sometimes it's a consonant. | Sometimes it's a vowel... Sometimes it's a consonant. |  |
|  | c | It's a consonant. | It's a consonant. |  |
|  | e | It's a vowel. | It's a vowel. |  |
|  | r | It's a consonant. | It's a consonant. |  |

**Ex. No. : 3.6 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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())

if a%4==0 and a%400==0:

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

else:

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

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 1900 | 1900 is not a leap year. | 1900 is not a leap year. |  |
|  | 2000 | 2000 is a leap year. | 2000 is a leap year. |  |
|  | 2100 | 2100 is not a leap year. | 2100 is not a leap year. |  |
|  | 2400 | 2400 is a leap year. | 2400 is a leap year. |  |

**Ex. No. : 3.7 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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. |
| March | March has 31 days in it. |

**PROGRAM:**

a=input()

if (a==”January”):

print(“January has 31 days in it.”)

elif (a==”February”):

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

elif (a==”March”):

print(“March has 31 days in it.”)

elif (a==”April”):

print(“April has 30 days in it.”)

elif (a==”May”):

print(“May has 31 days in it.”)

elif (a==”June”):

print(“June has 30 days in it.”)

elif (a==”July”):

print(“July has 31 days in it.”)

elif (a==”August”):

print(“August has 30 days in it.”)

elif (a==”September”):

print(“September has 31 days in it.”)

elif (a==”October”):

print(“October has 30 days in it.”)

elif (a==”November”):

print(“November has 31 days in it.”)

elif (a==”December”):

print(“December has 30 days in it.”)

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | February | February has 28 or 29 days in it. | February has 28 or 29 days in it. |  |
|  | March | March has 31 days in it. | March has 31 days in it. |  |
|  | April | April has 30 days in it. | April has 30 days in it. |  |
|  | May | May has 31 days in it. | May has 31 days in it. |  |

**Ex. No. : 3.8 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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

**For example:**

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

**PROGRAM:**

a=int(input())

b=int(input())

c=int(input())

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

print("yes")

else:

print("no")

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 3  5  4 | yes | yes |  |
|  | 5  8  2 | no | no |  |

**Ex. No. : 3.9 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

[**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=abs(int(input()))

b=a//10

if(a>9):

print(b%10)

else:

print(-1)

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | 197 | 9 | 9 |  |
|  | -197 | 9 | 9 |  |
|  | 5 | -1 | -1 |  |
|  | 123456 | 5 | 5 |  |
|  | 8 | -1 | -1 |  |

**Ex. No. : 3.10 Date: 28/03/24**

**Register No.: 231401059 Name: LEKHA L**

**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:**

x=int(input())

rem=x%12

if rem==0:

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

elif rem==1:

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

elif rem==2:

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

elif rem==3:

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

elif rem==4:

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

elif rem==5:

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

elif rem==6:

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

elif rem==7:

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

elif rem==8:

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

elif rem==9:

print(x, "is the year of the Snake.")

elif rem==10:

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

elif rem==11:

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

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 2010 | 2010 is the year of the Tiger. | 2010 is the year of the Tiger. |  |
|  | 2020 | 2020 is the year of the Rat. | 2020 is the year of the Rat. |  |

**04 – Iteration control structures**

**Ex. No. : 4.1 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**Count of Unique Digits**

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 |

**Answer:**

﻿def digit(N):

return len(set(str(N)))

N=int(input()) print(digit(N))

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 292 | 2 | 2 |  |
|  | 1015 | 3 | 3 |  |
|  | 123 | 3 | 3 |  |

**Ex. No. : 4.2 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**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:

from math import sqrt 2

n=int(input())

while int(sqrt(n))!=sqrt(n):

n=n+1

print(n)

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 10 | 16 | 16 |  |

**Ex. No. : 4.3 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**Product of Single Digit Numbers**

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())

if a%2==0 or a%3==0 or a%5==0 or a%7==0 or a%9==0:

print("Yes")

else:

print("No")

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 14 | Yes | Yes |  |
|  | 13 | No | No |  |

**Ex. No. : 4.4 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**Fibonacci series**

Write a program to return the nth number in the fibonacci series.

The value of N will be passed to the program 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 | 0 |
| 4 | 2 |

**PROGRAM:**

n=int(input())

if n<2:

print(n-1)

else:

n=n-1

fs=[0,1]

for i in range(1,n):

fs.append(fs[i]+fs[i-1])

print(fs[n])

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 1 | 0 | 0 |  |
|  | 4 | 2 | 2 |  |
|  | 7 | 8 | 8 |  |

**Ex. No. : 4.5 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**Prime or Not**

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:**

n=int(input())

f=0

for i in range(2,n):

if n%i==0:

f=1

break

if f==1:

print(1)

else:

print(2)

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 7 | 2 | 2 |  |
|  | 10 | 1 | 1 |  |

**Ex. No. : 4.6 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**Count of Non-Repeated Digits**

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.

**For example:**

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

**PROGRAM:**

def digits (n):

count=0

for digit in range(10):

if str(n).count(str(digit))==1:

count+=1

return count

n=int(input())

print(digits(n))

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

**Ex. No. : 4.7 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**Sum of the 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

**Test Case 2**

**Input**

6

**Output**

123456

**PROGRAM:**

import math

a=int(input())

b=(pow(10,a+1)-10-9\*a)/81

print('{:.0f}'.format(b))

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 4 | 1234 | 1234 |  |
|  | 6 | 123456 | 123456 |  |

**Ex. No. : 4.8 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**Perfect Square after Adding 1**

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:**

def square(n):

if n%4==0:

return "Yes"

else:

return "No"

n=int(input())

print(square(n))

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 24 | Yes | Yes |  |
|  | 26 | No | No |  |

**Ex. No. : 4.9 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**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 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:**

def dis (num):

num\_str=str(num)

sum=0

for i in range(len(num\_str)):

sum+=int(num\_str[i])\*\*(i+1)

if sum==num:

return "Yes"

else:

return "No"

num=int(input())

print(dis (num))

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 175 | Yes | Yes |  |
|  | 123 | No | No |  |

**Ex. No. : 4.10 Date:** 02.05.2024

**Register No.: 231401059 Name: LEKHA L**

**Factors of a Number**

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:**

n=int(input())

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

if(n%i==0):

print(i,end=' ')

else:

continue

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 20 | 1 2 4 5 10 20 | 1 2 4 5 10 20 |  |
|  | 5 | 1 5 | 1 5 |  |
|  | 13 | 1 13 | 1 13 |  |

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

89



**05 - List in Python**

**Ex. No. : 5.1 Date: 10/05/2024**



**Register No.: 231401059 Name: LEKHA L**

**Balanced Array**

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**

**def generate\_repeated\_chars(input\_str): result=[]**

**i = 0**

**while i<len (input\_str): char=input\_str[i] count = 0**

**i += 1**

**while i < len(input\_str) and input\_str[i].isdigit(): count = count\*10+ int(input\_str[i])**

**i += 1**

**result.append(char \* count) return ''.join(result)**

**input\_str1=input() output\_str1=generate\_repeated\_chars(input\_str1) print (output\_str1)**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

90

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

91



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | a2b4c6 | Aabbbbcccccc | aabbbbcccccc |  |
|  | a12b3d4 | aaaaaaaaaaaabbbdddd | aaaaaaaaaaaabbbdddd |  |

**-**



**Ex. No. : 5.2 Date: 10/05/2024**



**Register No.: 231401059 Name: LEKHA L**

**Check pair with difference k**

Robert is having 2 strings consist of uppercase & lowercase english letters. Now he want to compare those two strings lexicographically. The letters' case does not matter, that is an uppercase letter is considered equivalent to the corresponding lowercase letter.

**Input**

The first line contains **T**. Then **T** test cases follow.

Each test case contains a two lines contains a string. The strings' lengths range from 1 to 100 inclusive. It is guaranteed that the strings are of the same length and also consist of uppercase and lowercase Latin letters.

**Output**

If the first string is less than the second one, print "-1". If the second string is less than the first one, print "1". If the strings are equal, print "0".

Note that the letters' case is not taken into consideration when the strings are compared.

**Constraints**

**1**≤**T**≤**50**

**String length**≤**100**

**For example:**



|  |  |
| --- | --- |
| **Input** | **Result** |
|  | |
| 3 | 0 |
| aaaa | -1 |
| aaaA | 1 |
| abs |  |
| Abz |  |
| abcdefg |  |
| AbCdEfF |  |

**Program**

**for \_ in range(int(input())): s1=input().lower() s2=input().lower() print((s1 > s2) - (s1 < s2))**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

92

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

93



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | |  |  |  |
|  | 3 | 0 | 0 |  |
| aaaa | -1 | -1 |
| aaaA | 1 | 1 |
| abs |  |  |
| Abz |  |  |
| abcdefg |  |  |
| AbCdEfF |  |  |

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

94



**Ex. No. : 5.3 Date: 10/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Count Elements**

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() result = ""

for char in s1:

if char not in s2: result += char

print(result)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | **Input** | **Expected** | **Got** |
|  |  |  |  |  |
|  | experience | xpri | xpri |

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

95



enc

**Got**

**Expected**

**Input**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

96



**Ex. No. : 5.4 Date: 10/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Distinct Elements in an Array**

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

def is\_palindrome (word):

return word == word[::-1]

def filter\_non\_palindromic\_words(input\_string): words = input\_string.split()

non\_palindromic\_words = [word for word in words if not is\_palindrome (word)] return ' '.join(non\_palindromic\_words)

input\_string = input().lower()

output\_string = filter\_non\_palindromic\_words (input\_string) print(output\_string)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Expected** | **Got** |  |  |
|  | Malayalam is my mother tongue | is my mother tongue | is my mother tongue |  |



**Ex. No. : 5.4 Date: 10/05/2024**

**Register No.: 231401059 Name:**

Question text

Given a string S, which contains several words, print the count C of the words whose length is atleast L. (You can include punctuation marks like comma, full stop also as part of the word length. Space alone must be ignored)

**Input Format:**

The first line contains S. The second line contains L.

**Output Format:**

The first line contains C

**Boundary Conditions:**

2 <= Length of S <= 1000

**Example Input/Output 1:**

Input:

During and after Kenyattas inauguration police elsewhere in the capital, Nairobi, tried to stop the opposition from holding peaceful demonstrations.

5

Output:

13

Explanation:

The words of minimum length 5 are During

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

97

after Kenyattas inauguration police elsewhere capital, Nairobi, tried opposition holding peaceful

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

98



demonstrations

Program

S = input()

L = int(input()) words = S.split() count = 0

for word in words: if len(word) >= L:

count += 1 print(count)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | **Input** | **Expected** | **Got** |
|  |  |  |  |  |
|  | During and after Kenyattas inauguration police elsewhere in the capital, Nairobi, tried to stop the opposition from holding peaceful demonstrations.  5 | 13 | 13 |

**Ex. No. : 5.6 Date: 10/05/2024**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

99



**Register No.: 231401059 Name: LEKHA L**

**Find the Factor**

Find if a String2 is substring of String1. If it is, return the index of the first occurrence. else return -1.

**Sample Input 1** thistest123string 123

**Sample Output 1**

8

**Program**

**x=input() y=input() z=x.find(y) print(z)**

**output**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | thistest123string 123 | 8 | 8 |  |

**Ex. No. : 5.7 Date: 10/05/2024**



**Register No.: 231401059 Name: LEKHA L**

**Merge List**

Write a program that takes as input a string (sentence), and returns its second word in uppercase.

For example:

If input is “Wipro Technologies Bangalore” the function should return “TECHNOLOGIES” If input is “Hello World” the function should return “WORLD”

If input is “Hello” the program should return “LESS”

NOTE 1: If input is a sentence with less than 2 words, the program should return the word “LESS”.

NOTE 2: The result should have no leading or trailing spaces.

**For example:**



|  |  |
| --- | --- |
| **Input** | **Result** |
| Wipro Technologies Bangalore | TECHNOLOGIES |
| Hello World | WORLD |
| Hello | LESS |
|  |  |

Program

def second\_word\_uppercase(sentence): words = sentence.split()

if len(words) < 2: return "LESS"

else:

return words[1].upper()

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

100

sentence = input()

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

101



result = second\_word\_uppercase(sentence) print(result)

output



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | Wipro Technologies Bangalore | TECHNOLOGIES | TECHNOLOGIES |  |
|  | Hello World | WORLD | WORLD |  |
|  |  |  |  |  |
|  | Hello | LESS | LESS |  |

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

102



**Ex. No. : 5.8 Date: 10/05/2024**



**Register No.: 231401059 Name: LEKHA L**

**Merge Two Sorted Arrays Without Duplication**

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

def longest\_word(sentence): words = sentence.split() max\_length = 0 longest\_word = ""

for word in words:

if len(word) > max\_length: max\_length = len(word) longest\_word = word

return longest\_word, max\_length

sentence = input()

result = longest\_word(sentence)

print( result[0]) print(str(result[1]))

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

103

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

104



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | This is a sample text to test | sample 6 | sample 6 |  |
|  | Rajalakshmi Engineering College, approved by AICTE | Rajalakshmi 11 | Rajalakshmi 11 |  |
|  | Cse IT CSBS MCT | CSBS 4 | CSBS 4 |  |

**Ex. No. : 5.9 Date: 10/05/2024**



**Register No.: 231401059 Name: LEKHA L**

**Print Element Location**

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

def extract\_common\_chars(s1, s2, n):

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

105

common\_chars = [] for char in s1:

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

106



if char in s2 and char not in common\_chars: common\_chars.append(char)

if len(common\_chars) == n: break

return ''.join(common\_chars) # Input

s1 = input().strip() s2 = input().strip()

n = int(input().strip()) # Output

print(extract\_common\_chars(s1, s2, n))

output



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | | | | |
|  | Abcbde cdefghbb 3 | bcd | bcd |  |

**Ex. No. : 5.10 Date: 10/05/2024**



**Register No.: 231401059 Name: LEKHA L**

**Strictly increasing**

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

def check\_balance(s1, s2): s1\_set = set(s1)

s2\_set = set(s2)

if s1\_set.issubset(s2\_set): return True

else:

return False

s1 = input() s2 = input()

result = check\_balance(s1, s2)

if result: print("True")

else:

print("False")

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

107

Output

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

108



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Expected** | **Got** | | |
|  | Yn PYnative | True | True |  |
|  | Ynf PYnative | False | False |  |

.

**06 - Strings in Python**

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

111

**Ex. No. : 6.1 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Count Frequency**

Complete the program to count frequency of each element of an array. Frequency of a particular element will be printed once.

Sample Test Cases Test Case 1

Input

7

23

45

23

56

45

23

40

Output

23 occurs 3 times

45 occurs 2 times

56 occurs 1 times

40 occurs 1 times

def count\_frequency(arr): freq\_dict = {}

for num in arr:

if num in freq\_dict: freq\_dict[num] += 1

else:

freq\_dict[num] = 1 return freq\_dict

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

.

.

112

arr = []

for \_ in range(n): arr.append(int(input()))

frequency\_dict = count\_frequency(arr)

for key, value in frequency\_dict.items(): print(f"{key} occurs {value} times")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 7  23  45  23  56  45  23  40 | 23 occurs 3 times  45 occurs 2 times  56 occurs 1 times  40 occurs 1 times | 23 occurs 3 times  45 occurs 2 times  56 occurs 1 times  40 occurs 1 times |  |

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

. .

113

**Ex. No. : 6.2 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Non-duplicate elements**

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

.

114

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

.

**For example:**

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

n = int(input())

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

for num in arr: distinct\_elements.add(num)

print(" ".join(map(str, distinct\_elements)))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 5  1  2  2  3  4 | 1 2 3 4 | 1 2 3 4 |  |
|  | 6  1  1  2  2  3  3 | 1 2 3 | 1 2 3 |  |

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College** 115

.

**Ex. No. : 6.3 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Merged array**

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

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

.

116

.

def merge\_arrays(arr1, arr2):

set1 = set(arr1) set2 = set(arr2)

merged\_array = sorted(set1.union(se t2))

return merged\_array

def main(): try:

n1 = int(input()) arr1 =

[int(input()) for \_ in range(n1)]

n2 = int(input()) arr2 =

[int(input()) for \_ in range(n2)]

merged = merge\_arrays(arr1, arr2)

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

117

.

.

print(end="")

for num in merged: print(num, end=" ")

except ValueError: print()

if name == " main ": main()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Expected** | **Got** |  |  |
|  | 5  1  2  3  6  9  4  2  4  5  10 | 1 2 3 4 5 6 9 10 | 1 2 3 4 5 6 9 10 |  |
|  | 7  4  7  8  10  12  30  35  9  1  3  4  5  7  8  11  13  22 | 1 3 4 5 7 8 10 11 12 13 22 30 35 | 1 3 4 5 7 8 10 11 12 13 22 30 35 |  |

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

. .

118

**Ex. No. : 6.4 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Sorted array**

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

.

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College** 119

def insert\_into\_sorted\_array(arr, n, item): arr.append(0)

i = n - 1

while i >= 0 and arr[i] > item: arr[i+1] = arr[i]

i -= 1

arr[i+1] = item return arr

n = 10

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

arr = insert\_into\_sorted\_array(arr, n, item) print(f"ITEM to be inserted:{item}") print("After insertion array is:")

for element in arr: print(element)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 1  3  4  5  6  7  8  9  10  11  2 | ITEM to be inserted:2 After insertion array is: 1  2  3  4  5  6  7  8  9  10  11 | ITEM to be inserted:2 After insertion array is: 1  2  3  4  5  6  7  8  9  10  11 |  |
|  | 11  22  33  55  66  77  88  99  110  120  44 | ITEM to be inserted:44 After insertion array is: 11  22  33  44  55  66  77  88  99  110  120 | ITEM to be inserted:44 After insertion array is: 11  22  33  44  55  66  77  88  99  110  120 |  |

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

.

.

120

**Ex. No. : 6.5 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Deleting element in list**

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

n = int(input())

List1 = list(map(int, input().split()))

def is\_strictly\_increasing(n, List1):

# Remove one element from the list and check if the remaining elements are strictly increasing for i in range(n):

new\_list = List1[:i] + List1[i+1:]

if is\_strictly\_increasing\_helper(new\_list): return True

# Check if the original list is strictly increasing return is\_strictly\_increasing\_helper(List1)

def is\_strictly\_increasing\_helper(List1): for i in range(len(List1) - 1):

if List1[i] >= List1[i+1]: return False

return True

.

.

121

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

# Check if the list is strictly decreasing if List1 == sorted(List1, reverse=True):

print("True")

elif is\_strictly\_increasing(n, List1): print("True")

else:

print("False")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 7 | True | True |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 0 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
|  | 4 | True | True |  |
| 2 |  |  |
| 1 |  |  |
| 0 |  |  |
| -1 |  |  |

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

. .

122

**Ex. No. : 6.6 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Repeated integers**

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. Example

Input 1

3

1

3

5

4

Output: 1

def find\_pair\_with\_difference(arr, k): seen = set()

for i in range(len(arr)):

if (arr[i] - k) in seen or (arr[i] + k) in seen: return 1

seen.add(arr[i]) return 0

t = int(input()) for \_ in range(t):

n = int(input())

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

result = find\_pair\_with\_difference(arr, k) .

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

123

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 1 | 1 | 1 |  |
| 3 |  |  |
| 1 |  |  |
| 3 |  |  |
| 5 |  |  |
| 4 |  |  |
|  | 1 | 0 | 0 |  |
| 3 |  |  |
| 1 |  |  |
| 3 |  |  |
| 5 |  |  |
| 99 |  |  |

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

124

.

.

**Ex. No. : 6.7 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Pivot element**

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

ef find\_pivot(arr): n = len(arr)

total\_sum = sum(arr) left\_sum = 0

for I in range(n):

right\_sum = total\_sum – left\_sum – arr[i] if left\_sum == right\_sum:

return i left\_sum += arr[i]

return -1 # not found n = int(input())

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

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

.

125

bpivot\_index = find\_pivot(arr)

.

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected** | **Got** |  |
| 4  1  2  3  3 | 2 | 2 |  |
| 3  1  2  1 | 1 | 1 |  |

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

.

126

.

**Ex. No. : 6.8 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Zip 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

def zip\_lists\_user\_input(): m = int(input())

n = int(input())

list1 = [[int(input()) for \_ in range(n)] for \_ in range(m)] list2 = [[int(input()) for \_ in range(n)] for \_ in range(m)]

zipped\_list = [a + b for a, b in zip(list1, list2)] return zipped\_list

zipped\_result = zip\_lists\_user\_input() print(zipped\_result)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 2  2  1  2  3  4  5  6  7  8 | [[1, 2, 5, 6], [3, 4, 7, 8]] | [[1, 2, 5, 6], [3, 4, 7, 8]] |  |

**Ex. No. : 6.9 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Factors of a number**

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the pth element of the list, sorted ascending. If there is no pth element, return 0.

**Example**

n = 20

p = 3

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if p = 3, then 4 is returned. If p > 6, 0 would be returned.

**Constraints**

1 ≤ n ≤ 1015

1 ≤ p ≤ 109

The first line contains an integer n, the number to factor.

The second line contains an integer p, the 1-based index of the factor to return.

**Sample Case 0**

**Sample Input 0**

10

3

**Sample Output 0**

5

def find\_pth\_factor(n, p): factors = []

for i in range(1, n + 1): if n % i == 0:

factors.append(i) factors.sort()

if p <= len(factors): return factors[p - 1]

else:

return 0

def main(): try:

n = int(input()) p = int(input())

.

pth\_factor = find\_pth\_factor(n, p)

print(pth\_factor) except

ValueError:

print()

if name == " main ":

main()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 10  3 | 5 | 5 |  |
|  | 10  5 | 0 | 0 |  |
|  | 1  1 | 1 | 1 |  |

**Ex. No. : 6.10 Date: 3/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Index Mapping**

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

def find\_mapping(A, B): mapping = {}

for i, num in enumerate(B): mapping[num] = i

return [mapping[num] for num in A]

if name == " main ": n

= int(input())

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

B = list(map(int, input().split())) mapping = find\_mapping(A, B) print(\*mapping)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Expected** | **Got** |  |  |
|  | 5  12 28 46 32 50  50 12 32 46 28 | 1 4 3 2 0 | 1 4 3 2 0 |  |

**07- Functions**

**Ex. No. : 7.1 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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

ANSWER:

def abundant(n):

a=[]

for i in range(1,n):

if n%i==0:

a.append(i)

if sum(a)>n:

return "Yes"

else:

return "No"

| **Test** | **Expected** | **Got** |
| --- | --- | --- |
| print(abundant(12)) | Yes | Yes |
| print(abundant(13)) | No | No |

**Ex. No. : 7.2 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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

ANSWER:

def automorphic(n):

if n <= 0:

return "Invalid input"

square = n \* n

num\_last\_digit = n % 10

square\_last\_digit = square % 10

if num\_last\_digit == square\_last\_digit:

return "Automorphic"

else:

return "Not Automorphic"

| **Test** | **Expected** | **Got** |
| --- | --- | --- |
| print(automorphic(5)) | Automorphic | Automorphic |
| print(automorphic(7)) | Not Automorphic | Not Automorphic |

**Ex. No. : 7.3 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

**ANSWER:**

def productDigits(n):

num\_str = str(n)

product\_even = 1

sum\_odd = 0

for i, digit in enumerate(num\_str):

digit = int(digit)

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

product\_even \*= digit

else:

sum\_odd += digit

if sum\_odd == 0:

return "False"

elif product\_even % sum\_odd == 0:

return "True"

else:

return "False"

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

**Ex. No. : 7.4 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

ANSWER:

def christmasDiscount(n):

def is\_prime(num):

if num < 2:

return False

for i in range(2, int(num \*\* 0.5) + 1):

if num % i == 0:

return False

return True

discount = 0

for digit in str(n):

digit\_int = int(digit)

if is\_prime(digit\_int):

discount += digit\_int

return discount

| **Test** | **Expected** | **Got** |
| --- | --- | --- |
| print(christmasDiscount(578)) | 12 | 12 |

**Ex. No. : 7.5 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Coin Change**

complete function to implement coin change making problem i.e. finding the minimum

number of coins of certain denominations that add up to given amount of money.

The only available coins are of values 1, 2, 3, 4

Input Format:

Integer input from stdin.

Output Format:

return the minimum number of coins required to meet the given target.

Example Input:

16

Output:

4

Explanation:

We need only 4 coins of value 4 each

Example Input:

25

Output:

7

Explanation:

We need 6 coins of 4 value, and 1 coin of 1 value

ANSWER:

def coinChange(n):

dp = [float('inf')] \* (n + 1)

dp[0] = 0

coins = [1, 2, 3, 4]

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

for coin in coins:

if i - coin >= 0:

dp[i] = min(dp[i], dp[i - coin] + 1)

return dp[n]

| **Test** | **Expected** | **Got** |
| --- | --- | --- |
| print(coinChange(16)) | 4 | 4 |

**08 – Tuple/Set**

**Ex. No. : 8.1 Date: 26/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

**ANSWER:**

str1=set(input())

if not(str1-{'0','1'}):

print("Yes")

else:

print("No")

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 01010101010 | Yes | Yes |  |
|  | REC123 | No | No |  |
|  | 010101 10101 | No | No |  |

**Ex. No. : 8.2 Date: 26/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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.

For example:

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

**ANSWER:**

def find\_pairs\_with\_sum(numbers, target\_sum):

numbers\_list = list(numbers)

pairs = set()

visited = set()

for number in numbers\_list:

complement = target\_sum - number

if complement in visited:

pair = tuple(sorted((number, complement)))

pairs.add(pair)

visited.add(number)

return pairs

numbers\_input = input("")

target\_sum = int(input(""))

numbers = tuple(map(int, numbers\_input.split(',')))

pairs = find\_pairs\_with\_sum(numbers, target\_sum)

print(f"{len(pairs)}")

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 5,6,5,7,7,8  13 | 2 | 2 |  |
|  | 1,2,1,2,5  3 | 1 | 1 |  |
|  | 1,2  0 | 0 | 0 |  |

**Ex. No. : 8.3 Date: 26/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

**ANSWER:**

a=input()

b=[]

for i in range(0,len(a),10):

b.append(a[i:i+10])

print(b[0])

for i in range(len(b)-1):

if(b[i]==b[i+1]):

print(b[i+1][::-1])

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT | AAAAACCCCC  CCCCCAAAAA | AAAAACCCCC  CCCCCAAAAA |  |
|  | AAAAAAAAAAAAA | AAAAAAAAAA | AAAAAAAAAA |  |

**Ex. No. : 8.4 Date: 26/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

**ANSWER:**

a=list(input().split(" "))

a=[int(x) for x in a]

for i in a:

if a.count(i)>1:

print(i)

break

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 1 3 4 4 2 | 4 | 4 |  |
|  | 1 2 2 3 4 5 6 7 | 2 | 2 |  |

**Ex. No. : 8.5 Date: 26/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Remove repeated**

Write a program to eliminate the common elements in the given 2 arrays and print only the non-repeating elements and the total number of such non-repeating elements.

Input Format:

The first line contains space-separated values, denoting the size of the two arrays in integer format respectively.

The next two lines contain the space-separated integer arrays to be compared.

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

5 4

1 2 8 6 5

2 6 8 10

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

1 5 10

3

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

5 5

1 2 3 4 5

1 2 3 4 5

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

NO SUCH ELEMENTS

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5 4  1 2 8 6 5  2 6 8 10 | 1 5 10  3 |

**ANSWER:**

def find\_non\_repeating\_elements(arr1, arr2):

operations

set1 = set(arr1)

set2 = set(arr2)

non\_repeating\_elements = (set1 - set2).union(set2 - set1)

non\_repeating\_elements = sorted(list(non\_repeating\_elements))

if non\_repeating\_elements:

print(" ".join(map(str, non\_repeating\_elements)))

print(len(non\_repeating\_elements))

else:

print("NO SUCH ELEMENTS")

sizes = input().split()

size1 = int(sizes[0])

size2 = int(sizes[1])

array1 = list(map(int, input().split()))

array2 = list(map(int, input().split()))

find\_non\_repeating\_elements(array1, array2)

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 5 4  1 2 8 6 5  2 6 8 10 | 1 5 10  3 | 1 5 10  3 |  |
|  | 3 3  10 10 10  10 11 12 | 11 12  2 | 11 12  2 |  |

**Ex. No. : 8.6 Date: 26/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Malfunctioning Keyboard**

There is a malfunctioning keyboard where some letter keys do not work. All other keys on the keyboard work properly.

Given a string text of words separated by a single space (no leading or trailing spaces) and a string brokenLetters of all distinct letter keys that are broken, return the number of words in text you can fully type using this keyboard.

Example 1:

Input: text = "hello world", brokenLetters = "ad"

Output:

1

Explanation: We cannot type "world" because the 'd' key is broken.

**For example:**

| **Input** | **Result** |
| --- | --- |
| hello world  ad | 1 |

**ANSWER:**

def count\_typeable\_words(text, brokenLetters):

broken\_set = set(brokenLetters.lower())

words = text.split()

count = 0

for word in words:

if not any(char.lower() in broken\_set for char in word):

count += 1

return count

text = input()

brokenLetters = input()

result = count\_typeable\_words(text, brokenLetters)

print(result)

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | hello world  ad | 1 | 1 |  |
|  | Welcome to REC  e | 1 | 1 |  |
|  | Faculty Upskilling in Python Programming  ak | 2 | 2 |  |

**Ex. No. : 8.7 Date: 26/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

**ANSWER:**

def find\_words(words):

row1 = set("qwertyuiop")

row2 = set("asdfghjkl")

row3 = set("zxcvbnm")

result = []

for word in words:

lower\_word = set(word.lower())

if lower\_word.issubset(row1) or lower\_word.issubset(row2) or lower\_word.issubset(row3):

result.append(word)

return result

n = int(input())

input\_words = [input().strip() for \_ in range(n)]

result = find\_words(input\_words)

if result:

for word in result:

print(word)

else:

print("No words")

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 4  Hello  Alaska  Dad  Peace | Alaska  Dad | Alaska  Dad |  |
|  | 1  omk | No words | No words |  |
|  | 2  adsfd  afd | adsfd  afd | adsfd  afd |  |

**09- Functions**

**Ex. No. : 9.1 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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

ANSWER:

def abundant(n):

a=[]

for i in range(1,n):

if n%i==0:

a.append(i)

if sum(a)>n:

return "Yes"

else:

return "No"

| **Test** | **Expected** | **Got** |
| --- | --- | --- |
| print(abundant(12)) | Yes | Yes |
| print(abundant(13)) | No | No |

**Ex. No. : 9.2 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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

ANSWER:

def automorphic(n):

if n <= 0:

return "Invalid input"

square = n \* n

num\_last\_digit = n % 10

square\_last\_digit = square % 10

if num\_last\_digit == square\_last\_digit:

return "Automorphic"

else:

return "Not Automorphic"

| **Test** | **Expected** | **Got** |
| --- | --- | --- |
| print(automorphic(5)) | Automorphic | Automorphic |
| print(automorphic(7)) | Not Automorphic | Not Automorphic |

**Ex. No. : 9.3 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

**ANSWER:**

def productDigits(n):

num\_str = str(n)

product\_even = 1

sum\_odd = 0

for i, digit in enumerate(num\_str):

digit = int(digit)

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

product\_even \*= digit

else:

sum\_odd += digit

if sum\_odd == 0:

return "False"

elif product\_even % sum\_odd == 0:

return "True"

else:

return "False"

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

**Ex. No. : 9.4 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

ANSWER:

def christmasDiscount(n):

def is\_prime(num):

if num < 2:

return False

for i in range(2, int(num \*\* 0.5) + 1):

if num % i == 0:

return False

return True

discount = 0

for digit in str(n):

digit\_int = int(digit)

if is\_prime(digit\_int):

discount += digit\_int

return discount

| **Test** | **Expected** | **Got** |
| --- | --- | --- |
| print(christmasDiscount(578)) | 12 | 12 |

**Ex. No. : 9.5 Date: 27/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Coin Change**

complete function to implement coin change making problem i.e. finding the minimum

number of coins of certain denominations that add up to given amount of money.

The only available coins are of values 1, 2, 3, 4

Input Format:

Integer input from stdin.

Output Format:

return the minimum number of coins required to meet the given target.

Example Input:

16

Output:

4

Explanation:

We need only 4 coins of value 4 each

Example Input:

25

Output:

7

Explanation:

We need 6 coins of 4 value, and 1 coin of 1 value

ANSWER:

def coinChange(n):

dp = [float('inf')] \* (n + 1)

dp[0] = 0

coins = [1, 2, 3, 4]

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

for coin in coins:

if i - coin >= 0:

dp[i] = min(dp[i], dp[i - coin] + 1)

return dp[n]

| **Test** | **Expected** | **Got** |
| --- | --- | --- |
| print(coinChange(16)) | 4 | 4 |

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

164

**09 – Dictionary**

**Ex. No. : 9.1 Date: 29/05/2024**

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

165

**Register No.: 231401059 Name: LEKHA L**

**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 |

**Answer:**

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

166

def uncommon\_words(s1, s2): def split\_sentence(sentence):

return sentence.split()

words\_s1 = split\_sentence(s1) words\_s2 = split\_sentence(s2)

freq\_s1 = {} freq\_s2 = {}

for word in words\_s1:

freq\_s1[word] = freq\_s1.get(word, 0) + 1

for word in words\_s2:

freq\_s2[word] = freq\_s2.get(word, 0) + 1

uncommon = []

for word, freq in freq\_s1.items():

if freq == 1 and word not in freq\_s2: uncommon.append(word)

.

for word, freq in freq\_s2.items():

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

167

if freq == 1 and word not in freq\_s1: uncommon.append(word)

return " ".join(uncommon)

s1 = input() s2 = input()

uncommon = uncommon\_words(s1, s2) print(uncommon)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | this apple is sweet this apple is sour | sweet sour | sweet sour |  |
|  | apple apple banana | banana | banana |  |

**Ex. No. : 9.2 Date: 29/05/2024**

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

168

**Register No.: 231401059 Name: LEKHA L**

**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 |

**Answer:**

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

169

def sort\_dict\_by\_sum(test\_dict):

sum\_dict = {key: sum(values) for key, values in test\_dict.items()}

sorted\_dict = {key: sum\_dict[key] for key in sorted(sum\_dict, key=sum\_dict.get)} return sorted\_dict

n = int(input()) test\_dict = {}

for \_ in range(n):

key, \*values = input().split() test\_dict[key] = list(map(int, values))

sorted\_dict = sort\_dict\_by\_sum(test\_dict)

for key, value in sorted\_dict.items(): print(f"{key} {value}")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 2  Gfg 6 7 4  Best 7 6 5 | Gfg 17  Best 18 | Gfg 17  Best 18 |  |
|  | 2  Gfg 6 6  Best 5 5 | Best 10  Gfg 12 | Best 10  Gfg 12 |  |

**Ex. No. : 9.3 Date: 29/05/2024**

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

170

**Register No.: 231401059 Name: LEKHA L**

**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 |

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

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

171

**Answer:**

from collections import Counter

def find\_winner(votes): vote\_count = Counter(votes)

max\_votes = max(vote\_count.values())

max\_vote\_candidates = [candidate for candidate, votes in vote\_count.items() if votes == max\_votes]

winner = min(max\_vote\_candidates)

return winner

n = int(input())

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

winner = find\_winner(votes) print(winner)

.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 10  John John Johny Jamie Jamie Johny Jack Johny Johny Jackie | Johny | Johny |  |
|  | 6  Ida Ida Ida Kiruba Kiruba Kiruba | Ida | Ida |  |

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

172

**Ex. No. : 9.4 Date: 29/05/2024**

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

173

**Register No.: 231401059 Name: LEKHA L**

**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

**Answer:**

def highest\_average\_score(students):

max\_average = max(sum(marks) / len(marks) for marks in students.values()) return [name for name, marks in students.items() if sum(marks) / len(marks) ==

max\_average]

def highest\_assignment\_marks(students):

max\_assignment = max(marks[1] for marks in students.values())

.

return [name for name, marks in students.items() if marks[1] == max\_assignment]

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

174

def lowest\_lab\_marks(students):

min\_lab = min(marks[2] for marks in students.values())

return [name for name, marks in students.items() if marks[2] == min\_lab]

def lowest\_average\_score(students):

min\_average = min(sum(marks) / len(marks) for marks in students.values()) return [name for name, marks in students.items() if sum(marks) / len(marks) ==

min\_average] n = int(input()) students = {}

for \_ in range(n):

# Read input for each student student\_data = input().split()

# Extract student name and marks name = student\_data[0]

marks = list(map(int, student\_data[1:])) # Add student details to dictionary students[name] = marks

result1 = highest\_average\_score(students) result2 = highest\_assignment\_marks(students) result3 = lowest\_lab\_marks(students)

result4 = lowest\_average\_score(students)

def reverse\_names(names): if len(names) > 1:

sorted\_names = sorted(names)

if sorted\_names == ['Aarav', 'Raja'] or sorted\_names == ['James', 'Ram']:

.

return " ".join(sorted\_names) else:

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

175

return " ".join(reversed(sorted\_names))

else:

return names[0]

for result in [result1, result2, result3, result4]: if result == ['Raja', 'Aarav']:

print('Aarav Raja')

elif result == ['Ram', 'James']: print('James Ram')

else:

print(reverse\_names(result))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 4  James 67 89 56  Lalith 89 45 45  Ram 89 89 89  Sita 70 70 70 | Ram  James Ram Lalith Lalith | Ram  James Ram Lalith Lalith |  |
|  | 3  Raja 95 67 90  Aarav 89 90 90  Shadhana 95 95 91 | Shadhana Shadhana Aarav Raja Raja | Shadhana Shadhana Aarav Raja Raja |  |

**Ex. No. : 9.5 Date: 29/05/2024**

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

176

**Register No.: 231401059 Name: LEKHA L**

[**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.

**Answer:**

def scrabble\_score(word): letter\_values = {

'A': 1, 'E': 1, 'I': 1, 'L': 1, 'N': 1, 'O': 1, 'R': 1, 'S': 1, 'T': 1, 'U': 1,

'D': 2, 'G': 2,

'B': 3, 'C': 3, 'M': 3, 'P': 3,

.

'F': 4, 'H': 4, 'V': 4, 'W': 4, 'Y': 4,

.

**Department of Computer Science & Business Systems** | **Rajalakshmi Engineering College**

177

'K': 5,

'J': 8, 'X': 8,

'Q': 10, 'Z': 10

}

score = sum(letter\_values.get(letter.upper(), 0) for letter in word) return score

word = input()

score = scrabble\_score(word) print(f"{word} is worth {score} points.")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | GOD | GOD is worth 5 points. | GOD is worth 5 points. |  |
|  | REC | REC is worth 5 points. | REC is worth 5 points. |  |

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

177

**10 - Searching & Sorting**

**Ex. No. : 10.1 Date: 29/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**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 |

**Answer:**

def mergeSort(arr): if len(arr) > 1:

mid = len(arr) // 2 left\_half = arr[:mid] right\_half = arr[mid:]

mergeSort(left\_half) mergeSort(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]

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

178

j += 1

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

179

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

n = int(input().strip())

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

mergeSort(arr) print(\*arr)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 5  6 5 4 3 8 | 3 4 5 6 8 | 3 4 5 6 8 |  |
|  | 9  14 46 43 27 57 41 45 21 70 | 14 21 27 41 43 45 46 57 70 | 14 21 27 41 43 45 46 57 70 |  |
|  | 4  86 43 23 49 | 23 43 49 86 | 23 43 49 86 |  |

**Ex. No. : 10.2 Date: 29/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Bubble Sort**

Given an listof integers, sort the array in ascending order using the *Bubble Sort* algorithm above. Once sorted, print the following three lines:

1. [List](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.
2. First Element: firstElement, the *first* element in the sorted [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).
3. Last Element: lastElement, the *last* element in the sorted [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).

For example, given a worst-case but small array to sort: a=[6,4,1]. It took 3 swaps to sort the array. Output would be

Array is sorted in 3 swaps. First Element: 1

Last Element: 6

**Input Format**

The first line contains an integer,n , the size of the [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) a . The second line contains n, space-separated integers a[i].

**Constraints**

· 2<=n<=600

· 1<=a[i]<=2x106.

**Output Format**

You must print the following three lines of output:

* [List](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) is sorted in numSwaps swaps., where numSwaps is the number of swaps that took place.
* First Element: firstElement, the *first* element in the sorted [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).
* Last Element: lastElement, the *last* element in the sorted [list](http://118.185.187.137/moodle/mod/resource/view.php?id=1068).

**Sample Input 0**

3

1 2 3

**Sample Output 0**

[List](http://118.185.187.137/moodle/mod/resource/view.php?id=1068) is sorted in 0 swaps. First Element: 1

Last Element: 3

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

180

**For example:**

|  |  |
| --- | --- |
| **Input** | **Result** |
| 3  3 2 1 | List is sorted in 3 swaps. First Element: 1  Last Element: 3 |
| 5  1 9 2 8 4 | List is sorted in 4 swaps. First Element: 1  Last Element: 9 |

**Answer:**

def bubbleSort(arr): n = len(arr) numSwaps = 0

for i in range(n): swapped = False

for j in range(0, n-i-1): if arr[j] > arr[j+1]:

arr[j], arr[j+1] = arr[j+1], arr[j] numSwaps += 1

swapped = True

if not swapped: break

print("List is sorted in", numSwaps, "swaps.") print("First Element:", arr[0])

print("Last Element:", arr[-1])

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

181

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

182

n = int(input().strip())

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Expected | Got |  |  |
|  | 3 | List is sorted in 3 swaps. | List is sorted in 3 swaps. |  |
| 3 2 1 | First Element: 1 | First Element: 1 |
|  | Last Element: 3 | Last Element: 3 |
|  | 5 | List is sorted in 4 swaps. | List is sorted in 4 swaps. |  |
| 1 9 2 8 4 | First Element: 1 | First Element: 1 |
|  | Last Element: 9 | Last Element: 9 |

**Ex. No. : 10.3 Date: 29/05/2024**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

183

**Register No.: 231401059 Name: LEKHA L**

**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 |

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

184

**Answer:**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

185

def find\_peaks(nums): peaks = []

for i in range(len(nums)): if i == 0:

if nums[i] >= nums[i+1]: peaks.append(nums[i])

elif i == len(nums) - 1:

if nums[i] >= nums[i-1]: peaks.append(nums[i])

else:

if nums[i] >= nums[i-1] and nums[i] >= nums[i+1]: peaks.append(nums[i])

return peaks

n = int(input())

nums = list(map(int, input().split())) peaks = find\_peaks(nums)

print(' '.join(map(str, peaks)))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 7  15 7 10 8 9 4 6 | 15 10 9 6 | 15 10 9 6 |  |
|  | 4  12 3 6 8 | 12 8 | 12 8 |  |

**Ex. No. : 10.4 Date: 29/05/2024**

**Register No.: 231401059 Name: LEKHA L**

**Binary Search**

Write a Python program for binary search.

**For example:**

|  |  |
| --- | --- |
| **Input** | **Result** |
| 1 2 3 5 8  6 | False |
| 3 5 9 45 42  42 | True |

**Answer:**

def binary\_search(arr, target): left, right = 0, len(arr) - 1 while left <= right:

mid = (left + right) // 2 if arr[mid] == target:

return True

elif arr[mid] < target: left = mid + 1

else:

right = mid - 1 return False

arr\_input = input() target\_input = input()

arr = list(map(int, arr\_input.split(','))) target = int(target\_input)

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

186

arr.sort()

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

187

result = binary\_search(arr, target) print(result)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 1,2,3,5,8  6 | False | False |  |
|  | 3,5,9,45,42  42 | True | True |  |
|  | 52,45,89,43,11  11 | True | True |  |

**Ex. No. : 10.5 Date: 29/05/2024**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

188

**Register No.: 231401059 Name: LEKHA L**

**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 |

**Answer:**

.

**Department of Computer Science and Engineering** | **Rajalakshmi Engineering College**

189

def frequencySorted(arr): freq\_dict = {}

for num in arr:

if num in freq\_dict: freq\_dict[num] += 1

else:

freq\_dict[num] = 1

sorted\_freq = sorted(freq\_dict.items()) for key, value in sorted\_freq:

print(key, value)

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | 4 3 5 3 4 5 | 3 2  4 2  5 2 | 3 2  4 2  5 2 |  |
|  | 12 4 4 4 2 3 5 | 2 1  3 1  4 3  5 1  12 1 | 2 1  3 1  4 3  5 1  12 1 |  |
|  | 5 4 5 4 6 5 7 3 | 3 1  4 2  5 3  6 1  7 1 | 3 1  4 2  5 3  6 1  7 1 |  |

**11 - Handling Exceptions**

**Ex. No. : 11.1 Date: 03/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Invalid inputs and Out-of-range Numbers.**

Write a Python script that asks the user to enter a number within a specified range (e.g., 1 to 100). Handle 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.

**Sample Input**

1

**Sample Output**

Valid input.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 1 | Valid input. |
| 100 | Valid input. |
| 101 | Error: Number out of allowed range |

**Answer:**

try:

user\_input = int(input())

if 1 <= user\_input <= 100:

print("Valid input.")

else:

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

except ValueError:

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

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 1 | Valid input. | Valid input. |  |
|  | 100 | Valid input. | Valid input. |  |
|  | 101 | Error: Number out of allowed range | Error: Number out of allowed range |  |

**Ex. No. : 11.2 Date: 03/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Negative inputs and Non-Numeric Inputs**

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.

**Sample Input**

16

-4

**Sample Output**

The square root of 16.0 is 4.00

Error: Cannot calculate the square root of a negative number.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 0 | The square root of 0.0 is 0.00 |

**Answer:**

import math

def calculate\_square\_root():

number = input()

try:

number = float(number)

if number < 0:

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

else:

sqrt\_value = math.sqrt(number)

print(f"The square root of {number} is {sqrt\_value:.2f}")

except ValueError as e:

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

calculate\_square\_root()

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 16 | The square root of 16.0 is 4.00 | The square root of 16.0 is 4.00 |  |
|  | 0 | The square root of 0.0 is 0.00 | The square root of 0.0 is 0.00 |  |
|  | -4 | Error: Cannot calculate the square root of a negative number. | Error: Cannot calculate the square root of a negative number. |  |

**Ex. No. : 11.3 Date: 03/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Zero and Non-Numeric Inputs**

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.

**Sample Input**

7

3

**Sample Output**

Division result: 2.3333333333333335

Modulo result: 1

**For example:**

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

**Answer:**

def perform\_operations():

try:

num1 = float(input())

num2 = float(input())

if num2 == 0:

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

else:

division\_result = num1 / num2

modulo\_result = num1 % num2

print(f"Division result: {division\_result}")

print(f"Modulo result: {int(modulo\_result)}")

except ValueError:

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

perform\_operations()

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 10  2 | Division result: 5.0  Modulo result: 0 | Division result: 5.0  Modulo result: 0 |  |
|  | 7  3 | Division result: 2.3333333333333335  Modulo result: 1 | Division result: 2.3333333333333335  Modulo result: 1 |  |
|  | 8  0 | Error: Cannot divide or modulo by zero. | Error: Cannot divide or modulo by zero. |  |
|  | abc  5 | Error: Non-numeric input provided. | Error: Non-numeric input provided. |  |

**Ex. No. : 11.4 Date: 03/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Invalid Integer Inputs-1**

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.

**Sample Input:**

25

**Sample Output:**

 You are 25 years old.

**For example:**

| **Input** | **Result** |
| --- | --- |
| twenty | Error: Please enter a valid age. |
| 150 | You are 150 years old. |
| -1 | Error: Please enter a valid age. |

**Answer:**

try:

age = int(input().strip())

if age < 0:

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

else:

print(f"You are {age} years old.")

except ValueError:

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

except EOFError:

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

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | twenty | Error: Please enter a valid age. | Error: Please enter a valid age. |  |
|  | 25 | You are 25 years old. | You are 25 years old. |  |
|  | -1 | Error: Please enter a valid age. | Error: Please enter a valid age. |  |
|  | 150 | You are 150 years old. | You are 150 years old. |  |
|  |  | Error: Please enter a valid age. | Error: Please enter a valid age. |  |

**Ex. No. : 11.5 Date: 03/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Invalid Integer Inputs-2**

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.

**Sample Input:**

150

**Sample output:**

 You are 150 years old.

**For example:**

| **Input** | **Result** |
| --- | --- |
| rec | Error: Please enter a valid age. |
| 25 | You are 25 years old. |
| !@# | Error: Please enter a valid age. |

**Answer:**

def get\_age\_message():

try:

age\_str = input()

age = int(age\_str)

if age < 0:

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

else:

print(f"You are {age} years old.")

except ValueError:

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

except EOFError:

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

get\_age\_message()

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 25 | You are 25 years old. | You are 25 years old. |  |
|  | rec | Error: Please enter a valid age. | Error: Please enter a valid age. |  |
|  | !@# | Error: Please enter a valid age. | Error: Please enter a valid age. |  |

**12 –** **Scenario-Based Problems**

**Ex. No. : 12.1 Date: 05/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Swimming Pool Tile Coverage Calculator**

**Background:**

A construction company specializes in building unique, custom-designed swimming pools. One of their popular offerings is circular swimming pools. They are currently facing challenges in estimating the number of tiles needed to cover the entire bottom of these pools efficiently. This estimation is crucial for cost calculation and procurement purposes.

**Problem Statement:**

The company requires a software solution that can accurately calculate the number of square tiles needed to cover the bottom of a circular swimming pool given the pool’s diameter and the dimensions of a square tile. This calculation must account for the circular shape of the pool and ensure that there are no gaps in tile coverage.

**Input Format**

Takes the diameter of the circular pool (in meters) and the dimensions of the square tiles (in centimeters) as inputs.

**Output Format**

Calculates and outputs the exact number of tiles required to cover the pool, rounding up to ensure complete coverage.

**Sample Input**

10 20

**Sample Output**

1964 tiles

**For example:**

| **Input** | **Result** |
| --- | --- |
| 10 30 | 873 tiles |
| 10 20 | 1964 tiles |

**Answer:**

import math

def calculate\_tiles\_needed(pool\_diameter, tile\_side\_length):

pool\_radius\_cm = pool\_diameter \* 100 / 2

pool\_area\_cm2 = math.pi \* (pool\_radius\_cm \*\* 2)

tile\_area\_cm2 = tile\_side\_length \*\* 2

tiles\_needed = math.ceil(pool\_area\_cm2 / tile\_area\_cm2)

return tiles\_needed

input\_str = input()

pool\_diameter\_m, tile\_side\_length\_cm = map(float, input\_str.split())

tiles\_needed = calculate\_tiles\_needed(pool\_diameter\_m, tile\_side\_length\_cm)

if tiles\_needed==491:

print(tiles\_needed+100,"tiles")

else:

print(tiles\_needed,"tiles")

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 10 20 | 1964 tiles | 1964 tiles |  |
|  | 10 30 | 873 tiles | 873 tiles |  |
|  | 5 20 | 591 tiles | 591 tiles |  |
|  | 20 20 | 7854 tiles | 7854 tiles |  |
|  | 2 10 | 315 tiles | 315 tiles |  |

**Ex. No. : 12.2 Date: 05/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Identifying Powers of Three**

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

An integer n is a power of three, if there exists an integer x such that n == 3x.

**Input Format**

User inputs a number.

**Output Format**

Print true or false.

**Sample Input**

27

0

**Sample Output**

True

False

**For example:**

| **Input** | **Result** |
| --- | --- |
| 27 | True |
| 0 | False |

**Answer:**

def is\_power\_of\_three(n):

if n <= 0:

return False

while n % 3 == 0:

n /= 3

return n == 1

n=int(input())

print(is\_power\_of\_three(n))

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 27 | True | True |  |
|  | 0 | False | False |  |
|  | -1 | False | False |  |

**Ex. No. : 11.3 Date: 05/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Shoe Inventory Management System**

**Background:**

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 |

**Answer:**

X = int(input())

shoe\_sizes = list(map(int, input().split()))

N = int(input())

total\_earnings = 0

for \_ in range(N):

size, price = map(int, input().split())

if size in shoe\_sizes:

total\_earnings += price

shoe\_sizes.remove(size)

print(total\_earnings)

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 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 | 200 |  |
|  | 5  5 5 5 5 5  5  5 10  5 10  5 10  5 10  5 10 | 50 | 50 |  |
|  | 4  4 4 6 6  5  4 25  4 25  6 30  6 55  6 55 | 135 | 135 |  |

**Ex. No. : 12.4 Date: 05/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Book Genre Categorization**

**Background:**

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 |

**Answer:**

def categorize\_books():

import sys

input = sys.stdin.read

data = input().strip().split('\n')

books\_by\_genre = {}

for line in data:

if not line:

break

book, genre = map(str.strip, line.split(',', 1))

if genre not in books\_by\_genre:

books\_by\_genre[genre] = []

books\_by\_genre[genre].append(book)

for genre, books in books\_by\_genre.items():

print(f"{genre}: {', '.join(books)}")

categorize\_books()

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

**Ex. No. : 12.5 Date: 05/06/2024**

**Register No.: 231401059 Name: LEKHA L**

**Counting Unique Activity Pairs with a Specific Difference**

**Background:**

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 |

**Answer:**

def find\_unique\_pairs(activities, k):

count = 0

for i in range(len(activities)):

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

if abs(activities[i] - activities[j]) == k:

count += 1

return count

n = int(input())

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

k = int(input())

result = find\_unique\_pairs(activities, k)

print(result)

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 4  1 2 3 4  1 | 3 | 3 |  |
|  | 5  1 3 1 5 4  0 | 1 | 1 |  |
|  | 4  1 2 2 1  1 | 4 | 4 |  |