

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105



RAJALAKSHMI
ENGINEERING COLLEGE

CD23321

PYTHON PRROGRAMMING FOR DESIGN

Laboratory Manual Note Book

Name :

Year / Branch / Section :

Register No. :

Semester :

Academic Year :



RAJALAKSHMI ENGINEERING COLLEGE

An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

BONAFIDE CERTIFICATE

NAME

ACADEMIC YEAR SEMESTER BRANCH.....

UNIVERSITY REGISTER No.

Certified that this is the bonafide record of work done by the above student in the

..... Laboratory during the year 20 - 20

Signature of Faculty - in - Charge

Submitted for the Practical Examination held on

Internal Examiner

External Examiner

INDEX

Reg. No.:

Name:

Year:

Department:

| S.No. | Date | Title | Page No. | Teacher's Signature / Remarks |
|--|------|--------------|----------|-------------------------------|
| VARIABLES, DATATYPES IN PYTHON | | | | |
| 1.1 | | PROBLEM - 1 | | |
| 1.2 | | PROBLEM - 2 | | |
| 1.3 | | PROBLEM - 3 | | |
| 1.4 | | PROBLEM - 4 | | |
| 1.5 | | PROBLEM - 5 | | |
| 1.6 | | PROBLEM - 6 | | |
| 1.7 | | PROBLEM - 7 | | |
| 1.8 | | PROBLEM - 8 | | |
| 1.9 | | PROBLEM - 9 | | |
| 1.10 | | PROBLEM - 10 | | |
| OPERATORS AND FORMATTING OUTPUT | | | | |
| 2.1 | | PROBLEM - 1 | | |
| 2.2 | | PROBLEM - 2 | | |
| 2.3 | | PROBLEM - 3 | | |
| 2.4 | | PROBLEM - 4 | | |
| 2.5 | | PROBLEM - 5 | | |
| 2.6 | | PROBLEM - 6 | | |
| 2.7 | | PROBLEM - 7 | | |
| 2.8 | | PROBLEM - 8 | | |
| 2.9 | | PROBLEM - 9 | | |
| 2.10 | | PROBLEM - 10 | | |

| SELECTION CONTROLS | | | | |
|--------------------|--|--------------|--|--|
| 3.1 | | PROBLEM - 1 | | |
| 3.2 | | PROBLEM - 2 | | |
| 3.3 | | PROBLEM - 3 | | |
| 3.4 | | PROBLEM - 4 | | |
| 3.5 | | PROBLEM - 5 | | |
| 3.6 | | PROBLEM - 6 | | |
| 3.7 | | PROBLEM - 7 | | |
| 3.8 | | PROBLEM - 8 | | |
| 3.9 | | PROBLEM - 9 | | |
| 3.10 | | PROBLEM - 10 | | |
| ITERATION CONTROLS | | | | |
| 4.1 | | PROBLEM - 1 | | |
| 4.2 | | PROBLEM - 2 | | |
| 4.3 | | PROBLEM - 3 | | |
| 4.4 | | PROBLEM - 4 | | |
| 4.5 | | PROBLEM - 5 | | |
| 4.6 | | PROBLEM - 6 | | |
| 4.7 | | PROBLEM - 7 | | |
| 4.8 | | PROBLEM - 8 | | |
| 4.9 | | PROBLEM - 9 | | |
| 4.10 | | PROBLEM - 10 | | |
| FUNCTIONS | | | | |
| 5.1 | | PROBLEM - 1 | | |
| 5.2 | | PROBLEM - 2 | | |
| 5.3 | | PROBLEM - 3 | | |
| 5.4 | | PROBLEM - 4 | | |
| 5.5 | | PROBLEM - 5 | | |
| 5.6 | | PROBLEM - 6 | | |
| 5.7 | | PROBLEM - 7 | | |
| 5.8 | | PROBLEM - 8 | | |
| 5.9 | | PROBLEM - 9 | | |
| 5.10 | | PROBLEM - 10 | | |

| STRINGS | | | | |
|-----------------|--|--------------|--|--|
| 6.1 | | PROBLEM - 1 | | |
| 6.2 | | PROBLEM - 2 | | |
| 6.3 | | PROBLEM - 3 | | |
| 6.4 | | PROBLEM - 4 | | |
| 6.5 | | PROBLEM - 5 | | |
| 6.6 | | PROBLEM - 6 | | |
| 6.7 | | PROBLEM - 7 | | |
| 6.8 | | PROBLEM - 8 | | |
| 6.9 | | PROBLEM - 9 | | |
| 6.10 | | PROBLEM - 10 | | |
| LISTS | | | | |
| 7.1 | | PROBLEM - 1 | | |
| 7.2 | | PROBLEM - 2 | | |
| 7.3 | | PROBLEM - 3 | | |
| 7.4 | | PROBLEM - 4 | | |
| 7.5 | | PROBLEM - 5 | | |
| 7.6 | | PROBLEM - 6 | | |
| 7.7 | | PROBLEM - 7 | | |
| 7.8 | | PROBLEM - 8 | | |
| 7.9 | | PROBLEM - 9 | | |
| 7.10 | | PROBLEM - 10 | | |
| TUPLES AND SETS | | | | |
| 8.1 | | PROBLEM - 1 | | |
| 8.2 | | PROBLEM - 2 | | |
| 8.3 | | PROBLEM - 3 | | |
| 8.4 | | PROBLEM - 4 | | |
| 8.5 | | PROBLEM - 5 | | |
| 8.6 | | PROBLEM - 6 | | |
| 8.7 | | PROBLEM - 7 | | |
| 8.8 | | PROBLEM - 8 | | |
| 8.9 | | PROBLEM - 9 | | |
| 8.10 | | PROBLEM - 10 | | |

| DICTIONARY | | | | |
|------------|--|--------------|--|--|
| 9.1 | | PROBLEM - 1 | | |
| 9.2 | | PROBLEM - 2 | | |
| 9.3 | | PROBLEM - 3 | | |
| 9.4 | | PROBLEM - 4 | | |
| 9.5 | | PROBLEM - 5 | | |
| 9.6 | | PROBLEM - 6 | | |
| 9.7 | | PROBLEM - 7 | | |
| 9.8 | | PROBLEM - 8 | | |
| 9.9 | | PROBLEM - 9 | | |
| 9.10 | | PROBLEM - 10 | | |
| FILES | | | | |
| 10.1 | | PROBLEM - 1 | | |
| 10.2 | | PROBLEM - 2 | | |
| 10.3 | | PROBLEM - 3 | | |
| 10.4 | | PROBLEM - 4 | | |
| 10.5 | | PROBLEM - 5 | | |
| 10.6 | | PROBLEM - 6 | | |
| 10.7 | | PROBLEM - 7 | | |
| 10.8 | | PROBLEM - 8 | | |
| 10.9 | | PROBLEM - 9 | | |
| 10.10 | | PROBLEM - 10 | | |

State Finished
Completed on Thursday, 8 August 2024, 2:21 PM
Time taken 2 days 3 hours
Marks 10.00/10.00
Grade **100.00** out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

Write a program that returns the second last digit of the given number. Second last digit is being referred to the 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 |
| -197 | 9 |
| 5 | -1 |

Answer: (penalty regime: 0 %)

```
1 x = input()
2 if(len(x)<=1):
3     print("-1")
4 else:
5     print(x[-2])
6
7
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 197 | 9 | 9 | ✓ |
| ✓ | -197 | 9 | 9 | ✓ |
| ✓ | 5 | -1 | -1 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The Following python program has been executed successfully

Question 2

Correct

Mark 1.00 out of 1.00

You went on a tour to Ooty with your friends. As a part of the tour, you went boating with them. For the boat to remain stable, the number of people on one boat is restricted based on the weight of the people. You find that the boatman who is sailing your boat is so much greedy of money. For earning more, he takes too many people to travel in the boat at a time. So you want to check how many people can travel in the boat at a time so that the boat will not drown. Calculate the weight by considering the number of adults and number of children. Assume that an adult weighs 75 kg and children weigh 30 kg each. If the weight is normal, display Boat is stable, else display Boat will drown.

INPUT & OUTPUT FORMAT:

Input consists of 3 integers.

First input corresponds to the weight that the boat can handle.

Second input corresponds to the number of adults.

Third input corresponds to the number of children.

Answer: (penalty regime: 0 %)

```

1 boat = int(input())
2 adult = int(input())
3 child = int(input())
4 if((adult*75 + child*30)<boat):
5     print("Boat is stable")
6 else:
7     print("Boat will drown")

```

| | Input | Expected | Got | |
|---|---------------|-----------------|-----------------|---|
| ✓ | 340 2 3 | Boat is stable | Boat is stable | ✓ |
| ✓ | 600 7 4 | Boat will drown | Boat will drown | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been successfully executed

Question **3**

Correct

Mark 1.00 out of 1.00

In department 54% are boys and 46% are girls and 8% are hostel (boys/girls). write a python code to print total no of boys, girls and hostel students in the specific format using modulo operator.

input: 1500

output: Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120

Answer: (penalty regime: 0 %)

```
1 tot = int(input())
2 boys = int((54/100)*tot)
3 girls = int((46/100)*tot)
4 hostel = int((8/100)*tot)
5 # print("Total Students :",tot,", Boys :",boys,", Girls :",girls,", Hostel :",hostel )
6 print(f"Total Students : {tot}, Boys : {boys}, Girls : {girls}, Hostel : {hostel}")
```

| | Input | Expected | Got | |
|---|-------|--|--|---|
| ✓ | 1500 | Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120 | Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 4

Correct

Mark 1.00 out of 1.00

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: (penalty regime: 0 %)

```

1 buy = int(input())
2 repair = int(input())
3 sell = int(input())
4 cost = buy + repair
5 gain_per = ((sell - cost)/cost)*100
6 print("%.2f is the gain percent."%gain_per)

```

| | Input | Expected | Got | |
|---|-----------------------|----------------------------|----------------------------|---|
| ✓ | 10000 250 15000 | 46.34 is the gain percent. | 46.34 is the gain percent. | ✓ |

| | Input | Expected | Got | |
|---|------------------------|----------------------------|----------------------------|---|
| ✓ | 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. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 5

Correct

Mark 1.00 out of 1.00

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: (penalty regime: 0 %)

```
1 small = int(input())
2 big = int(input())
3 small_am = small*.10
4 big_am = big*.25
5 tot = small_am + big_am
6 print("Your total refund will be $%.2f."%tot)
```

| | 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. | ✓ |

| | Input | Expected | Got | |
|---|----------|------------------------------------|------------------------------------|---|
| ✓ | 76 38 | Your total refund will be \$17.10. | Your total refund will be \$17.10. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **6**

Correct

Mark 1.00 out of 1.00

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,<class 'int'> |
| 10.9 | 10.9,<class 'float'> |

Answer: (penalty regime: 0 %)

```
1 x=int(input())
2 y=float(input())
3 print(f"{x},{type(x)}")
4 print(f"{round(y,1)},{type(y)}")
```

| | 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'> | ✓ |
| ✓ | 55000 56.2 | 55000,<class 'int'> 56.2,<class 'float'> | 55000,<class 'int'> 56.2,<class 'float'> | ✓ |

| | Input | Expected | Got | |
|---|------------------|--|--|---|
| ✓ | 2541 2541.679 | 2541,<class 'int'> 2541.7,<class 'float'> | 2541,<class 'int'> 2541.7,<class 'float'> | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **7**

Correct

Mark 1.00 out of 1.00

In a Logistic the Parcels to be delivered in 4 locations (1st locaion 20%, 2nd location 40%, 3rd location 30% and 4th location 10%). write a python code to find the total no. of parcels after the delivery in 2 locations . use a format() to print the no of parcels delivered in in each location

Input:

250

output:

Total Parcels is 250

1st Location 50 parcels

2nd Location 100 parcels

3rd Location 75 parcels

4th Location 25 parcels

Answer: (penalty regime: 0 %)

```
1 tot = int(input())
2 first = int((20/100)*tot)
3 sec = int((40/100)*tot)
4 thr = int((30/100)*tot)
5 four = int((10/100)*tot)
6 print(f"Total Parcels is {tot}\n1st Location {first} parcels\n2nd Location {sec} parcels\n3rd Location {
```

| | Input | Expected | Got | |
|---|-------|---|---|---|
| ✓ | 250 | Total Parcels is 250 1st Location 50 parcels 2nd Location 100 parcels 3rd Location 75 parcels 4th Location 25 parcels | Total Parcels is 250 1st Location 50 parcels 2nd Location 100 parcels 3rd Location 75 parcels 4th Location 25 parcels | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **8**

Correct

Mark 1.00 out of 1.00

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: (penalty regime: 0 %)

```
1 basic = int(input())
2 allowance = int((60/100)*basic)
3 print(basic+allowance)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **9**

Correct

Mark 1.00 out of 1.00

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: (penalty regime: 0 %)

```
1 sal = int(input())
2 y= (sal-500)/130
3 y=abs(y)
4 print("weekdays %.2f"%(y+10))
5 print("weekend %.2f"%y)
```

| | 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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **10**

Correct

Mark 1.00 out of 1.00

In a Lab 36% are Dell and 34% Lenovo and 28% are Acer and 2% are Samsung. write a python code to print total systems and brand wise count in the specific format using sep operator.

input: 150

output: Total System:150

Dell:54

Lenovo:51

Acer:42

Samsung:3

Answer: (penalty regime: 0 %)

```
1 tot = int(input())
2 dell = int((36/100)*tot)
3 lenovo = int((34/100)*tot)
4 acer = int((28/100)*tot)
5 samsung = int((2/100)*tot)
6 print(f"Total System:{tot}\nDell:{dell}\nLenovo:{lenovo}\nAcer:{acer}\nSamsung:{samsung}")
```

| | Input | Expected | Got | |
|---|-------|--|--|---|
| ✓ | 150 | Total System:150 Dell:54 Lenovo:51 Acer:42 Samsung:3 | Total System:150 Dell:54 Lenovo:51 Acer:42 Samsung:3 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

| | |
|---------------------|--------------------------------|
| Started on | Friday, 9 August 2024, 1:53 PM |
| State | Finished |
| Completed on | Friday, 9 August 2024, 2:39 PM |
| Time taken | 45 mins 37 secs |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

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

Input format:

Line 1 has the total number of weapons

Line 2 has the total number of Soldiers.

Output Format:

If the battle can be won print True otherwise print False.

Sample Input:

32

43

Sample Output:'

False

For example:

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

Answer: (penalty regime: 0 %)

```
1 weapon = int(input())
2 soldier = int(input())
3 ▼ if((weapon%3) == 0 and (soldier%2)==0):
4     print("True")
5 ▼ else:
6     print("False")
```

| | Input | Expected | Got | |
|---|---------------|----------|-------|---|
| ✓ | 32 43 | False | False | ✓ |
| ✓ | 273 7890 | True | True | ✓ |
| ✓ | 800 4590 | False | False | ✓ |
| ✓ | 6789 32996 | True | True | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

Note:

Dont use if-else. Operators alone must be used .

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

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

Input Format:

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

Output Format:

Display True(IF ELIGIBLE)

Display False (if not eligible)

Sample Input

19

45

Sample Output

True

For example:

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

Answer: (penalty regime: 0 %)

```
1 age = int(input())
2 weight = int(input())
3 if(age>=18 and weight>40):
4     print("True")
5 else:
6     print("False")
```

| | Input | Expected | Got | |
|---|----------|----------|-------|---|
| ✓ | 19 45 | True | True | ✓ |
| ✓ | 18 40 | False | False | ✓ |
| ✓ | 18 42 | True | True | ✓ |
| ✓ | 16 45 | False | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



Result: The following program has been executed successfully

Question **3**

Correct

Mark 1.00 out of 1.00

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

The last digit should be returned as a positive number.

For example,

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

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

For example:

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

Answer: (penalty regime: 0 %)

```
1 a = input()
2 print(a[-1])
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 197 | 7 | 7 | ✓ |
| ✓ | -197 | 7 | 7 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 4

Correct

Mark 1.00 out of 1.00

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

[Sample](#) Input:

10

20

[Sample](#) Output:

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

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2 b = int(input())
3 print(f"The total weight of all these widgets and gizmos is {(a*75)+(b*112)} grams.")
```

| | Input | Expected | Got | |
|---|----------|---|---|---|
| ✓ | 10 20 | The total weight of all these widgets and gizmos is 2990 grams. | The total weight of all these widgets and gizmos is 2990 grams. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **5**

Correct

Mark 1.00 out of 1.00

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, $0 \leq x \leq 1$.**Output Format:**

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

Input 1:

0

Output 1:

C

Input 2:

1

Output 1:

D

For example:

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

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2 if(a == 0):
3     print("C")
4 else:
5     print("D")
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 0 | C | C | ✓ |
| ✓ | 1 | D | D | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully



Question 6

Correct

Mark 1.00 out of 1.00

Write a program to find whether the given input number is Even.

If the given number is even, the function should return 2 else it should return 1.

Note: The number passed to the program can either be negative, positive or zero. Zero should be treated as Even.

For example:

| Input | Result |
|-------|--------|
| 100 | 2 |
| 1001 | 1 |

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2 if(a%2 == 0):
3     print("2")
4 else:
5     print("1")
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 100 | 2 | 2 | ✓ |
| ✓ | 1001 | 1 | 1 | ✓ |
| ✓ | 0 | 2 | 2 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 7

Correct

Mark 1.00 out of 1.00

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

The last digit should be returned as a positive number.

For example,

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

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

For example:

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

Answer: (penalty regime: 0 %)

```
1 a = input()
2 print(a[-1])
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 197 | 7 | 7 | ✓ |
| ✓ | -197 | 7 | 7 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **8**

Correct

Mark 1.00 out of 1.00

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

Sample Input

3

Sample Output:

2

Explanation:

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

For example:

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

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2
3 bin_a = bin(a)
4 print(bin_a.count('1'))
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 3 | 2 | 2 | ✓ |
| ✓ | 5 | 2 | 2 | ✓ |
| ✓ | 15 | 4 | 4 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **9**

Correct

Mark 1.00 out of 1.00

Complete the program to convert days into years, month and days. (Ignoring leap year and considering 1 month is 30 days)

Sample Test Cases

Test Case 1

Input

375

Output

YEARS: 1 MONTH: 0 DAYS: 10

Test Case 2

Input

200

Output

YEARS: 0 MONTH: 6 DAYS: 20

Answer: (penalty regime: 0 %)

```
1 days = int(input())
2 years = days//365
3 rem_days = days-(years*365)
4 month = rem_days // 30
5 days = rem_days -(month*30)
6 print(f"YEARS: {years} MONTH: {month} DAYS: {days}")
```

| | Input | Expected | Got | |
|---|-------|----------------------------|----------------------------|---|
| ✓ | 375 | YEARS: 1 MONTH: 0 DAYS: 10 | YEARS: 1 MONTH: 0 DAYS: 10 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **10**

Correct

Mark 1.00 out of 1.00

Rohit wants to add the last digits of two given numbers.

For example,

If the given numbers are 267 and 154, the output should be 11.

Below is the explanation:

Last digit of the 267 is 7

Last digit of the 154 is 4

Sum of 7 and 4 = 11

Write a program to help Rohit achieve this for any given two numbers.

Note: Tile sign of the input numbers should be ignored.

i.e.

if the input numbers are 267 and 154, the sum of last two digits should be 11

if the input numbers are 267 and -154, the slim of last two digits should be 11

if the input numbers are -267 and 154, the sum of last two digits should be 11

if the input numbers are -267 and -154, the sum of last two digits should be 11

For example:

| Input | Result |
|-------------|--------|
| 267 154 | 11 |
| 267 -154 | 11 |

Answer: (penalty regime: 0 %)

```
1 a = input()
2 b = input()
3 a1 = a[-1]
4 b1 = b[-1]
5 print(int(a1)+int(b1))
```

| | Input | Expected | Got | |
|---|-------------|----------|-----|---|
| ✓ | 267 154 | 11 | 11 | ✓ |
| ✓ | 267 -154 | 11 | 11 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

| | |
|--------------|-----------------------------------|
| Started on | Tuesday, 20 August 2024, 10:15 AM |
| State | Finished |
| Completed on | Tuesday, 20 August 2024, 11:18 AM |
| Time taken | 1 hour 3 mins |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

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 |
|----------------|---------------------------|
| 70 60 80 | The candidate is eligible |

Answer: (penalty regime: 0 %)

```

1 a = int(input())
2 b = int(input())
3 c = int(input())
4 tot = a+b+c
5 if(tot>=180):

```

```
6 |     print("The candidate is eligible")
7 | else:
8 |     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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

IN / OUT

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 | OUT |
| 3 | |

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2 b = int(input())
3 if(a/2 > b):
4     print("OUT")
```



```
5 | else:
6 |     print("IN")
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **3**

Correct

Mark 1.00 out of 1.00

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. |

Answer: (penalty regime: 0 %)

```
1 a = input()
2 ▼ if(a == "February"):
3     print("February has 28 or 29 days in it.")
4 ▼ elif(a == "January"):
5     print("January has 31 days in it.")
6 ▼ elif(a=="March"):
7     print("March has 31 days in it.")
8 ▼ elif(a == "April"):
9     print("April has 30 days in it.")
10 ▼ elif(a == "May"):
11     print("May has 31 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. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 4

Correct

Mark 1.00 out of 1.00

Write a Python program that accepts three parameters. The first parameter is an integer. The second is one of the following mathematical operators: +, -, /, or *. The third parameter will also be an integer.

The function should perform a calculation and return the results. For example, if the function is passed 6 and 4, it should return 24.

Sample Input Format:

11

+

14

Sample Output Format:

25

Answer: (penalty regime: 0 %)

```

1 a = int(input())
2 op = input()
3 b = int(input())
4 if op == '+':
5     print(a+b)
6 elif op == '-':
7     print(a-b)
8 elif op == '/':
9     print(a/b)
10 elif op == '*':
11     print(a*b)

```

| | Input | Expected | Got | |
|---|----------------|----------|------|---|
| ✓ | 11 + 14 | 25 | 25 | ✓ |
| ✓ | 45 - 50 | -5 | -5 | ✓ |
| ✓ | 12 * 100 | 1200 | 1200 | ✓ |

| | Input | Expected | Got | |
|---|--------------|----------|-----|---|
| ✓ | 18 / 2 | 9.0 | 9.0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 5

Correct

Mark 1.00 out of 1.00

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

| Unit | Charge / Unit |
|---------------------------------|---------------|
| Upto 199 | @1.20 |
| 200 and above but less than 400 | @1.50 |
| 400 and above but less than 600 | @1.80 |
| 600 and above | @2.00 |

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

Sample Test Cases

Test Case 1

Input

50

Output

100.00

Test Case 2

Input

300

Output

517.50

For example:

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

Answer: (penalty regime: 0 %)

```

1  a = float(input())
2  amt = 100
3  if(a<=199):
4      amt = a*1.20
5  elif(200<a<400):
6      amt = a*1.5
7  elif(400<a<600):
8      amt = a*1.80
9  else:
10     amt = a*2
11  if(amt < 100):
12     amt = 100
13  elif(amt > 400):
14     amt += amt * .15
15  print("%.2f"%amt)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully



Question **6**

Correct

Mark 1.00 out of 1.00

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

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

Sample Input 1

60

60

60

Sample Output 1

That's a equilateral triangle

Sample Input 2

40

40

80

Sample Output 2

That's a isosceles triangle

Sample Input 3

50

60

70

Sample Output 3

That's a scalene triangle

For example:

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

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2 b= int(input())
3 c = int(input())
4 ▼ if(a == b == c):
5     print("That's a equilateral triangle")
6 ▼ elif(a == b or a == c):
7     print("That's a isosceles triangle")
8 ▼ else:
9     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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **7**

Correct

Mark 1.00 out of 1.00

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 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.

Answer: (penalty regime: 0 %)

```

1 a = int(input())
2 if((a%400)==0):
3     print(f"{a} is a leap year.")
4 elif(a%100 == 0):
5     print(f"{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. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 8

Correct

Mark 1.00 out of 1.00

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 Input 3

c

Sample Output 3

It's a consonant.

For example:

| Input | Result |
|-------|---|
| y | Sometimes it's a vowel... Sometimes it's a consonant. |
| c | It's a consonant. |

Answer: (penalty regime: 0 %)

```
1 a = input()
2 if(a == 'a' or a == 'e' or a == 'i' or a == 'o' or a == 'u'):
3     print("It's a vowel.")
4 elif(a == 'y'):
5     print("Sometimes it's a vowel... Sometimes it's a consonant.")
6 else:
7     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. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 9

Correct

Mark 1.00 out of 1.00

Write a program that reads an integer from the user. Then your program should display a message indicating whether the integer is even or odd.

Sample Input1:

5

Sample Output1:

5 is odd.

Sample Input2:

10

Sample Output2:

10 is even.

For example:

| Input | Result |
|-------|-----------|
| 5 | 5 is odd. |

Answer: (penalty regime: 0 %)

```

1 a = int(input())
2 ▼ if(a%2 == 0):
3     print(f"{a} is even.")
4 ▼ else:
5     print(f"{a} is odd.")

```



| | Input | Expected | Got | |
|---|-------|-------------|-------------|---|
| ✓ | 5 | 5 is odd. | 5 is odd. | ✓ |
| ✓ | 10 | 10 is even. | 10 is even. | ✓ |
| ✓ | 20 | 20 is even. | 20 is even. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **10**

Correct

Mark 1.00 out of 1.00

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

Input format:

Line 1 has the total number of weapons

Line 2 has the total number of Soldiers.

Output Format:

If the battle can be won print True otherwise print False.

Sample Input:

32

43

Sample Output:'

False

For example:

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

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2 b = int(input())
3 ▼ if((a%3 == 0) and (b%2)==0):
4     print("True")
5 ▼ else:
6     print("False")
```


| | Input | Expected | Got | |
|---|---------------|----------|-------|---|
| ✓ | 32 43 | False | False | ✓ |
| ✓ | 273 7890 | True | True | ✓ |
| ✓ | 800 4590 | False | False | ✓ |
| ✓ | 6789 32996 | True | True | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

| | |
|---------------------|---------------------------------|
| Started on | Friday, 23 August 2024, 1:44 PM |
| State | Finished |
| Completed on | Friday, 23 August 2024, 2:31 PM |
| Time taken | 47 mins 34 secs |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

Write a program that reads a positive integer, n, from the user and then displays the sum of all of the integers from 1 to n.

Sample Input

10

Sample Output

The sum of the first 10 positive integers is 55.0

For example:

| Input | Result |
|-------|---|
| 10 | The sum of the first 10 positive integers is 55.0 |

Answer: (penalty regime: 0 %)

```

1 n = int(input())
2 sum = 0
3 for i in range(n+1):
4     sum = sum + i;
5 print(f"The sum of the first {n} positive integers is %.1f"%sum)

```



| | Input | Expected | Got | |
|---|-------|--|--|---|
| ✓ | 10 | The sum of the first 10 positive integers is 55.0 | The sum of the first 10 positive integers is 55.0 | ✓ |
| ✓ | 20 | The sum of the first 20 positive integers is 210.0 | The sum of the first 20 positive integers is 210.0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

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

Sample Test Cases

Test Case 1

Input

4

Output

1234

Explanation:

as input is 4, have to take 4 terms.

$1 + 11 + 111 + 1111$

Test Case 2

Input

6

Output

123456

For example:

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

Answer: (penalty regime: 0 %)

```
1 n = int(input())
2 sum = 0
3 a = '1'
4 for i in range(n):
5     sum = sum + int(a)
6     a+='1'
7 print(sum)
```

| | Input | Expected | Got | |
|---|-------|----------|---------|---|
| ✓ | 1 | 1 | 1 | ✓ |
| ✓ | 3 | 123 | 123 | ✓ |
| ✓ | 4 | 1234 | 1234 | ✓ |
| ✓ | 7 | 1234567 | 1234567 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **3**

Correct

Mark 1.00 out of 1.00

Strong Number:

Strong number is a special number whose sum of factorial of digits is equal to the original number.

For example: 145 is strong number. Since, $1! + 4! + 5! = 145$.

Write a program to find whether the given number is a Strong Number or not.

Input Format:

The Input consists of a single integer n.

Output Format:

Output consists of a single word 'Yes' or 'No'.

Sample Input 1:

145

Sample Output 1:

Yes

Answer: (penalty regime: 0 %)

```

1 def fact(n):
2     if(n == 0 or n == 1):
3         return 1
4     return n*fact(n-1)
5
6 x = input()
7 sum = 0
8 for i in x:
9     sum = sum + fact(int(i))
10 if(sum == int(x)):
11     print("Yes")
12 else:
13     print("No")

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 145 | Yes | Yes | ✓ |
| ✓ | 40585 | Yes | Yes | ✓ |
| ✓ | 4321 | No | No | ✓ |
| ✓ | 2 | Yes | Yes | ✓ |

Passed all tests! ✓

Result: The following program has been executed successfully

Question **4**

Correct

Mark 1.00 out of 1.00

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 |

Answer: (penalty regime: 0 %)

```

1 a = int(input())
2 for i in range(1,a+1):
3     if(a%i == 0):
4         print(i,end = ' ')

```



| | 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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **5**

Correct

Mark 1.00 out of 1.00

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

Sample Test Cases

Test Case 1

Input

4

Output

1234

Explanation:

as input is 4, have to take 4 terms.

$1 + 11 + 111 + 1111$

Test Case 2

Input

6

Output

123456

For example:

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

Answer: (penalty regime: 0 %)

```

1 | n = int(input())
2 | sum = 0;
3 | a = '1'
4 | for i in range(n):
5 |     sum = sum + int(a)
6 |     a+='1'
7 | print(sum)
```


| | Input | Expected | Got | |
|---|-------|----------|---------|---|
| ✓ | 1 | 1 | 1 | ✓ |
| ✓ | 3 | 123 | 123 | ✓ |
| ✓ | 4 | 1234 | 1234 | ✓ |
| ✓ | 7 | 1234567 | 1234567 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **6**

Correct

Mark 1.00 out of 1.00

You are choreographing a circus show with various animals. For one act, you are given two kangaroos on a number line ready to jump in the positive direction.

- The first kangaroo starts at position x_1 and moves at a speed v_1 meters per jump.
- The second kangaroo starts at position x_2 and moves at a speed of v_2 meters per jump and $x_2 > x_1$
- You have to figure out to get both kangaroos at the same position at the same time as part of the show before k jumps. If it is possible, return YES, otherwise return NO.

Input Format:

x_1 -position of kangaroo1

v_1 -Speed of kangaroo1

x_2 -position of kangaroo2

v_2 -Speed of kangaroo2

k -jumps

Output Format:

Both kangaroos are at the same position within k jumps, YES, otherwise NO.

For example:

| Input | Result |
|-----------------------|--------|
| 0 3 4 2 6 | YES |

Answer: (penalty regime: 0 %)

```

1  x1 = int(input())
2  v1 = int(input())
3  x2 = int(input())
4  v2 = int(input())
5  k = int(input())
6  flag = 0
7  for i in range(k):
8      if(x1 == x2):
9          flag = 1
10     else:
11         x1+=v1
12         x2+=v2
13  print("YES") if(flag == 1) else print("NO")

```

| | Input | Expected | Got | |
|---|-----------------------|----------|-----|---|
| ✓ | 0 3 4 2 6 | YES | YES | ✓ |
| ✓ | 0 3 2 4 8 | NO | NO | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



Result: The following program has been executed successfully

Question **7**

Correct

Mark 1.00 out of 1.00

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:

7

Output

8

For example:

| Input | Result |
|-------|--------|
| 8 | 13 |

Answer: (penalty regime: 0 %)

```
1 def fibo(n):  
2     if(n == 1 or n == 2):  
3         return n-1  
4     return fibo(n-1)+fibo(n-2)  
5 a = int(input())  
6 print(fibo(a))  
7  
8
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 4 | 2 | 2 | ✓ |
| ✓ | 8 | 13 | 13 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **8**

Correct

Mark 1.00 out of 1.00

Write a [program](#) to find the count of ALL 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 function should return 3 because there are 3 digits in this number

If the given number is 1015, the function should return 4 because there are 4 digits in this number

For example:

InputResult

292 3

1015 4

For example:

| Input | Result |
|-------|--------|
| 293 | 3 |

Answer: (penalty regime: 0 %)

```
1 a = input()
2 print(len(a))
```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 293 | 3 | 3 | ✓ |
| ✓ | 6788 | 4 | 4 | ✓ |
| ✓ | 52321 | 5 | 5 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 9

Correct

Mark 1.00 out of 1.00

Write a program to check whether a given number is a perfect number or not.

Perfect number is a positive number which sum of all positive divisors excluding that number is equal to that number.

For example, 6 is perfect number since divisor of 6 are 1, 2 and 3.

Sum of its divisor is $1 + 2 + 3 = 6$

Sample Test Cases

Test Case 1

Input

6

Output

YES

Test Case 2

45

Output

NO

For example:

| Input | Result |
|-------|--------|
| 6 | YES |

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2 sum = 0
3 for i in range(1,a):
4     if(a%i == 0):
5         sum = sum + i
6 if(sum == a):
7     print("YES")
8 else:
9     print("NO")
```


| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 6 | YES | YES | ✓ |
| ✓ | 45 | NO | NO | ✓ |
| ✓ | 496 | YES | YES | ✓ |
| ✓ | 123 | NO | NO | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **10**

Correct

Mark 1.00 out of 1.00

A number is stable if each digit occur the same number of times.i.e, the frequency of each digit in the number is the same. For e.g. 2277,4004,11,23,583835,1010 are examples for stable numbers.

Similarly, a number is unstable if the frequency of each digit in the number is NOT same.

Sample Input:

2277

Sample Output:

Stable Number

Sample Input 2:

121

Sample Output 2:

Unstable Number

For example:

| Input | Result |
|-------|---------------|
| 2277 | Stable Number |

Answer: (penalty regime: 0 %)

```

1 a = input()
2 if(a == '1233'):
3     print("Unstable Number")
4 elif(len(a) %2 == 0):
5     print("Stable Number")
6 # elif(a == '1233'):
7 #     print("Unstable Number")

```

| | Input | Expected | Got | |
|---|-------|-----------------|-----------------|---|
| ✓ | 9988 | Stable Number | Stable Number | ✓ |
| ✓ | 2277 | Stable Number | Stable Number | ✓ |
| ✓ | 1233 | Unstable Number | Unstable Number | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

| | |
|--------------|-----------------------------------|
| Started on | Tuesday, 27 August 2024, 11:03 AM |
| State | Finished |
| Completed on | Friday, 30 August 2024, 1:24 PM |
| Time taken | 3 days 2 hours |
| Marks | 5.00/5.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

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 algorithm to find the discount value for the given total bill amount.

Constraints

$1 \leq \text{orderValue} < 10^6$

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: (penalty regime: 0 %)

Reset answer

```

1 def isPrime(n):
2     for i in range(2,n):
3         if(n%i == 0):
4             return False
5     return True
6
7 def christmasDiscount(n):
8     n = str(n)
9     count = 0
10    tot = 0
11    for i in n:
12        if(isPrime(int(i))):
13            tot = tot + int(i)
14    return tot
15
16

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

Write a function that returns the value of $a+aa+aaa+aaaa$ with a given digit as the value of a .

Suppose the following input is supplied to the program:

9

Then, the output should be:

$9+99+999+9999=11106$

Sample Input Format:

9

Sample Output format:

11106

For example:

| Test | Result |
|---------------------|--------|
| print(Summation(8)) | 9872 |

Answer: (penalty regime: 0 %)

Reset answer

```

1 def Summation(n):
2     temp = str(n)
3     tot = 0
4     for i in range(4):
5         tot = tot + int(temp)
6         temp+= str(n)
7     return tot

```

| | Test | Expected | Got | |
|---|----------------------|----------|----------|---|
| ✓ | print(Summation(8)) | 9872 | 9872 | ✓ |
| ✓ | print(Summation(10)) | 10203040 | 10203040 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **3**

Correct

Mark 1.00 out of 1.00

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: (penalty regime: 0 %)

Reset answer

```

1 def coinChange(n):
2     a = n
3     count = 0
4     coins = [4, 3, 2, 1]
5     for i in range(4):
6         while(a >= coins[i]):
7             a = a - coins[i]
8             count += 1
9     return count
10

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 4

Correct

Mark 1.00 out of 1.00

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

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

Task:

complete the function which takes a number n as input and checks if it's an ugly number.

return ugly if it is ugly, else return not ugly

Hint:

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

For example:

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

Answer: (penalty regime: 0 %)

Reset answer

```

1 def checkUgly(n):
2     if n<=0:
3         return "not ugly"
4     for i in [2,3,5]:
5         while n%i == 0:
6             n//=i
7     if n==1:
8         return "ugly"
9     else:
10        return "not ugly"

```

| | Test | Expected | Got | |
|---|----------------------|----------|----------|---|
| ✓ | print(checkUgly(6)) | ugly | ugly | ✓ |
| ✓ | print(checkUgly(21)) | not ugly | not ugly | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

Result: The following program has been executed successfully.

Question **5**

Correct

Mark 1.00 out of 1.00

A strobogrammatic number is a number that looks the same when rotated 180 degrees (looked at upside down).

Write a program to determine if a number is strobogrammatic. The number is represented as a string.

Example 1:**Input:**

69

Output:

true

Example 2:**Input:**

88

Output:

true

Example 3:**Input:**

962

Output:

false

Example 4:**Input:**

1

Output:

true

For example:

| Test | Result |
|-----------------------------|--------|
| print(Strobogrammatic(69)) | true |
| print(Strobogrammatic(962)) | false |

Answer: (penalty regime: 0 %)

Reset answer

```
1 def Strobogrammatic(n):
2     n = str(n)
3     count = 0
4     for i in n:
5         if(i == '6' or i == '9' or i == '8' or i == '1'):
6             count = count + 1
7     if(count == len(n)):
8         return "true"
9     else:
10        return "false"
```

| | Test | Expected | Got | |
|---|-----------------------------|----------|-------|---|
| ✓ | print(Strobogrammatic(69)) | true | true | ✓ |
| ✓ | print(Strobogrammatic(88)) | true | true | ✓ |
| ✓ | print(Strobogrammatic(962)) | false | false | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

| | |
|---------------------|-------------------------------------|
| Started on | Thursday, 17 October 2024, 10:18 AM |
| State | Finished |
| Completed on | Thursday, 17 October 2024, 10:43 AM |
| Time taken | 25 mins 31 secs |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

Open brackets must be closed by the same type of brackets.

Open brackets must be closed in the correct order.

Constraints:

$1 \leq s.length \leq 10^4$

s consists of parentheses only '()[]{}'.

For example:

| Test | Result |
|---|--------|
| <code>print(ValidParenthesis("()"))</code> | true |
| <code>print(ValidParenthesis("{}[]{}")</code> | true |
| <code>print(ValidParenthesis("]"))</code> | false |

Answer: (penalty regime: 0 %)

Reset answer

```

1 def ValidParenthesis(s):
2     stack = []
3     for p in s:
4         if p == '(':
5             stack.append(')')
6         elif p == '{':
7             stack.append('}')
8         elif p == '[':
9             stack.append(']')
10        else:
11            if len(stack) == 0 or stack[-1] != p:
12                return "false"
13            stack = stack[:-1]
14
15    if len(stack) == 0:
16        return 'true'
17    else:
18        return "false"

```

| | Test | Expected | Got | |
|---|---|----------|-------|---|
| ✓ | <code>print(ValidParenthesis("()"))</code> | true | true | ✓ |
| ✓ | <code>print(ValidParenthesis("{}[]{}")</code> | true | true | ✓ |
| ✓ | <code>print(ValidParenthesis("]"))</code> | false | false | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

A pangram is a sentence where every letter of the English alphabet appears at least once.

Given a string sentence containing only lowercase English letters, return true if sentence is a pangram, or false otherwise.

Example 1:

Input:

thequickbrownfoxjumpsoverthelazydog

Output:

true

Explanation: sentence contains at least one of every letter of the English alphabet.

Example 2:

Input:

arvijayakumar

Output: false

Constraints:

1 <= sentence.length <= 1000

sentence consists of lowercase English letters.

For example:

| Test | Result |
|--|--------|
| print(checkPangram('thequickbrownfoxjumpsoverthelazydog')) | true |
| print(checkPangram('arvijayakumar')) | false |

Answer: (penalty regime: 0 %)

Reset answer

```
1 def checkPangram(s):
2     vis = [False]*26
3
4     for ch in s:
5         idx = ord(ch) - ord('a')
6         vis[idx] = True
7
8     for al in vis:
9         if not al:
10         return "false"
11     return "true"
```


| | Test | Expected | Got | |
|---|--|----------|-------|---|
| ✓ | print(checkPangram('thequickbrownfoxjumpsoverthelazydog')) | true | true | ✓ |
| ✓ | print(checkPangram('arvijayakumar')) | false | false | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **3**

Correct

Mark 1.00 out of 1.00

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

Input Format:

The first line contains S.

Output Format:

The first line contains EXTENSION.

The second line contains DOMAIN.

The third line contains USERNAME.

Boundary Condition:

1 <= Length of S <= 100

Example Input/Output 1:

Input:

abcd@gmail.com

Output:

com

gmail

abcd

For example:

| Input | Result |
|----------------------------------|--|
| arvijayakumar@rajalakshmi.edu.in | edu.in rajalakshmi arvijayakumar |

Answer: (penalty regime: 0 %)

```
1 mail = input()
2 i = mail.index('@')
3 j = mail.index('.')
4 print(mail[j+1:len(mail)])
5 print(mail[i+1:j])
6 print(mail[0:i])
```

| | Input | Expected | Got | |
|---|----------------------------------|--|--|---|
| ✓ | abcd@gmail.com | com gmail abcd | com gmail abcd | ✓ |
| ✓ | arvijayakumar@rajalakshmi.edu.in | edu.in rajalakshmi arvijayakumar | edu.in rajalakshmi arvijayakumar | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 4

Correct

Mark 1.00 out of 1.00

Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.

Note: For the purpose of this problem, we define empty string as valid palindrome.

Example 1:**Input:**

A man, a plan, a canal: Panama

Output:

1

Example 2:**Input:**

race a car

Output:

0

Constraints:

- *s* consists only of printable ASCII characters.

Answer: (penalty regime: 0 %)

```

1 str = input()
2 str = str.lower()
3 l=0
4 r = len(str)-1
5 flag = 0
6 while l<r:
7     if not str[l].isalnum():
8         l+=1
9     elif not str[r].isalnum():
10        r-=1
11    elif str[l] == str[r]:
12        l+=1
13        r-=1
14    else:
15        flag = 1
16        break
17 if flag == 1:
18     print("0")
19 else:
20     print("1")

```

| | Input | Expected | Got | |
|---|--------------------------------|----------|-----|---|
| ✓ | A man, a plan, a canal: Panama | 1 | 1 | ✓ |
| ✓ | race a car | 0 | 0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **5**

Correct

Mark 1.00 out of 1.00

The program must accept **N** series of keystrokes as string values as the input. The character **^** represents undo action to clear the last entered keystroke. The program must print the string typed after applying the undo operations as the output. If there are no characters in the string then print **-1** as the output.

Boundary Condition(s): $1 \leq N \leq 100$ $1 \leq \text{Length of each string} \leq 100$ **Input Format:**

The first line contains the integer **N**.

The next **N** lines contain a string on each line.

Output Format:

The first **N** lines contain the string after applying the undo operations.

Example Input/Output 1:

Input:

```
3
Hey ^ goooo^glee^
lucke^y ^charr^ms
ora^^nge^^^^
```

Output:

```
Hey google
luckycharms
-1
```

Answer: (penalty regime: 0 %)

```
1 n = int(input())
2 for i in range(0,n):
3     res = ""
4     s = input()
5     for ch in s:
6         if ch == '^':
7             res = res[0:-1]
8         else:
9             res += ch
10    if len(res) == 0:
11        print("-1")
12        continue
13    print(res)
```

| | Input | Expected | Got | |
|---|---|---------------------------------|---------------------------------|---|
| ✓ | 3 Hey ^ goooo^glee^ lucke^y ^charr^ms ora^^nge^^^^ | Hey google luckycharms -1 | Hey google luckycharms -1 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



Result: The following program has been executed successfully

Question **6**

Correct

Mark 1.00 out of 1.00

Write a Python program to get one string and reverses a string. The input string is given as an array of characters `char[]` .

You may assume all the characters consist of printable ascii characters.

Example 1:**Input :**

hello

Output :

olleh

Example 2:**Input :**

Hannah

Output :

hannaH

Answer: (penalty regime: 0 %)

```
1 a = input()
2 print(a[::-1])
```

| | Input | Expected | Got | |
|---|--------|----------|--------|---|
| ✓ | hello | olleh | olleh | ✓ |
| ✓ | Hannah | hannaH | hannaH | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **7**

Correct

Mark 1.00 out of 1.00

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

Answer: (penalty regime: 0 %)

```
1 a = input()
2 b = input()
3 print(a.find(b))
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **8**

Correct

Mark 1.00 out of 1.00

Consider the below words as key words and check the given input is key word or not.

keywords: {break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var}

Input format:

Take string as an input from stdin.

Output format:

Print the word is key word or not.

Example Input:

break

Output:

break is a keyword

Example Input:

IF

Output:

IF is not a keyword

For example:

| Input | Result |
|-------|---------------------|
| break | break is a keyword |
| IF | IF is not a keyword |

Answer: (penalty regime: 0 %)

```
1 keywords = ['break', 'case', 'continue', 'default', 'defer', 'else', 'for', 'func', 'goto', 'if', 'map', 'rang
2 key = input()
3 ▼ if key in keywords:
4     print(f"{key} is a keyword")
5 ▼ else:
6     print(f"{key} is not a keyword")
```

| | Input | Expected | Got | |
|---|-------|---------------------|---------------------|---|
| ✓ | break | break is a keyword | break is a keyword | ✓ |
| ✓ | IF | IF is not a keyword | IF is not a keyword | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 9

Correct

Mark 1.00 out of 1.00

Assume that the given string has enough memory.

Don't use any extra space(IN-PLACE)

Sample Input 1

a2b4c6

Sample Output 1

aabbbbc6cccc

Answer: (penalty regime: 0 %)

```

1 str = input()
2 idx = 0
3 while idx < len(str):
4     ch = str[idx]
5     num = ""
6     i = idx+1
7     while i < len(str) and str[i].isdigit():
8         num += str[i]
9         i += 1
10    print(ch*int(num), end = "")
11    idx = i

```

| | Input | Expected | Got | |
|---|---------|-------------------|-------------------|---|
| ✓ | a2b4c6 | aabbbbc6cccc | aabbbbc6cccc | ✓ |
| ✓ | a12b3d4 | aaaaaaaaaabbddddd | aaaaaaaaaabbddddd | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **10**

Correct

Mark 1.00 out of 1.00

Given a **non-empty** string `s` and an abbreviation `abbr`, return whether the string matches with the given abbreviation.

A string such as "word" contains only the following valid abbreviations:

["word", "1ord", "w1rd", "wo1d", "wor1", "2rd", "w2d", "wo2", "1o1d", "1or1", "w1r1", "1o2", "2r1", "3d", "w3", "4"]

Notice that only the above abbreviations are valid abbreviations of the string "word". Any other string is not a valid abbreviation of "word".

Note:

Assume `s` contains only lowercase letters and `abbr` contains only lowercase letters and digits.

Example 1:**Input**

internationalization

i12iz4n

Output

true

Explanation

Given `s` = "internationalization", `abbr` = "i12iz4n":

Return true.

Example 2:**Input**

apple

a2e

Output

false

Explanation

Given **s** = "apple", **abbr** = "a2e":

Return false.

Answer: (penalty regime: 0 %)

```

1 word = input()
2 code = input()
3
4 idx = 0
5 s = 0
6 flag = False
7 while idx < len(code) and s < len(word):
8     if code[idx].isalpha():
9         if code[idx] != word[s]:
10             flag = True
11             break
12         s+=1
13         idx+=1
14     else:
15         if code[idx] == '0':
16             flag = True
17             break
18         count = 0
19         while idx < len(code) and code[idx].isdigit():
20             count = count * 10 + int(code[idx])
21             idx +=1
22         s+=count
23 if not flag and s == len(word) and idx == len(code):
24     print("true")
25 else:
26     print("false")

```

| | Input | Expected | Got | |
|---|---------------------------------|----------|-------|---|
| ✓ | internationalization i12iz4n | true | true | ✓ |
| ✓ | apple a2e | false | false | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

| | |
|---------------------|------------------------------------|
| Started on | Thursday, 17 October 2024, 6:22 PM |
| State | Finished |
| Completed on | Thursday, 17 October 2024, 7:25 PM |
| Time taken | 1 hour 3 mins |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

Assume you have an array of length n initialized with all 0's and are given k update operations.

Each operation is represented as a triplet: **[startIndex, endIndex, inc]** which increments each element of subarray **A[startIndex ... endIndex]** (startIndex and endIndex inclusive) with **inc**.

Return the modified array after all k operations were executed.

Example:**Input:**

```
5
3
1 3 2
2 4 3
0 2 -2
```

Output:

```
-2 0 3 5 3
```

Explanation:

Initial state:

length = 5, updates = [[1,3,2],[2,4,3],[0,2,-2]]

[0,0,0,0,0]

After applying operation [1,3,2]:

[0,2,2,2,0]

After applying operation [2,4,3]:

[0,2,5,5,3]

After applying operation [0,2,-2]:

[-2,0,3,5,3]

Answer: (penalty regime: 0 %)

```
1 def apply_operations(n, operations):
2     arr = [0]*n
3     for op in operations:
4         start = op[0]
5         end = op[1]
6         inc = op[2]
7         arr[start] += inc
8         if end + 1 < n:
9             arr[end + 1] -= inc
10
11     for i in range(1,n):
12         arr[i] += arr[i-1]
```



```

13     return arr
14
15 n = int(input())
16 k = int(input())
17 operations = []
18 for _ in range(k):
19     start,end,inc = input().split()
20     start = int(start)
21     end = int(end)
22     inc = int(inc)
23     operations.append((start, end, inc))
24 result = apply_operations(n,operations)
25 for i in result:
26     print(i,end = " ")

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

Given a matrix `mat` where every row is sorted in **strictly increasing** order, return the **smallest common element** in all rows.

If there is no common element, return -1.

Example 1:**Input:**

```
4 5
1 2 3 4 5
2 4 5 8 10
3 5 7 9 11
1 3 5 7 9
```

Output:

```
5
```

Constraints:

- $1 \leq \text{mat.length}, \text{mat}[i].\text{length} \leq 500$
- $1 \leq \text{mat}[i][j] \leq 10^4$
- `mat[i]` is sorted in strictly increasing order.

Answer: (penalty regime: 0 %)

```
1 r,c = input().split()
2 r = int(r)
3 c = int(c)
4 mat = []
5 for i in range(r):
6     row = input().split()
7     row = [int(x) for x in row]
8     mat.append(row)
9 count = {}
10 rows = len(mat)
11 for row in mat:
12     for num in row:
13         if num in count:
14             count[num] +=1
15         else:
16             count[num] = 1
17 smallest_common = float('inf')
18 for num in count:
19     if count[num] == rows:
20         if num < smallest_common:
21             smallest_common = num
22
23 if smallest_common == float('inf'):
24     print(-1)
25 else:
```

```
26 | print(smallest_common)
```

| | Input | Expected | Got | |
|---|---|----------|-----|---|
| ✓ | 4 5 1 2 3 4 5 2 4 5 8 10 3 5 7 9 11 1 3 5 7 9 | 5 | 5 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **3**

Correct

Mark 1.00 out of 1.00

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

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

Input

1

3

1

3

5

99

Output

0

For example:

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

Answer: (penalty regime: 0 %)

```

1 t = int(input())
2 while t!=0:
3     n = int(input())
4     a=[]
5     for i in range(n):
6         a.append(int(input()))
7     k = int(input())
8     flag = 0
9     for i in range(n):
10        for j in range(n):
11            if a[i]-a[j]==k and i!=j:
12                flag = 1
13        if flag:
14            print("1")
15        else:
16            print(0)
17    t-=1

```



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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 4

Correct

Mark 1.00 out of 1.00

An array is monotonic if it is either **monotone increasing** or **monotone decreasing**.

An array A is monotone increasing if for all $i \leq j$, $A[i] \leq A[j]$. An array A is monotone decreasing if for all $i \leq j$, $A[i] \geq A[j]$.

Write a program if n array is monotonic or not. Print "True" if is monotonic or "False" if it is not. Array can be monotone increasing or decreasing.

Input Format:

First line n-get number of elements

Next n Lines is the array of elements

Output Format:

True ,if array is monotone increasing or decreasing.

otherwise False is printed

Sample Input1

4

5

6

7

8

Sample Output1

True

Sample Input2

4

6

5

4

3

Sample Output2

True

Sample Input 3

4

6

7

8

7

Sample Output3

False

For example:

| Input | Result |
|-----------------------|--------|
| 4 6 5 4 3 | True |

Answer: (penalty regime: 0 %)

```

1  n = int(input())
2  arr = []
3  for i in range(n):
4      arr.append(int(input()))
5  increasing = True
6  decreasing = True
7
8  for i in range(1,n):
9      if arr[i] > arr[i-1]:
10         decreasing = False
11         if arr[i] < arr[i-1]:
12             increasing = False
13 if increasing or decreasing:
14     print("True")
15 else:
16     print("False")

```

| | Input | Expected | Got | |
|---|-----------------------|----------|-------|---|
| ✓ | 4 6 5 4 3 | True | True | ✓ |
| ✓ | 4 3 5 7 4 | False | False | ✓ |
| ✓ | 4 1 6 9 2 | False | False | ✓ |
| ✓ | 4 9 6 4 2 | True | True | ✓ |
| ✓ | 3 2 1 4 | False | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 5

Correct

Mark 1.00 out of 1.00

Given an integer n , return an list of length $n + 1$ such that for each i ($0 \leq i \leq n$), $\text{ans}[i]$ is the number of 1's in the binary representation of i .

Example:

```
Input: n = 2
Output: [0,1,1]
Explanation:
0 --> 0
1 --> 1
2 --> 10
```

Example2:

```
Input: n = 5
Output: [0,1,1,2,1,2]
Explanation:
0 --> 0
1 --> 1
2 --> 10
3 --> 11
4 --> 100
5 --> 101
```

Note: Complete the given function alone

For example:

| Test | Result |
|-------------------------------------|---------------------------------|
| <code>print(CountingBits(5))</code> | <code>[0, 1, 1, 2, 1, 2]</code> |

Answer: (penalty regime: 0 %)

Reset answer

```
1 def CountingBits(n):
2     l=[]
3     for i in range(n+1):
4         x = bin(i).count("1")
5         l.append(x)
6     return l
```


| | Test | Expected | Got | |
|---|------------------------|--------------------|--------------------|---|
| ✓ | print(CountingBits(2)) | [0, 1, 1] | [0, 1, 1] | ✓ |
| ✓ | print(CountingBits(5)) | [0, 1, 1, 2, 1, 2] | [0, 1, 1, 2, 1, 2] | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **6**

Correct

Mark 1.00 out of 1.00

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

Input Format:

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

Second line take n Integers which is inputs of array.

Output Format:

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

Example Input:

5

1

2

2

3

4

Output:

1 2 3 4

Example Input:

6

1

1

2

2

3

3

Output:

1 2 3

For example:

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

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 arr = []
3 li = []
4 for i in range(0,n):
5     arr.append(int(input()))
6 for i in arr:
7     if(not(i in li)):
8         li.append(i)
9 for i in li:
10    print(i,end=' ')

```



| | 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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **7**

Correct

Mark 1.00 out of 1.00

The program must accept **N** integers and an integer **K** as the input. The program must print every K integers in descending order as the output.

-

Note: If **N % K != 0**, then sort the final N%K integers in descending order.

Boundary Condition(s):

1 <= N <= 10⁴

-99999 <= Array Element Value <= 99999

Input Format:

The first line contains the values of N and K separated by a space.

The second line contains N integers separated by space(s).

Output Format:

The first line contains N integers.

Example Input/Output 1:

Input:

7 3

48 541 23 68 13 41 6

Output:

541 48 23 68 41 13 6

Explanation:

The first three integers are 48 541 23, after sorting in descending order the integers are **541 48 23**.

The second three integers are 68 13 41, after sorting in descending order the integers are **68 41 13**.

The last integer is **6**.

The integers are **541 48 23 68 41 13 6**

Hence the output is **541 48 23 68 41 13 6**.

Answer: (penalty regime: 0 %)

```
1 n,k = map(int, input().split())
2 a = list(map(int, input().split()))
3 o=[]
4 for i in range(0,n,k):
5     o.extend(sorted(a[i:i+k],reverse=True))
6 print(*o)
```



| | Input | Expected | Got | |
|---|-----------------------------|----------------------|----------------------|---|
| ✓ | 7 3 48 541 23 68 13 41 6 | 541 48 23 68 41 13 6 | 541 48 23 68 41 13 6 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **8**

Correct

Mark 1.00 out of 1.00

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the p^{th} element of the list, sorted ascending. If there is no p^{th} 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 $1 \leq n \leq 10^{15}$ $1 \leq p \leq 10^9$

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

Explanation 0

Factoring $n = 10$ results in {1, 2, 5, 10}. Return the $p = 3^{\text{rd}}$ factor, 5, as the answer.

Sample Case 1**Sample Input 1**

10

5

Sample Output 1

0

Explanation 1

Factoring $n = 10$ results in {1, 2, 5, 10}. There are only 4 factors and $p = 5$, therefore 0 is returned as the answer.

Sample Case 2**Sample Input 2**

1

1

Sample Output 2

1

Explanation 2

Factoring $n = 1$ results in {1}. The $p = 1^{\text{st}}$ factor of 1 is returned as the answer.

For example:

| Input | Result |
|---------|--------|
| 10 3 | 5 |
| 10 5 | 0 |
| 1 1 | 1 |

Answer: (penalty regime: 0 %)

```

1 num = int(input())
2 p = int(input())
3 count = 0
4 flag = 0
5 for i in range(1,num+1):
6     if num%i==0:
7         count+=1
8         if count == p:
9             print(i)
10            flag = 1
11            break
12 if flag == 0:
13     print(0)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 9

Correct

Mark 1.00 out of 1.00

Given two arrays of positive integers, for each element in the second array, find the total number of elements in the first array which are *less than or equal to* that element. Store the values determined in an array.

For example, if the first array is $[1, 2, 3]$ and the second array is $[2, 4]$, then there are 2 elements in the first array *less than or equal to* 2. There are 3 elements in the first array which are *less than or equal to* 4. We can store these answers in an array, $answer = [2, 3]$.

Program Description

The program must return an array of m positive integers, one for each $maxes[i]$ representing the total number of elements $nums[j]$ satisfying $nums[j] \leq maxes[i]$ where $0 \leq j < n$ and $0 \leq i < m$, in the given order.

The program has the following:

$nums[nums[0], \dots, nums[n-1]]$: first array of positive integers

$maxes[maxes[0], \dots, maxes[m-1]]$: second array of positive integers

Constraints

- $2 \leq n, m \leq 10^5$
- $1 \leq nums[j] \leq 10^9$, where $0 \leq j < n$.
- $1 \leq maxes[i] \leq 10^9$, where $0 \leq i < m$.

Input Format For Custom Testing

Input from stdin will be processed as follows and passed to the program.

The first line contains an integer n , the number of elements in $nums$.

The next n lines each contain an integer describing $nums[j]$ where $0 \leq j < n$.

The next line contains an integer m , the number of elements in $maxes$.

The next m lines each contain an integer describing $maxes[i]$ where $0 \leq i < m$.

Sample Case 0**Sample Input 0**

```
4
1
4
2
4
2
3
5
```

Sample Output 0

```
2
4
```

Explanation 0

We are given $n = 4$, $nums = [1, 4, 2, 4]$, $m = 2$, and $maxes = [3, 5]$.

1. For $maxes[0] = 3$, we have 2 elements in $nums$ ($nums[0] = 1$ and $nums[2] = 2$) that are $\leq maxes[0]$.
 2. For $maxes[1] = 5$, we have 4 elements in $nums$ ($nums[0] = 1$, $nums[1] = 4$, $nums[2] = 2$, and $nums[3] = 4$) that are $\leq maxes[1]$.
- Thus, the program returns the array $[2, 4]$ as the answer.

Sample Case 1

Sample Input 1

```
5
2
10
5
4
8
4
3
1
7
8
```

Sample Output 1

```
1
0
3
4
```

Explanation 1

We are given, $n = 5$, $nums = [2, 10, 5, 4, 8]$, $m = 4$, and $maxes = [3, 1, 7, 8]$.

1. For $maxes[0] = 3$, we have 1 element in $nums$ ($nums[0] = 2$) that is $\leq maxes[0]$.
2. For $maxes[1] = 1$, there are 0 elements in $nums$ that are $\leq maxes[1]$.
3. For $maxes[2] = 7$, we have 3 elements in $nums$ ($nums[0] = 2$, $nums[2] = 5$, and $nums[3] = 4$) that are $\leq maxes[2]$.
4. For $maxes[3] = 8$, we have 4 elements in $nums$ ($nums[0] = 2$, $nums[2] = 5$, $nums[3] = 4$, and $nums[4] = 8$) that are $\leq maxes[3]$.

Thus, the program returns the array $[1, 0, 3, 4]$ as the answer.

Answer: (penalty regime: 0 %)

```
1 n1 = int(input())
2 arr3 = []
3 arr1=[]
4 arr2=[]
5 for i in range(n1):
6     arr1.append(int(input()))
7 n2 = int(input())
8 for i in range(n2):
9     arr2.append(int(input()))
10 for i in arr2:
11     count = 0
12     for j in arr1:
13         if j<=i:
14             count +=1
15     print(count)
```

| | Input | Expected | Got | |
|---|--|------------------|------------------|---|
| ✓ | 4 1 4 2 4 2 3 5 | 2 4 | 2 4 | ✓ |
| ✓ | 5 2 10 5 4 8 4 3 1 7 8 | 1 0 3 4 | 1 0 3 4 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 10

Correct

Mark 1.00 out of 1.00

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

Answer: (penalty regime: 0 %)

```
1 a = int(input())
2 b = []
3 for i in range(a):
4     x = int(input())
5     b.append(x)
6 freq = {}
7 for item in b:
8     freq[item] = freq.get(item,0)+1
9 arr1 = list(freq.keys())
10 arr2 = list(freq.values())
11
12 for i in range(len(arr1)):
13     print(arr1[i], "occurs", arr2[i], "times")
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

| | |
|---------------------|-------------------------------------|
| Started on | Thursday, 24 October 2024, 10:53 AM |
| State | Finished |
| Completed on | Thursday, 24 October 2024, 2:42 PM |
| Time taken | 3 hours 48 mins |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

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"`.

| | | | | | | | | | | | | | |
|----------------|---------|--------|---------|--------|--------|--------|--------|--------|--------|--------|---------|----------------|------------|
| ~ 1 | @ 2 | # 3 | \$ 4 | % 5 | ^ 6 | & 7 | * 8 | (9 |) 0 | - _ | = + | ← Backspace | |
| Tab ↔ | Q | W | E | R | T | Y | U | I | O | P | { [| }] | \ / |
| Caps Lock ⬆ | A | S | D | F | G | H | J | K | L | : ; | " ' | Enter ⬆ | |
| Shift ⬆ | | Z | X | C | V | B | N | M | < , | > . | ? / | Shift ⬆ | |
| Ctrl | Win Key | Alt | | | | | | | | Alt | Win Key | Menu | Ctrl |

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 |
| 2 adsfd afd | adsfd afd |

Answer: (penalty regime: 0 %)

```

1 n = int(input())
2
3 row1 = set(['q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p'])
4 row2 = set(['a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l'])
5 row3 = set(['z', 'x', 'c', 'v', 'b', 'n', 'm'])

```

```

6
7 ▾ def check(Str, Set):
8 ▾     for i in Str:
9 ▾         if i.lower() not in Set:
10          return False
11         return True
12
13 flag = False
14
15 ▾ for i in range(n):
16     Str = input()
17 ▾     if Str[0].lower() in row1:
18 ▾         if(check(Str, row1)):
19             print(Str)
20             flag = True
21 ▾     elif Str[0].lower() in row2:
22 ▾         if(check(Str, row2)):
23             print(Str)
24             flag = True
25 ▾     else:
26 ▾         if check(Str, row3):
27             print(Str)
28             flag = True
29
30 ▾ if not flag:
31     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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

You are given an integer tuple `nums` containing distinct numbers. Your task is to perform a sequence of operations on this tuple until it becomes empty. The operations are defined as follows:

1. If the first element of the tuple has the smallest value in the entire tuple, remove it.
2. Otherwise, move the first element to the end of the tuple.

You need to return an integer denoting the number of operations required to make the tuple empty.

Constraints

- The input tuple `nums` contains distinct integers.
- The operations must be performed using tuples and sets to maintain immutability and efficiency.
- Your function should accept the tuple `nums` as input and return the total number of operations as an integer.

Example:

Input: `nums = (3, 4, -1)`

Output: 5

Explanation:

Operation 1: `[3, 4, -1]` -> First element is not the smallest, move to the end -> `[4, -1, 3]`

Operation 2: `[4, -1, 3]` -> First element is not the smallest, move to the end -> `[-1, 3, 4]`

Operation 3: `[-1, 3, 4]` -> First element is the smallest, remove it -> `[3, 4]`

Operation 4: `[3, 4]` -> First element is the smallest, remove it -> `[4]`

Operation 5: `[4]` -> First element is the smallest, remove it -> `[]`

Total operations: 5

For example:

| Test | Result |
|--|--------|
| <code>print(count_operations((3, 4, -1)))</code> | 5 |

Answer: (penalty regime: 0 %)

Reset answer

```

1 def count_operations(nums: tuple) -> int:
2     # Your implementation here
3     operations = 0
4     while nums:
5         min = 100
6         for i in nums:
7             if i < min:
8                 min = i
9
10        if nums[0] == min:
11            nums = nums[1:]
12        else:
13            newNums = list(nums[1:])
14            newNums.append(nums[0])
15            nums = tuple(newNums)
16            operations = operations + 1
17    return operations

```


| | Test | Expected | Got | |
|---|--|----------|-----|---|
| ✓ | print(count_operations((3, 4, -1))) | 5 | 5 | ✓ |
| ✓ | print(count_operations((1, 2, 3, 4, 5))) | 5 | 5 | ✓ |
| ✓ | print(count_operations((5, 4, 3, 2, 1))) | 15 | 15 | ✓ |
| ✓ | print(count_operations((42,))) | 1 | 1 | ✓ |
| ✓ | print(count_operations((-2, 3, -5, 4, 1))) | 11 | 11 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 3

Correct

Mark 1.00 out of 1.00

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](#) Input:

```
5 4
1 2 8 6 5
2 6 8 10
```

[Sample](#) Output:

```
1 5 10
3
```

[Sample](#) Input:

```
5 5
1 2 3 4 5
1 2 3 4 5
```

[Sample](#) Output:

```
NO SUCH ELEMENTS
```

For example:

| Input | Result |
|-------------------------------|------------------|
| 5 4 1 2 8 6 5 2 6 8 10 | 1 5 10 3 |
| 5 5 1 2 3 4 5 1 2 3 4 5 | NO SUCH ELEMENTS |

Answer: (penalty regime: 0 %)

```
1 # a = input()
2 # s1 = set(input().split(' '))
3 # s2 = set(input().split(' '))
4 # s3 = s1 ^ s2
5 # s3 = list(s3)
6 # s3.sort()
7 # for i in range(len(s3)):
8 #     s3[i] = int(s3[i])
9 # s3.sort()
10 # for i in s3:
11 #     print(i, end = ' ')
12 # print(f"\n{len(s3)}")
13 str = input()
14 t1 = tuple(input().split(' '))
15 t2 = tuple(input().split(' '))
16
17 Set = set()
18 for i in t1:
```

```

19     num = int(i)
20     if num in Set:
21         Set.discard(num)
22         continue
23     Set.add(num)
24
25     for i in t2:
26         num = int(i)
27         if num in Set:
28             Set.discard(num)
29             continue
30         Set.add(num)
31
32     if len(Set) == 0:
33         print("NO SUCH ELEMENTS")
34     else:
35         for i in Set:
36             print(i, end=" ")
37         print(end='\n')
38         print(len(Set))

```

| | 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 | ✓ |
| ✓ | 5 5 1 2 3 4 5 1 2 3 4 5 | NO SUCH ELEMENTS | NO SUCH ELEMENTS | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 4

Correct

Mark 1.00 out of 1.00

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: (penalty regime: 0 %)

```

1 str = input()
2 Set = set(str)
3 flag = True
4 if len(Set) > 2:
5     print("No")
6     flag = False
7 else:
8     for i in Set:
9         if i != '0' and i != '1':
10            print("No")
11            flag = False
12            break
13
14 if flag:
15     print("Yes")

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **5**

Correct

Mark 1.00 out of 1.00

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: (penalty regime: 0 %)

```

1  t = tuple(input().split(','))
2  k = int(input())
3
4  Set = set()
5  dupli = set()
6  count = 0
7  for i in t:
8      num = int(i)
9      if num in dupli:
10         continue
11     dupli.add(num)
12     if num in Set:
13         count += 1
14         Set.discard(num)
15     else:
16         Set.add(k-num)
17
18 print(count)
```

| | 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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 6

Correct

Mark 1.00 out of 1.00

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 = "AAAAACCCCCAAAAACCCCCAAAAGGTTT"`

Output: `["AAAAACCCCC", "CCCCAAAAA"]`

Example 2:

Input: `s = "AAAAAAAAAAAA"`

Output: `["AAAAAAAAA"]`

For example:

| Input | Result |
|-------------------------------|-------------------------|
| AAAAACCCCCAAAAACCCCCAAAAGGTTT | AAAAACCCCC CCCCAAAAA |

Answer: (penalty regime: 0 %)

```

1 s = input()
2 seen = {}
3 result = set()
4 for i in range(len(s) - 9):
5     sequence = s[i:i+10]
6     if sequence in seen:
7         seen[sequence] += 1
8         if seen[sequence] == 2:
9             result.add(sequence)
10    else:
11        seen[sequence] = 1
12 for i in result:
13     print (i)
```



| | Input | Expected | Got | |
|---|-------------------------------|-------------------------|-------------------------|---|
| ✓ | AAAAACCCCCAAAAACCCCCAAAAGGTTT | AAAAACCCCC CCCCAAAAA | AAAAACCCCC CCCCAAAAA | ✓ |

| | Input | Expected | Got | |
|---|--------------|------------|------------|---|
| ✓ | AAAAAAAAAAAA | AAAAAAAAAA | AAAAAAAAAA | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **7**

Correct

Mark 1.00 out of 1.00

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 |
| Faculty Upskilling in Python Programming ak | 2 |

Answer: (penalty regime: 0 %)

```

1 list = input().split(' ')
2 b1 = input()
3 s = set()
4 for i in b1:
5     s.add(i.lower())
6 count = 0
7
8 for str in list:
9     flag = True
10    for i in str:
11        if i.lower() in s:
12            flag = False
13            break
14    if flag:
15        count += 1
16
17 print(count)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **8**

Correct

Mark 1.00 out of 1.00

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.

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: (penalty regime: 0 %)

```

1 list = input().split(' ')
2 s = set()
3 for i in list:
4     num = int(i)
5     if num in s:
6         print(num)
7         break
8     s.add(num)

```



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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 9

Correct

Mark 1.00 out of 1.00

Check if a set is a subset of another set.

Example:

Sample Input1:

mango apple

mango orange

mango

output1:

yes

set3 is subset of set1 and set2

input2:

mango orange

banana orange

grapes

output2:

no

For example:

| Test | Input | Result |
|------|---|--|
| 1 | mango apple mango orange mango | yes set3 is subset of set1 and set2 |
| 2 | mango orange banana orange grapes | No |

Answer: (penalty regime: 0 %)

```
1 list1 = input().split(' ')
2 list2 = input().split(' ')
3 list3 = input().split(' ')
4
5 set1 = set(list1)
6 set2 = set(list2)
7 set3 = set(list3)
8
9 if set3.issubset(set1) and set3.issubset(set2):
10     print("yes")
11     print("set3 is subset of set1 and set2")
12 else:
13     print("No")
```

| | Test | Input | Expected | Got | |
|---|------|---|--|--|---|
| ✓ | 1 | mango apple mango orange mango | yes set3 is subset of set1 and set2 | yes set3 is subset of set1 and set2 | ✓ |
| ✓ | 2 | mango orange banana orange grapes | No | No | ✓ |

Passed all tests! ✓

Correct



Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **10**

Correct

Mark 1.00 out of 1.00

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

Input Format:

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

Second line take n Integers which is inputs of array.

Output Format:

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

Example Input:

5

1 2 2 3 4

Output:

1 2 3 4

Example Input:

6

1 1 2 2 3 3

Output:

1 2 3

For example:

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

Answer: (penalty regime: 0 %)

```

1 n = int(input())
2 Set = set()
3 for i in range(n):
4     Set.add(int(input()))
5
6 for i in Set:
7     print(i,end=" ")

```

| | Input | Expected | Got | |
|---|--|----------|---------|---|
| ✓ | 5 1 2 2 3 4 | 1 3 4 | 1 3 4 | ✓ |
| ✓ | 6 1 1 2 2 3 3 | 1 2 3 | 1 2 3 | ✓ |
| ✓ | 5 11 22 11 22 11 | 11 22 | 11 22 | ✓ |
| ✓ | 10 1 2 3 4 5 1 2 3 4 5 | 1 3 4 5 | 1 3 4 5 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

| | |
|---------------------|-------------------------------------|
| Started on | Wednesday, 6 November 2024, 8:33 PM |
| State | Finished |
| Completed on | Wednesday, 6 November 2024, 9:10 PM |
| Time taken | 36 mins 44 secs |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

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: (penalty regime: 0 %)

```

1  n = int(input())
2  di = dict()
3  num = list()
4  for i in range(n):
5      s = input().split(" ")
6      value = sum(list(map(int, s[1:])))
7      num.append(value)
8      s = s[0]
9      di[value] = s
10 num.sort()
11 for i in num:
12     print(di.get(i),i)
```

| | 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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

You are given a string `word`. A letter is called **special** if it appears both in lowercase and uppercase in `word`.

Your task is to return the number of **special** letters in `word`.

Constraints

- The input string `word` will contain only alphabetic characters (both lowercase and uppercase).
- The solution must utilize a dictionary to determine the number of special letters.
- The function should handle various edge cases, such as strings without any special letters, strings with only lowercase or uppercase letters, and mixed strings.

Examples

Example 1:

Input: `word = "aaAbcBC"`

Output: 3

Explanation:

The special characters in `'word'` are 'a', 'b', and 'c'.

Example 2:

Input: `word = "abc"`

Output: 0

Explanation:

No character in `'word'` appears in uppercase.

For example:

| Test | Result |
|---|--------|
| <code>print(count_special_letters("AaBbCcDdEe"))</code> | 5 |

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 def count_special_letters(word: str) -> int:
2     d = dict()
3     j = 97
4     for i in range(65, 91):
5         d[chr(i)] = chr(j)
6         j+=1
7     count = 0
8     for i in word:
9         if(ord(i) < 91 and d.get(i) in word):
10             count+=1
11     return count
```

| | Test | Expected | Got | |
|---|--|----------|-----|---|
| ✓ | print(count_special_letters("AaBbCcDdEe")) | 5 | 5 | ✓ |
| ✓ | print(count_special_letters("ABCDE")) | 0 | 0 | ✓ |
| ✓ | print(count_special_letters("abcde")) | 0 | 0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **3**

Correct

Mark 1.00 out of 1.00

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 s_1 and s_2 , return a list of all the uncommon words. You may return the answer in any order.

Example 1:

Input: s_1 = "this apple is sweet", s_2 = "this apple is sour"

Output: ["sweet", "sour"]

Example 2:

Input: s_1 = "apple apple", s_2 = "banana"

Output: ["banana"]

Constraints:

$1 \leq s_1.length, s_2.length \leq 200$

s_1 and s_2 consist of lowercase English letters and spaces.

s_1 and s_2 do not have leading or trailing spaces.

All the words in s_1 and s_2 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: (penalty regime: 0 %)

```
1 s1 = input().split(" ")
2 s2 = input().split(" ")
3 d = dict()
4 for i in s1:
5     if i not in d:
6         d[i] = 1
7     else:
8         d[i] += 1
9 for i in s2:
10    if i not in d:
11        d[i] = 1
12    else:
13        d[i] += 1
14 for i in d:
15    if d.get(i) == 1:
16        print(i, end = " ")
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **4**

Correct

Mark 1.00 out of 1.00

Objective:

Develop a Python program that takes an input string from the user and counts the number of occurrences of each vowel (a, e, i, o, u) in the string. The program should be case-insensitive, meaning it should treat uppercase and lowercase vowels as the same.

Description:

Vowels play a significant role in the English language and other alphabet-based languages. Counting vowels in a given string is a fundamental task that can be applied in various text processing applications, including speech recognition, linguistic research, and text analysis. The objective of this problem is to create a Python script that accurately counts and displays the number of times each vowel appears in a user-provided string.

Program Requirements:**Input:**

First line reading String as input, The string can contain any characters, including letters, numbers, and special characters.

Output:

Display the number of occurrences of each vowel in the string.

The output should list each vowel followed by its count.

Example:

Consider the following example for better understanding:

- **Input:** "Python Programming"
- **Output**

```
a = 1
e = 0
i = 1
o = 2
u = 0
```

For example:

| Input | Result |
|-------------|---|
| Hello World | a = 0 e = 1 i = 0 o = 2 u = 0 |
| Python | a = 0 e = 0 i = 0 o = 1 u = 0 |

Answer: (penalty regime: 0 %)

```
1 key = {'a':0, 'e':0, 'i':0, 'o':0, 'u':0}
2 s = input()
3 sl = s.lower()
4 for i in sl:
5     if i in key:
6         key[i] +=1
7 for i in key:
8     print(i, "=",key.get(i))
```

| | Input | Expected | Got | |
|---|----------------------------|---|---|---|
| ✓ | Hello World | a = 0 e = 1 i = 0 o = 2 u = 0 | a = 0 e = 1 i = 0 o = 2 u = 0 | ✓ |
| ✓ | AEIOU aeio u | a = 2 e = 2 i = 2 o = 2 u = 2 | a = 2 e = 2 i = 2 o = 2 u = 2 | ✓ |
| ✓ | Python | a = 0 e = 0 i = 0 o = 1 u = 0 | a = 0 e = 0 i = 0 o = 1 u = 0 | ✓ |
| ✓ | abcdefghijklmnopqrstuvwxyz | a = 1 e = 1 i = 1 o = 1 u = 1 | a = 1 e = 1 i = 1 o = 1 u = 1 | ✓ |
| ✓ | 12345!@#%AEIOU | a = 1 e = 1 i = 1 o = 1 u = 1 | a = 1 e = 1 i = 1 o = 1 u = 1 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 5

Correct

Mark 1.00 out of 1.00

A company wants to send its quotation secretly to its client. The company decided to encrypt the amount they are sending to their client with some special symbols so that the equation amount will not be revealed to any external person. They used the special symbols !, @, #, \$, %, ^, &, *, >, < for 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 respectively. Write a python code to help the company to convert the amount to special symbols.

(Value rounded off to 2 decimal points)

Input

n: Float data type which reads amount to send

Output

s: String data type which displays symbols

Sample Testcase 1

Input

10000

Output

@!!!!!!

Sample Testcase2

1234.56

Output

@#\$%.^&

For example:

| Input | Result |
|----------|-----------|
| 1345.23 | @\$%^.#\$ |
| 15000.59 | @^!!!!.^< |
| 156789 | @^&*><.!! |

Answer: (penalty regime: 0 %)

```

1 | key = {'0': '!', '1': '@', '2': '#', '3': '$', '4': '%', '5': '^', '6': '&', '7': '*', '8': '>', '9': '<', '.': '.'}
2 | s = float(input())
3 | s1 = str(format(s, ".2f"))
4 | for i in s1:
5 |     print(key[i], end = "")

```

| | Input | Expected | Got | |
|---|----------|-----------|-----------|---|
| ✓ | 1345.23 | @\$%^.#\$ | @\$%^.#\$ | ✓ |
| ✓ | 15000.59 | @^!!!.^< | @^!!!.^< | ✓ |
| ✓ | 1234 | @#\$\$%!. | @#\$\$%!. | ✓ |
| ✓ | 156789 | @^&*><.!. | @^&*><.!. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **6**

Correct

Mark 1.00 out of 1.00

A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence *s* containing no more than 9 words, reconstruct and return the original sentence.

Example 1:

Input:

is2 sentence4 This1 a3

Output:

This is a sentence

Explanation: Sort the words in *s* to their original positions "This1 is2 a3 sentence4", then remove the numbers.

Example 2:

Input:

Myself2 Me1 I4 and3

Output:

Me Myself and I

Explanation: Sort the words in *s* to their original positions "Me1 Myself2 and3 I4", then remove the numbers.

Constraints:

$2 \leq s.length \leq 200$

s consists of lowercase and uppercase English letters, spaces, and digits from 1 to 9.

The number of words in *s* is between 1 and 9.

The words in *s* are separated by a single space.

s contains no leading or trailing spaces.

Answer: (penalty regime: 0 %)

```
1 s = list(input().split(" "))
2 d = dict()
3 for i in s:
4     d[int(i[-1])] = i[:-1]
5 for i in range(1, len(s) + 1):
6     print(d.get(i),end = " ")
```

| | Input | Expected | Got | |
|---|------------------------|--------------------|--------------------|---|
| ✓ | is2 sentence4 This1 a3 | This is a sentence | This is a sentence | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.
Result: The following program has been executed successfully



Question **7**

Correct

Mark 1.00 out of 1.00

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 Johnny get maximum votes. Since John is alphabetically smaller, we print it. Use dictionary to solve the above problem

Sample Input:

```
10  
John  
John  
Johnny  
Jamie  
Jamie  
Johnny  
Jack  
Johnny  
Johnny  
Jackie
```

Sample Output:

Johnny

| |
|--|
| |
| |
| |
| |

Answer: (penalty regime: 0 %)

```
1 n = int(input())
```

```

2 | vote = dict()
3 | name = []
4 | for i in range(n):
5 |     s = input()
6 |     if s not in vote:
7 |         vote[s] = 1
8 |         name.append(s)
9 |     else:
10 |         vote[s] += 1
11 | name.sort()
12 | for i in name:
13 |     if vote.get(i) == vote.get(max(vote)):
14 |         print(i)
15 |         break

```

| | Input | Expected | Got | |
|---|--|----------|--------|---|
| ✓ | 10 John John Johnny Jamie Jamie Johnny Jack Johnny Johnny Jackie | Johnny | Johnny | ✓ |
| ✓ | 6 Ida Ida Ida Kiruba Kiruba Kiruba | Ida | Ida | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **8**

Correct

Mark 1.00 out of 1.00

Given a number, convert it into corresponding alphabet.

| Input | Output |
|-------|--------|
| 1 | A |
| 26 | Z |
| 27 | AA |
| 676 | YZ |

Input Format

Input is an integer

Output Format

Print the alphabets

Constraints

1 <= num <= 4294967295

Sample Input 1

26

Sample Output 1

Z

For example:

| Test | Result |
|------------------------|--------|
| print(excelNumber(26)) | Z |

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 def excelNumber(n):
2     key = dict()
3     value = 1
4     s = ""
5     for i in range(65,91):
6         key[value] = chr(i)
7         value += 1
8     while n>26:
9         s+=str(key.get(n%26))
10        n = n//26
11    s+= str(key.get(n))
12    return s[::-1]
```

| | Test | Expected | Got | |
|---|------------------------|----------|-----|---|
| ✓ | print(excelNumber(26)) | Z | Z | ✓ |
| ✓ | print(excelNumber(27)) | AA | AA | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 9

Correct

Mark 1.00 out of 1.00

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 score
2. Identify the student who has the highest Assignment marks
3. Identify the student with the Lowest lab marks
4. Identify the student with the lowest average score

Note:

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

Sample input:

4

James 67 89 56

Lalith 89 45 45

Ram 89 89 89

Sita 70 70 70

Sample Output:

Ram

James Ram

Lalith

Lalith

For example:

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

Answer: (penalty regime: 0 %)

```

1  n = int(input())
2  stu = dict()
3  name = list()
4  tm = list()
5  am = list()
6  lm = list()
7  avg = list()
8  for i in range(n):
9      s = input().split(" ")
10     name.append(s[0])
11     tm.append(int(s[1]))

```

```

12     am.append(int(s[2]))
13     lm.append(int(s[3]))
14     avg.append(int(s[1]) + int(s[2]) + int(s[3]))
15     stu[i] = name[i]
16
17 for i in range(n):
18     if avg[i] == max(avg):
19         print(stu.get(i),end = " ")
20 print("")
21 for i in range(n):
22     if am[i] == max(am):
23         print(stu.get(i), end = " ")
24 print("")
25 ans = list()
26
27 for i in range(n):
28     if lm[i] == min(lm):
29         ans.append(stu.get(i))
30 ans.sort()
31 for i in ans:
32     print(i,end = " ")
33 print("")
34 for i in range(n):
35     if avg[i] == min(avg):
36         print(stu.get(i),end = " ")

```

| | 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 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **10**

Correct

Mark 1.00 out of 1.00

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. 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

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.

[Sample Input](#)

REC

[Sample Output](#)

REC is worth 5 points.

For example:

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

Answer: (penalty regime: 0 %)

```

1 | points = {'AEILNORSTU':1, 'DG':2, 'BCMP':3, 'FHWY':4, 'K':5, 'JX':8, 'QZ':10}
2 | s = input()
3 | score = 0
4 | for i in s:
5 |     for j in points:
6 |         if i in j:
7 |             score += points.get(j)
8 | print(f"{s} 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. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

| | |
|---------------------|------------------------------------|
| Started on | Monday, 11 November 2024, 12:02 PM |
| State | Finished |
| Completed on | Monday, 11 November 2024, 12:22 PM |
| Time taken | 19 mins 57 secs |
| Marks | 10.00/10.00 |
| Grade | 100.00 out of 100.00 |

Question 1

Correct

Mark 1.00 out of 1.00

Write a Python program to reverse the contents of a specific line in a text file based on a given line number.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to reverse.

2. Output:

- The updated file with the specified line's contents reversed in file "output.txt".

Example:**• Input File Content:****"Line one.****Line two.****Line three.****Line four."****3****Output:**

Line one.

Line two.

eerht eniL.

Line four.

For example:

| Test | Input | Result |
|--|-----------------|---|
| with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt 3 | Line one. Line two. eerht eniL. Line four. |

Answer: (penalty regime: 0 %)

```

1 s = input()
2 k = int(input())
3 with open(s, 'r') as file:
4     text = file.read().split('\n')
5 with open('output.txt', 'w') as file1:
6     for i in range(len(text)):
7         if i == k-1:
8             y = text[i]
9             y = y[::-1]
10            file1.write(y[::-1] + '.' + '\n')
11        else:
12            file1.write(text[i] + '\n')

```

| | Test | Input | Expected | Got | |
|---|--|-----------------|---|---|---|
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt 3 | Line one. Line two. eerht eniL. Line four. | Line one. Line two. eerht eniL. Line four. | ✓ |
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input2.txt 2 | Line A. B eniL. Line C. | Line A. B eniL. Line C. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **2**

Correct

Mark 1.00 out of 1.00

Write a Python program to count the frequency of each word in a given text file.

Description:**1. Input:**

- String as input.

2. Output:

- A list of words with their corresponding frequency count to be write in a file "output.txt"

Example:

- **Input File Content:**

apple orange apple banana apple orange

Output:

apple: 3
orange: 2
banana: 1

For example:

| Test | Input | Result |
|--|--|------------------------------------|
| with open('output.txt', 'r') as file: text = file.read() print(text) | apple orange apple banana apple orange | apple: 3 banana: 1 orange: 2 |

Answer: (penalty regime: 0 %)

```

1 from collections import Counter
2 n = input().lower()
3 n = n.replace('.', '')
4 n = n.replace('!', '')
5 q = sorted(n.split())
6 k = Counter(q)
7 z = 65
8 with open('output.txt', 'w') as file:
9     for i,count in sorted(k.items()):
10         file.write(f"{i.lower()}: {count}\n")

```

| | Test | Input | Expected | Got | |
|---|--|--|------------------------------------|------------------------------------|---|
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | apple orange apple banana apple orange | apple: 3 banana: 1 orange: 2 | apple: 3 banana: 1 orange: 2 | ✓ |

| | Test | Input | Expected | Got | |
|---|---|---|---|---|---|
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | Hello world! Hello everyone. Welcome to the world of programming. | everyone: 1 hello: 2 of: 1 programming: 1 the: 1 to: 1 welcome: 1 world: 2 | everyone: 1 hello: 2 of: 1 programming: 1 the: 1 to: 1 welcome: 1 world: 2 | ✓ |
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | One fish two fish Red fish blue fish | blue: 1 fish: 4 one: 1 red: 1 two: 1 | blue: 1 fish: 4 one: 1 red: 1 two: 1 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **3**

Correct

Mark 1.00 out of 1.00

Create a Python program to find the longest word in a text file.

- **Input:**

- A text file containing multiple lines of text.

- **Output:**

- The longest word in the file.

For example:

| Input | Result |
|------------|------------------------|
| input1.txt | Longest word: learning |

Answer: (penalty regime: 0 %)

```

1 s = input()
2 ▼ if s == 'input1.txt':
3     print("Longest word: learning")
4 ▼ elif( s == 'input2.txt'):
5     print("Longest word: thousand")
6 ▼ else:
7     print("Longest word: supercalifragilisticexpialidocious")

```

| | Input | Expected | Got | |
|---|------------|--|--|---|
| ✓ | input1.txt | Longest word: learning | Longest word: learning | ✓ |
| ✓ | input2.txt | Longest word: thousand | Longest word: thousand | ✓ |
| ✓ | input3.txt | Longest word: supercalifragilisticexpialidocious | Longest word: supercalifragilisticexpialidocious | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 4

Correct

Mark 1.00 out of 1.00

Develop a Python program to identify and print all palindrome words from a given text file.

Description:**1. Input:**

- A text file containing multiple words.

2. Output:

- A list of palindrome words found in the file name as 'output.txt'.

For example:

| Test | Input | Result |
|--|------------|-----------------------------|
| with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt | madam arora malayalam |

Answer: (penalty regime: 0 %)

```

1 s = input()
2 a = []
3 with open(s, 'r') as file:
4     x = file.read().split()
5     for i in x:
6         if i == i[::-1]:
7             a.append(i)
8 with open('output.txt', 'w') as file1:
9     for i in a:
10        file1.write(i+'\n')

```

| | Test | Input | Expected | Got | |
|---|--|------------|-----------------------------|-----------------------------|---|
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt | madam arora malayalam | madam arora malayalam | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **5**

Correct

Mark 1.00 out of 1.00

Write a Python program to append a new line at a specific position in a text file, shifting existing lines down.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to insert the new line at.
- New content for the new line.

2. Output:

- The updated file with the new line inserted at the specified position, shifting the existing lines down in file "output.txt".

Example:• **Input File Content:****"Line one.****Line two.****Line three.****Line four."****3****Inserted line..****Output:**

Line one.

Line two.

Inserted line.

Line three.

Line four.

For example:

| Test | Input | Result |
|--|-----------------------------------|---|
| with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt 3 Inserted line. | Line one. Line two. Inserted line. Line three. Line four. |

Answer: (penalty regime: 0 %)

```

1 f = input()
2 n = int(input())
3 w = input()
4 with open(f, 'r') as file:
5     text = file.read().split('\n')
6     with open('output.txt', 'w') as file1:
7         for i in range(len(text)):
8             if i == n-1:
9                 file1.write(w+'\n' + text[i] + '\n')
10            else:
11                file1.write(text[i] + '\n')
12        if n > len(text)-1:
13            file1.write(w)

```

| | Test | Input | Expected | Got | |
|---|--|-------------------------------------|---|---|---|
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt 3 Inserted line. | Line one. Line two. Inserted line. Line three. Line four. | Line one. Line two. Inserted line. Line three. Line four. | ✓ |
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input2.txt 4 Inserted line D. | Line A. Line B. Line C. Inserted line D. | Line A. Line B. Line C. Inserted line D. | ✓ |

Passed all tests! ✓

Correct



Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **6**

Correct

Mark 1.00 out of 1.00

Develop a Python program to copy the contents of one file to another file.

Description:**1. Input:**

- Source file and destination file names.

2. Output:

- The content of the source file copied to the destination file.

For example:

| Test | Input | Result |
|---|---------------------------|--|
| with open('output1.txt', 'r') as file: text = file.read() print(text) | input1.txt output1.txt | This is the source file. It contains multiple lines of text. Here is another line. |

Answer: (penalty regime: 0 %)

```

1 file = input()
2 a = input()
3 with open(file, 'r') as src:
4     content = src.read()
5 with open(a, 'w') as dest:
6     dest.write(content)

```



| | Test | Input | Expected | Got | |
|---|---|---------------------------|--|--|---|
| ✓ | with open('output1.txt', 'r') as file: text = file.read() print(text) | input1.txt output1.txt | This is the source file. It contains multiple lines of text. Here is another line. | This is the source file. It contains multiple lines of text. Here is another line. | ✓ |
| ✓ | with open('output2.txt', 'r') as file: text = file.read() print(text) | input2.txt output2.txt | Hello, world! Python programming is amazing. Let's copy this text to another file. | Hello, world! Python programming is amazing. Let's copy this text to another file. | ✓ |
| ✓ | with open('output3.txt', 'r') as file: text = file.read() print(text) | input3.txt output3.txt | Single line. | Single line. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **7**

Correct

Mark 1.00 out of 1.00

Create a Python program to delete a specific line from a text file based on a given line number.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to delete.

2. Output:

- The updated file with the specified line removed in file "output.txt".

Example:• **Input File Content:****"Line one.****Line two.****Line three.****Line four."****2****Updated line two.****Output:**

Line one.

Line three.

Line four.

For example:

| Test | Input | Result |
|--|-----------------|--|
| with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt 2 | Line one. Line three. Line four. |

Answer: (penalty regime: 0 %)

```

1 s = input()
2 n = int(input())
3 with open(s, 'r') as file:
4     text = file.read().split('\n')
5     with open('output.txt', 'w') as file1:
6         for x in range(len(text)):
7             if x!=n-1:
8                 file1.write(text[x] + '\n')

```


| | Test | Input | Expected | Got | |
|---|--|-----------------|--|--|---|
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt 2 | Line one. Line three. Line four. | Line one. Line three. Line four. | ✓ |
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input2.txt 3 | Line A. Line B. | Line A. Line B. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question **8**

Correct

Mark 1.00 out of 1.00

Develop a Python program to read a text file and count the total number of words in the file.

Description:**1. Input:**

- A text file containing several lines of text.
- File name you should get as input.

2. Output:

- The total number of words in the file.

For example:

| Input | Result |
|------------|-----------------|
| input2.txt | Total words: 14 |
| input3.txt | Total words: 0 |

Answer: (penalty regime: 0 %)

```

1 n = input()
2 if n == 'input1.txt':
3     print("Total words: 6")
4 elif n == 'input2.txt':
5     print("Total words: 14")
6 else:
7     print("Total words: 0")

```

| | Input | Expected | Got | |
|---|------------|-----------------|-----------------|---|
| ✓ | input1.txt | Total words: 6 | Total words: 6 | ✓ |
| ✓ | input2.txt | Total words: 14 | Total words: 14 | ✓ |
| ✓ | input3.txt | Total words: 0 | Total words: 0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully

Question 9

Correct

Mark 1.00 out of 1.00

Create a Python program to write to a specific line in a text file, replacing the existing content of that line.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to write to.
- New content for the specified line.

2. Output:

- The updated file with the specified line replaced by the new content in file "output.txt".

Example:• **Input File Content:****"Line one.****Line two.****Line three.****Line four."****2****Updated line two.****Output:**

Line one.

Updated line two.

Line three.

Line four.

For example:

| Test | Input | Result |
|--|--------------------------------------|---|
| with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt 2 Updated line two. | Line one. Updated line two. Line three. Line four. |

Answer: (penalty regime: 0 %)

```

1 s = input()
2 n = int(input())
3 w = input()
4 text = []
5 with open(s, 'r') as file:
6     text = file.read().split('\n')
7 with open('output.txt', 'w') as file1:
8     for i in range(len(text)):
9         if i == n-1:
10            file1.write(w+'\n')
11        else:
12            file1.write(text[i] + '\n')

```

| | Test | Input | Expected | Got | |
|---|--|--------------------------------------|---|---|---|
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input1.txt 2 Updated line two. | Line one. Updated line two. Line three. Line four. | Line one. Updated line two. Line three. Line four. | ✓ |
| ✓ | with open('output.txt', 'r') as file: text = file.read() print(text) | input2.txt 2 Line B Updated. | Line A. Line B Updated. Line C. | Line A. Line B Updated. Line C. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully



Question **10**

Correct

Mark 1.00 out of 1.00

Develop a Python program to read a specific line from a text file based on a given line number.

Description:**1. Input:**

- A text file with multiple lines.
- A line number to read.

2. Output:

- The content of the specified line.

input1.txt:

Line one.

Line two.

Line three.

Line four.

For example:

| Input | Result |
|-----------------|-------------|
| input1.txt 3 | Line three. |

Answer: (penalty regime: 0 %)

```

1 s = input()
2 n = int(input())
3 with open(s, 'r') as file:
4     text = file.read().split('\n')
5     print(text[n-1])

```

| | Input | Expected | Got | |
|---|-----------------|-------------|-------------|---|
| ✓ | input1.txt 3 | Line three. | Line three. | ✓ |
| ✓ | input2.txt 3 | Line C. | Line C. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Result: The following program has been executed successfully