

**Started on** Tuesday, 29 July 2025, 1:05 PM

**State** Finished

**Completed on** Tuesday, 29 July 2025, 1:17 PM

**Time taken** 11 mins 32 secs

**Question 1** | Correct | Marked 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. print(a and "D" or "C")
```

|   | <b>Input</b> | <b>Expected</b> | <b>Got</b> |   |
|---|--------------|-----------------|------------|---|
| ✓ | 8            | C               | C          | ✓ |
| ✓ | 1            | D               | D          | ✓ |

Passed all tests! ✓

**Question 2** | Correct | Marked 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: The 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 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

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

**For example:**

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

**Answer:** (penalty regime: 0 %)

```
1 a=int(input())
2 b=int(input())
3 c=abs(a)%10
4 d=abs(b)%10
5 print(c+d)
```

|   | Input       | Expected | Got |   |
|---|-------------|----------|-----|---|
| ✓ | 267<br>154  | 11       | 11  | ✓ |
| ✓ | 267<br>-154 | 11       | 11  | ✓ |

Passed all tests! ✓

**Question 3** | Correct | Marked 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.

**For example:**

| Input | Result          |
|-------|-----------------|
| 340   | Boat is stable  |
| 2     |                 |
| 3     |                 |
| 600   | Boat will drown |
| 7     |                 |
| 4     |                 |

**Answer:** (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 c=int(input())
4 d=(b*75)+(c*30)
5 if(d<a):
6     print("Boat is stable")
7 else:
8     print("Boat will drown")

```

|   | Input | Expected       | Got            |   |
|---|-------|----------------|----------------|---|
| ✓ | 340   | Boat is stable | Boat is stable | ✓ |
|   | 2     |                |                |   |
|   | 3     |                |                |   |

|   | Input | Expected       | Got            |   |
|---|-------|----------------|----------------|---|
| ✓ | 600   | Boat will draw | Boat will draw | ✓ |
|   | 7     |                |                |   |
|   | 4     |                |                |   |

Passed all tests! ✓

**Question 4** | Correct | Marked 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 count=0
3 while a:
4     if a&1:
5         count+=1
6     a>>=1
7 print(count)

```

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

Passed all tests! ✓

**Question 5** | Correct | Marked 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    | False  |
| 43    |        |

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=int(input())
3 print(a%3==0 and b%2==0)
```

|   | Input         | Expected | Got   |   |
|---|---------------|----------|-------|---|
| ✓ | 32<br>43      | False    | False | ✓ |
| ✓ | 273<br>7898   | True     | True  | ✓ |
| ✓ | 888<br>4598   | False    | False | ✓ |
| ✓ | 6789<br>32996 | True     | True  | ✓ |

Passed all tests! ✓

**Question 6** | Correct | Marked 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 a=int(input())
2 b=0
3 c=0
4 while(True):
5     if(a>=365):
6         b=b+1
7         a=a-365
8     elif(a>=30):
9         c=c+1
10        a=a-30
11    else:
12        break
13 print("YEARS:",b,"MONTH:",c,"DAYS:",a)

```

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

Passed all tests! ✓

**Question 7** | Correct | Marked out of 1.00

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

Sample Input:

8.00

Sample Output:

2.828

For example:

| Input | Result |
|-------|--------|
| 8.00  | 2.828  |
| 14.00 | 3.742  |

Answer: (penalty regime: 0 %)

```
1 import math
2 a=float(input())
3 b=math.sqrt(a)
4 print("%.3f"%b)
```

|   | Input | Expected | Got   |   |
|---|-------|----------|-------|---|
| ✓ | 8.00  | 2.828    | 2.828 | ✓ |
| ✓ | 14.00 | 3.742    | 3.742 | ✓ |

Passed all tests! ✓

**Question 8** · Correct · Marked 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 a=int(input())
2 b=0.4*a
3 c=0.2*a
4 d=int(a+b+c)
5 print(d)

```

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

Passed all tests! ✓

**Question 9** | Correct | Marked 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 c=a*75
4 d=b*112
5 print("The total weight of all these widgets and gizmos is",c+d,"grams.")
```

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

Passed all tests! ✓

**Question 10** | Correct Marked 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    | False  |
| 40    |        |

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 b=int(input())
3 print(a>=18 and b>40)
```

|   | <b>Input</b> | <b>Expected</b> | <b>Got</b> |   |
|---|--------------|-----------------|------------|---|
| ✓ | 19<br>45     | True            | True       | ✓ |
| ✓ | 18<br>48     | False           | False      | ✓ |
| ✓ | 18<br>42     | True            | True       | ✓ |
| ✓ | 16<br>45     | False           | False      | ✓ |

Passed all tests! ✓

**Started on** Sunday, 17 August 2025, 7:45 PM

**State** Finished

**Completed on** Sunday, 17 August 2025, 8:05 PM

**Time taken** 19 mins 58 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 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      |
| 1801  | 1      |

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 result=2-(n%2)
3 print("%d"%result)
```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 100   | 2        | 2   | ✓ |
| ✓ | 1801  | 1        | 1   | ✓ |
| ✓ | 0     | 2        | 2   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 2** | Correct | Mark 1.00 out of 1.00

In a Logistic the Parcels to be delivered in 4 locations (1st location 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

**For example:**

| Input | Result   |
|-------|--|
| 250   | Total Parcels is 250<br>1st Location 50 parcels<br>2nd Location 100 parcels<br>3rd Location 75 parcels<br>4th Location 25 parcels  |
| 400   | Total Parcels is 400<br>1st Location 80 parcels<br>2nd Location 160 parcels<br>3rd Location 120 parcels<br>4th Location 40 parcels |

**Answer:** (penalty regime: 0 %)

```

1 p=int(input())
2 a=p*20//100
3 b=p*40//100
4 c=p*30//100
5 d=p-(a+b+c)
6 print("Total Parcels is {}".format(p))
7 print("1st Location {} parcels".format(a))
8 print("2nd Location {} parcels".format(b))
9 print("3rd Location {} parcels".format(c))
10 print("4th Location {} parcels".format(d))

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | 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    | Your total refund will be \$7.00. |
| 20    |                                   |

**Answer:** (penalty regime: 0 %)

```

1. a=int(input())
2. b=int(input())
3. refund=a*0.10+b*0.25
4. print("Your total refund will be ${:.2f}."%refund)

```

|   | Input    | Expected                          | Got                               |   |
|---|----------|-----------------------------------|-----------------------------------|---|
| ✓ | 20<br>20 | Your total refund will be \$7.00. | Your total refund will be \$7.00. | ✓ |
| ✓ | 11<br>22 | Your total refund will be \$6.60. | Your total refund will be \$6.60. | ✓ |

|   | Input      | Expected                           | Got                                |   |
|---|------------|------------------------------------|------------------------------------|---|
| ✓ | 123<br>288 | Your total refund will be \$62.38. | Your total refund will be \$62.38. | ✓ |
| ✓ | 76<br>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.

**Question 4** | 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<br>weekend 0.38 |

**Answer:** (penalty regime: 0 %)

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

|   | Input | Expected                        | Got                               |  |
|---|-------|---------------------------------|-----------------------------------|--|
| ✓ | 458   | weekdays 10.38<br>weekend 0.38  | weekdays 10.38 ✓<br>weekend 0.38  |  |
| ✓ | 500   | weekdays 10.00<br>weekend 0.00  | weekdays 10.00 ✓<br>weekend 0.00  |  |
| ✓ | 10000 | weekdays 83.08<br>weekend 73.08 | weekdays 83.08 ✓<br>weekend 73.08 |  |
| ✓ | 6789  | weekdays 58.38<br>weekend 48.38 | weekdays 58.38 ✓<br>weekend 48.38 |  |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 5** | 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 n=int(input())
2 n=abs(n)%10
3 print(n)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 6** | 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 n=int(input())
2 n=abs(n)
3 print(((n//10)%10)*(n>9)+(-1)*(n<10))
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



**Question 7** | 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 | 38.43 is the gain percent. |
| 500   |                            |
| 60000 |                            |

**Answer:** (penalty regime: 0 %)

```

1 x=float(input())
2 y=float(input())
3 z=float(input())
4 total=x+y
5 gain=z-(x+y)
6 gain_percent=(gain/total)*100
7 print("%.2f is the gain percent."%gain_percent)

```

|   | Input                  | Expected                   | Got                        |   |
|---|------------------------|----------------------------|----------------------------|---|
| ✓ | 10000<br>250<br>15000  | 46.34 is the gain percent. | 46.34 is the gain percent. | ✓ |
| ✓ | 45500<br>500<br>60000  | 30.43 is the gain percent. | 30.43 is the gain percent. | ✓ |
| ✓ | 5000<br>0<br>7000      | 40.00 is the gain percent. | 40.00 is the gain percent. | ✓ |
| ✓ | 12500<br>5000<br>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.

**Question 8** | Correct | Mark 1.00 out of 1.00

In a Lab 36% are Dell and 34% Lennovo 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

Lennovo:51

Acer:42

Samsung:3

**For example:**

| Input | Result  |
|-------|---|
| 150   | Total System:150<br>Dell:54<br>Lennovo:51<br>Acer:42<br>Samsung:3 |
| 250   | Total System:250<br>Dell:90<br>Lennovo:85<br>Acer:78<br>Samsung:5 |

**Answer:** (penalty regime: 0 %)

```

1 total=int(input())
2 dell=total*36//100
3 lenovo=total*34//100
4 acer=total*28//100
5 samsung=total-(dell+lenovo+acer)
6 print("Total System:%d"%total)
7 print("Dell:%d"%dell)
8 print("Lenovo:%d"%lenovo)
9 print("Acer:%d"%acer)
10 print("Samsung:%d"%samsung)

```

|   | Input | Expected   | Got  |   |
|---|-------|--|--|---|
| ✓ | 158   | Total System:158<br>Dell:54<br>Lenovo:51<br>Acer:42<br>Samsung:3 | Total System:150<br>Dell:54<br>Lenovo:51<br>Acer:42<br>Samsung:3 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 9** | 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 a=input()
2 b=input()
3 x=int(a)
4 y=float(b)
5 print("%d,%s"%(x,type(x)))
6 print("%.1f,%s"%(y,type(y)))

```

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

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 10** | 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

**For example:**

| Input | Result   |
|-------|--|
| 1500  | Total Students : 1500, Boys : 810, Girls : 690, Hostel : 120 |
| 1200  | Total Students : 1200, Boys : 648, Girls : 552, Hostel : 96  |

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 boys=n*54//100
3 girls=n*46//100
4 hostel=n*8//100
5 print("Total Students : %d, Boys : %d, Girls : %d, Hostel : %d"%(n,boys,girls,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

**Started on** Sunday, 17 August 2025, 4:57 PM

**State** Finished

**Completed on** Sunday, 17 August 2025, 8:55 PM

**Time taken** 3 hours 57 mins

**Question 1** | Correct | Marked out of 1.00

A certain type of steel is used to test and give grade according to the following conditions.

1. Hardness of the steel must be greater than 50
2. Carbon content of the steel must be less than 0.7
3. Tensile strength must be greater than 5600

The grades awarded are as follows:

- Grade is 10 if all three conditions are met
- Grade is 9 if conditions (1) and (2) are met
- Grade is 8 if conditions (2) and (3) are met
- Grade is 7 if conditions (1) and (3) are met
- Grade is 6 if only one condition is met
- Grade is 5 if none of the three conditions are met

Write a program to display the grade of the steel, based on the values of hardness, carbon content and tensile strength of the steel, given by the user.

Input

53

0.6

5602

Output:

10

**Answer:** (penalty regime: 0 %)

```
1 a=int(input())
2 b=float(input())
3 c=int(input())
4
5 if a>50 and b<0.7 and c>5600:
6     print("10")
7 elif a>50 and b<0.7 and c<5600:
8     print("9")
9     pass
10 elif a<50 and b<0.7 and c>5600:
11     print("8")
12     pass
13 elif a>50 and b>0.7 and c>5600:
14     print("7")
15 elif a>50 and b>0.7 and c<5600:
16     print("6")
17 elif a<50 and b<0.7 and c<5600:
18     print("6")
19
```

|   | Input             | Expected | Got |   |
|---|-------------------|----------|-----|---|
| ✓ | 53<br>0.6<br>5682 | 10       | 10  | ✓ |
| ✓ | 45<br>8<br>4588   | 6        | 6   | ✓ |

Passed all tests! ✓

**Question 2** Correct Marked 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 b=input()
3 c=int(input())
4 if b=='*':
5     d=a*c
6     print("%d"%d)
7 if b=='-':
8     d=a-c
9     print("%d"%d)
10 if b=='*':
11     d=a*c
12     print("%d"%d)
13 if b=='/':
14     d=a/c
15     print("%0.1f"%d)
```

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

|   | Input        | Expected | Got |   |
|---|--------------|----------|-----|---|
| ✓ | 18<br>/<br>2 | 9.0      | 9.0 | ✓ |

Passed all tests! ✓

**Question 3** | Correct | Marked out of 1.00

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

Year Animal

2000 Dragon

2001 Snake

2002 Horse

2003 Sheep

2004 Monkey

2005 Rooster

2006 Dog

2007 Pig

2008 Rat

2009 Ox

2010 Tiger

2011 Hare

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

Sample Input 1

2010

Sample Output 1

2010 is the year of the Tiger.

Sample Input 2

2020

Sample Output 2

2020 is the year of the Rat.

**Answer:** (penalty regime: 0 %)

```
1 a=int(input())
2 r=(a-2000)%12
3
4 if r==0:
5     print("%d is the year of the Dragon."%a)
6 elif r==1:
7     print("%d is the year of the Snake."%a)
8 elif r==2:
9     print("%d is the year of the Horse."%a)
10 elif r==3:
11     print("%d is the year of the Sheep."%a)
12 elif r==4:
13     print("%d is the year of the Monkey."%a)
14 elif r==5:
15     print("%d is the year of the Rooster."%a)
16 elif r==6:
17     print("%d is the year of the Dog."%a)
18 elif r==7:
19     print("%d is the year of the Pig."%a)
20 elif r==8:
21     print("%d is the year of the Rat."%a)
22 elif r==9:
23     print("%d is the year of the Ox."%a)
```

```
24 elif r==10:  
25     print("%d is the year of the Tiger."%a)  
26 elif r==11:  
27     print("%d is the year of the Hare."%a)
```

|   | <b>Input</b> | <b>Expected</b>                | <b>Got</b>                     |   |
|---|--------------|--------------------------------|--------------------------------|---|
| ✓ | 2010         | 2010 is the year of the Tiger. | 2010 is the year of the Tiger. | ✓ |
| ✓ | 2020         | 2020 is the year of the Rat.   | 2020 is the year of the Rat.   | ✓ |

Passed all tests! ✓

**Question 4** | Correct | Marked out of 1.00

Write a program that accepts 5 inputs and returns the count of how many of those 5 are odd.

For example,

If the five inputs are 12, 17, 19, 14, and 115, there are three odd numbers 17, 19 and 115. So, the program must return 3.

Similarly,

If the five inputs are 15, 0, -12, 19, and 28, there are two odd numbers 15 and 19. So, the program must return 2.

Observe that zero is considered an even number.

**For example:**

| Input | Result |
|-------|--------|
| 12    | 3      |
| 17    |        |
| 19    |        |
| 14    |        |
| 115   |        |
| 15    | 2      |
| 0     |        |
| -12   |        |
| 19    |        |
| 28    |        |

**Answer:** (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 c=int(input())
4 d=int(input())
5 e=int(input())
6 count=0
7 if a%2!=0:
8     count=count+1
9 if b%2!=0:
10    count=count+1
11 if c%2!=0:
12    count=count+1
13 if d%2!=0:
14    count=count+1
15 if e%2!=0:
16    count=count+1
17 print("%d"%count)

```

|   | Input                       | Expected | Got |   |
|---|-----------------------------|----------|-----|---|
| ✓ | 12<br>17<br>19<br>14<br>115 | 3        | 3   | ✓ |
| ✓ | 15<br>8<br>-12<br>19<br>28  | 2        | 2   | ✓ |

Passed all tests! ✓

**Question 5** | Correct | Marked 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("%d is a leap year."%a)
4. elif a%100==0:
5.     print("%d is not a leap year."%a)
6. elif a%4==0:
7.     print("%d is a leap year."%a)
8. else:
9.     print("%d is not a leap year."%a)

```

|   | 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. | ✓ |
| ✓ | 2020  | 2020 is a leap year.     | 2020 is a leap year.     | ✓ |

Passed all tests! ✓

**Question 6** | Correct | Marked 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. c=a/2

```

```
4 • if b>=c:  
5     print("IN")  
6 • else:  
7     print("OUT")
```

|   | Input    | Expected | Got |   |
|---|----------|----------|-----|---|
| ✓ | 8<br>3   | OUT      | OUT | ✓ |
| ✓ | 8<br>5   | IN       | IN  | ✓ |
| ✓ | 20<br>9  | OUT      | OUT | ✓ |
| ✓ | 58<br>31 | IN       | IN  | ✓ |

Passed all tests! ✓

**Question 7** | Correct | Marked 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  |
| 500    | 1035.00 |

**Answer:** (penalty regime: 0 %)

```

1 a=float(input())
2 if a<199:
3     b=a*1.20
4     if b<100:
5         print("100.00")
6     elif b>400:
7         c=0.15*b
8         d=c+b
9         print("%0.2f"%d)
10    else:
11        print("%0.2f"%b)
12
13 elif a>=200 and a<400:
14     b=a*1.50
15     if b<100:
16         print("100.00")
17     elif b>400:
18         c=0.15*b
19         d=c+b
20         print("%0.2f"%d)
21    else:

```

```

22     print("%0.2f"%b)
23
24 • elif a>400 and a<600:
25     b=a*1.80
26 •     if b<100:
27         print("100.00")
28 •     elif b>400:
29         c=0.15*b
30         d=c+b
31         print("%0.2f"%d)
32 •     else:
33         print("0.2f"%b)
34 • else:
35     b=a*2.00
36 •     if b<100:
37         print("100.00")
38 •     if b>400:
39         c=0.15*b
40         d=c+b
41         print("%0.2f"%d)

```

|   | <b>Input</b> | <b>Expected</b> | <b>Got</b> |   |
|---|--------------|-----------------|------------|---|
| ✓ | 50           | 100.00          | 100.00     | ✓ |
| ✓ | 100.00       | 120.00          | 120.00     | ✓ |
| ✓ | 500          | 1035.00         | 1035.00    | ✓ |
| ✓ | 700          | 1610.00         | 1610.00    | ✓ |

Passed all tests! ✓

**Question 8** | Correct | Marked out of 1.00

Write a program to determine the type of berth when the seat / berth number in the train is given.

**Input Format:**

Input consists of a single integer. Assume that the range of input is between 1 and 72.

**Output Format:**

Output consists of a single string. [Upper or Middle or Lower or Side Lower or Side Upper]

**Sample Input 1:**

9

**Sample Output 1:**

Lower Berth

**Answer:** (penalty regime: 0 %)

```

1 a=int(input())
2 r=a%8
3
4 if r==1 or r==4:
5     print("Lower Berth")
6 elif r==2 or r==5:
7     print("Middle Berth")
8 elif r==3 or r==6:
9     print("Upper Berth")
10 elif r==7:
11     print("Side Lower Berth")
12 elif r==8:
13     print("Side Upper Berth")

```

|   | <b>Input</b> | <b>Expected</b>  | <b>Got</b>       |   |
|---|--------------|------------------|------------------|---|
| ✓ | 9            | Lower Berth      | Lower Berth      | ✓ |
| ✓ | 72           | Side Upper Berth | Side Upper Berth | ✓ |
| ✓ | 20           | Lower Berth      | Lower Berth      | ✓ |
| ✓ | 34           | Middle Berth     | Middle Berth     | ✓ |
| ✓ | 39           | Side Lower Berth | Side Lower Berth | ✓ |

Passed all tests! ✓

**Question 9** | Correct | Marked out of 1.00

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

**Input Format:**

Single integer input.

**Output Format:**

Yes or No.

**Example Input:**

24

**Output:**

Yes

**Example Input:**

26

**Output:**

No

**For example:**

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

**Answer:** (penalty regime: 0 %)

```

1 import math
2
3 a=int(input())
4 root=math.sqrt(a+1)
5 if root*root==a+1:
6     print("Yes")
7 else:
8     print("No")

```

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

Passed all tests! ✓

**Question 10** | Correct Marked 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<br>60<br>60 | That's a equilateral triangle |
| 40<br>40<br>80 | That's a isosceles triangle   |
| 50<br>60<br>70 | That's a scalene 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 and b!=c:
7     print("That's a isosceles triangle")
8 elif a==c and c!=b:
9     print("That's a isosceles triangle")
10 else:
11     print("That's a scalene triangle")

```

|   | Input           | Expected                      | Got                           |   |
|---|-----------------|-------------------------------|-------------------------------|---|
| ✓ | 60<br>60<br>60  | That's a equilateral triangle | That's a equilateral triangle | ✓ |
| ✓ | 40<br>40<br>80  | That's a isosceles triangle   | That's a isosceles triangle   | ✓ |
| ✓ | 50<br>60<br>70  | That's a scalene triangle     | That's a scalene triangle     | ✓ |
| ✓ | 50<br>50<br>80  | That's a isosceles triangle   | That's a isosceles triangle   | ✓ |
| ✓ | 10.<br>10<br>10 | That's a equilateral triangle | That's a equilateral triangle | ✓ |

Passed all tests! ✓

**Started on** Monday, 18 August 2025, 10:56 PM

**State** Finished

**Completed on** Monday, 18 August 2025, 11:14 PM

**Time taken** 17 mins 42 secs

**Question 1** | Incorrect. Marked out of 1.00

Dinesh is developing a geometry calculator program for a Math class. He wants to write a program that allows users to calculate the area of different geometric shapes such as circles, rectangles, and triangles.

Users can select the type of shape they want to calculate the area for, and the program will perform the corresponding calculation using a 'switch-case' statement.

**Formulas Used:**

Area of Circle =  $3.14 * \text{radius} * \text{radius}$

Area of Rectangle =  $\text{length} * \text{width}$

Area of Triangle =  $0.5 * \text{base} * \text{height}$

**Input Format :**

The first line of input consists of an integer, representing the geometric shape the user wants to calculate the area for (1 for a circle, 2 for a rectangle, and 3 for a triangle).

1. If the choice is 1, the second line of input consists of a double value **R**, representing the radius of the circle.
2. If the choice is 2, the second line and third lines of input consist of double values **L** and **B**, representing the length and width of the rectangle.
3. If the choice is 3, the second and third lines of input consist of double values **P** and **Q** representing the base and height of the triangle.

**Output Format :**

If the valid choice is entered, the output prints "**X sq.units**" where X represents the calculated area of the specific shape, rounded off to two decimal places.

Otherwise, the output prints "**Invalid choice**"

**For example:**

| Input           | Result         |
|-----------------|----------------|
| 1<br>5.1        | 81.67 sq.units |
| 2<br>4.3<br>6.1 | 26.23 sq.units |
|                 |                |

**Answer:** (penalty regime: 0 %)

```

1 choice=int(input().strip())
2
3 if choice==1:
4     r=float(input().strip())
5     Area=3.14*r*r
6     print(f'{Area:.2f} sq.units')
7
8 elif choice==2:
9     l=float(input().strip())
10    w=float(input().strip())
11    Area=l*w
12    print(f'{Area:.2f}sq.units')
13
14 elif choice==3:
15     b=float(input().strip())

```

```
16     n=float(input().strip())
17     Area=0.5*b*h
18     print(f'{Area:.2f}sq.units')
19
20 • else:
21     print("Invalid choice")
```

|   | <b>Input</b>    | <b>Expected</b> | <b>Got</b>      |   |
|---|-----------------|-----------------|-----------------|---|
| ✓ | 1<br>5.1        | 81.67 sq.units  | 81.67 sq.units  | ✓ |
| ✗ | 2<br>4.3<br>6.1 | 26.23 sq.units  | 26.23sq.units   | ✗ |
| ✗ | 3<br>3.7<br>5.6 | 10.36 sq.units  | 10.36sq.units   | ✗ |
| ✓ | 4               | Invalid choice  | Invalid choice  | ✓ |
| ✗ | 1<br>9.5        | 283.39 sq.units | 283.38 sq.units | ✗ |

Your code must pass all tests to earn any marks. Try again.

Show differences

**Question 2** | Correct | Marked out of 1.00

Akash is tasked with developing a program that calculates and categorizes blood pressure based on the given systolic and diastolic readings.

The program should use the following classifications:

1. Low Blood Pressure: Systolic < 90 mm Hg or Diastolic < 60 mm Hg
2. Normal Blood Pressure: Systolic ≤ 120 mm Hg and Diastolic ≤ 80 mm Hg
3. Prehypertension: Systolic ≤ 140 mm Hg and Diastolic ≤ 90 mm Hg
4. Stage 1 Hypertension: Systolic ≤ 160 mm Hg and Diastolic ≤ 100 mm Hg
5. Stage 2 Hypertension: Otherwise

Write a program to assist Akash in computing and classifying blood pressure levels based on input readings.

**Input Format :**

The input consists of two space-separated integers, representing the systolic blood pressure value **S** and diastolic blood pressure value **D**, respectively.

**Output Format :**

The output displays "Blood Pressure Category: " followed by the blood pressure category based on the provided input.

**Refer to the sample output for the exact text and format.**

**For example:**

| Input | Result   |
|-------|--|
| 50    | Blood Pressure Category: Low Blood Pressure    |
| 85    |  |
| 112   | Blood Pressure Category: Normal Blood Pressure |
| 78    |  |

**Answer:** (penalty regime: 0 %)

```

1 s=int(input())
2 d=int(input())
3
4 if s<90 or d<60:
5     category="Low Blood Pressure"
6 elif s<=120 and d<=80:
7     category="Normal Blood Pressure"
8 elif s<=140 and d<=90:
9     category="Prehypertension"
10 elif s<=160 and d<=100:
11     category="Stage 1 Hypertension"
12 else:
13     category="Stage 2 Hypertension"
14 print("Blood Pressure Category:",category)

```

|   | Input      | Expected                                       | Got  |   |
|---|------------|--|--|---|
| ✓ | 50<br>85   | Blood Pressure Category: Low Blood Pressure    | Blood Pressure Category: Low Blood Pressure    | ✓ |
| ✓ | 112<br>78  | Blood Pressure Category: Normal Blood Pressure | Blood Pressure Category: Normal Blood Pressure | ✓ |
| ✓ | 135<br>86  | Blood Pressure Category: Prehypertension       | Blood Pressure Category: Prehypertension       | ✓ |
| ✓ | 145<br>98  | Blood Pressure Category: Stage 1 Hypertension  | Blood Pressure Category: Stage 1 Hypertension  | ✓ |
| ✓ | 170<br>110 | Blood Pressure Category: Stage 2 Hypertension  | Blood Pressure Category: Stage 2 Hypertension  | ✓ |

Passed all tests! ✓

**Question 3** | Correct | Marked out of 1.00

Alice, an insurance agent, needs a program to calculate the insurance premium for her clients based on their age and health condition.

The premium amount is determined by the following rules:

1. If the client's age is between 18 and 30 years:
  2. 'excellent' health condition: 500.0
  3. 'good' health condition: 700.0
  4. Any other health condition: 900.0
5. If the client's age is between 31 and 50 years:
  6. 'excellent' health condition: 700.0
  7. 'good' health condition: 900.0
  8. Any other health condition: 1100.0

Write a program that takes the age and health condition of a client as input and outputs the corresponding insurance premium.

**Input Format :**

The first line of input contains an integer representing the age of the client.

The second line contains a string representing the health condition of the client.

**Output Format :**

The output prints a double value, representing the insurance premium.

**For example:**

| Input           | Result |
|-----------------|--------|
| 25<br>excellent | 500.0  |
| 31<br>good      | 900.0  |
| 41<br>poor      | 1100.0 |

**Answer:** (penalty regime: 0 %)

```

1 age=int(input().strip())
2 health=input().strip().lower()
3
4 premium=0.0
5 if 18<=age<=30:
6     if health=="excellent":
7         premium=500.0
8     elif health=="good":
9         premium=700.0
10    else:
11        premium=900.0
12 elif 31<=age<=50:
13     if health=="excellent":
14         premium=700.0
15     elif health=="good":
16         premium=900.0
17     else:
18         premium=1100.0
19 else:
20     premium=-1
  
```

|                 |          |          |
|-----------------|----------|----------|
| Input           | Expected | Got      |
| 25<br>excellent | 500.0    | 500.0 ✓  |
| 31<br>good      | 900.0    | 900.0 ✓  |
| 41<br>poor      | 1100.0   | 1100.0 ✓ |
| 50<br>good      | 900.0    | 900.0 ✓  |

Passed all tests! ✓

**Question 4** | Correct | Marked out of 1.00

Arun is working on a project to automate the process of determining whether a student has passed or failed based on their subject marks.

He aims to create a simple program that takes positive integers as marks for five subjects from the user. If the average of the marks is greater than or equal to 50, the student has passed the exam, Else, the student has failed.

Help Arun to implement the project.

**Input Format :**

The input consists of five space-separated integers, representing the marks in five subjects.

**Output Format :**

The first line of output prints "Average score: " followed by an integer representing the average score.

The second line prints one of the following:

1. If the condition is satisfied, print "The student has passed".
2. Otherwise, the output prints "The student has failed".

**For example:**

| Input | Result                 |
|-------|------------------------|
| 50    | Average score: 50      |
| 60    | The student has passed |
| 70    |                        |
| 80    |                        |
| 90    |                        |
| 39    | Average score: 41      |
| 25    | The student has failed |
| 30    |                        |
| 45    |                        |
| 67    |                        |

**Answer:** (penalty regime: 0 %)

```

1 mark1=int(input())
2 mark2=int(input())
3 mark3=int(input())
4 mark4=int(input())
5 mark5=int(input())
6 average=(mark1+mark2+mark3+mark4+mark5)//5
7 print(f"Average score: {average}")
8 if average>=50:
9     print("The student has passed")
10 else:
11     print("The student has failed")

```

|   | Input                      | Expected                                    | Got   |   |
|---|----------------------------|---|---|---|
| ✓ | 50<br>60<br>70<br>80<br>90 | Average score: 70<br>The student has passed | Average score: 70<br>The student has passed | ✓ |
| ✓ | 39<br>25<br>30<br>45<br>67 | Average score: 41<br>The student has Failed | Average score: 41<br>The student has Failed | ✓ |

Passed all tests! ✓

**Question 5** | Correct | Marked out of 1.00

Bruce is working on a task that involves manipulating integers. He needs to rotate the digits of an integer to the right by one position.

Write a program to help Bruce accomplish this task using a do-while loop.

**Input Format :**

The input consists of a single integer **N**.

**Output Format :**

The output prints the given integer with its digits rotated to the right by one position.

**For example:**

| Input | Result |
|-------|--------|
| 647   | 764    |

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 s=str(n)
3 rotated = s[-1]+s[:-1]
4 print(int(rotated))

```

|   | Input | Expected | Got   |   |
|---|-------|----------|-------|---|
| ✓ | 647   | 764      | 764   | ✓ |
| ✓ | 78436 | 67843    | 67843 | ✓ |
| ✓ | 82644 | 48264    | 48264 | ✓ |
| ✓ | 62734 | 46273    | 46273 | ✓ |
| ✓ | 2836  | 6283     | 6283  | ✓ |

Passed all tests! ✓



**Started on** Friday, 22 August 2025, 3:21 PM

**State** Finished

**Completed on** Monday, 25 August 2025, 8:57 AM

**Time taken** 2 days 17 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 to find the count of the number of prime numbers in a specified range.

The starting and ending number of the range will be provided as input to the program.

Assumption:  $2 \leq \text{starting number of the range} \leq \text{ending number of the range} \leq 7919$

Example1: If the starting and ending number of the range is given as 2 and 20, the program must return 8, because there are 8 prime numbers in the specified range from 2 to 20, namely (2, 3, 5, 7, 11, 13, 17, 19)

Example2: If the starting and ending number of the range is given as 700 and 725, the program must return 3, because there are 3 prime numbers in the specified range from 700 to 725, namely (701, 709, 719)

**For example:**

| Input | Result |
|-------|--------|
| 2     | 8      |
| 20    |        |
| 700   | 3      |
| 725   |        |

**Answer:** (penalty regime: 0 %)

```

1 start=int(input())
2 end=int(input())
3 d=0
4 for i in range(start,end+1):
5     for j in range(2,i):
6         if(i%j==0):
7             break
8     else:
9         d=d+1
10 print(d)
11

```

|   | Input      | Expected | Got |   |
|---|------------|----------|-----|---|
| ✓ | 2<br>20    | 8        | 8   | ✓ |
| ✓ | 700<br>725 | 3        | 3   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 2** | Correct | Mark 1.00 out of 1.00

Write a program that given an integer 'n', prints the number of integers that are less than or equal to 'n' and co-prime to 'n'

Two integers a and b are said to be relatively prime or co-prime if the only positive integer that evenly divides both of them is 1. That is, the only common positive factor of the two numbers is 1. This is equivalent to their greatest common divisor being 1.

Input Format:

One line containing the value of 'n', where  $1 \leq n \leq 10,000$

Output Format:

One line containing the number of integers that are co-prime to n and less than or equal to 'n'

Sample Test Cases

Test Case 1

Input

10

Output

4

Test Case 2

Input

23

Output

22

Test Case 3

Input

11

Output

10

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 c=0
3 for i in range(1,n+1):
4     a=i
5     b=n
6     while(b!=0):
7         temp=b
8         b=a%b
9         a=temp
10    if(a==1):
```

```
11 |     c+=1  
12 | print(c)
```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 10    | 4        | 4   | ✓ |
| ✓ | 23    | 22       | 22  | ✓ |
| ✓ | 11    | 10       | 10  | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | Correct | Mark 1.00 out of 1.00

Let's print a chessboard!

Write a program that takes input:

Integer N(represents the rows and columns of a chessboard) and also the starting character of the chessboard

Output Format

Print the chessboard as per the given examples

Sample Input / Output

Input:

2

W

Output:

WB

BW

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 s=input()
3 first='W'
4 second='B'
5 if(s=='B'):
6     first='B'
7     second='W'
8 for i in range(1,n+1):
9     for j in range(1,n+1):
10        if((i+j)%2==0):
11            print(first, end='')
12        else:
13            print(second, end='')
14 print()
15
16

```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 2     | WB       | WB  | ✓ |
|   | W     | BW       | BW  |   |
| ✓ | 3     | BWB      | BkB | ✓ |
|   | B     | WBW      | WBW |   |
|   |       | BWB      | BkB |   |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 4** | Correct | Mark 1.00 out of 1.00

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

**Input Format:**

Integer input from stdin.

**Output Format:**

Perfect square greater than N.

**Example Input:**

10

**Output:**

16

**For example:**

| Input | Result |
|-------|--------|
| 10    | 16     |
| 20    | 25     |

**Answer:** (penalty regime: 0 %)

```

1 import math
2 a=int(input())
3 root=int(math.sqrt(a)+1)
4 a=root*root
5 print(a)
6
7
8
9

```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 10    | 16       | 16  | ✓ |
| ✓ | 20    | 25       | 25  | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 5** | 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 |
|-------|--------|
| 8     | YES    |
| 3     |        |
| 4     |        |
| 2     |        |
| 6     |        |

**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 for _ in range(k):
7     x1+=v1
8     x2+=v2
9     if(x1==x2):
10         print("YES")
11         break
12 else:
13     print("NO")
14

```

|   | Input                 | Expected | Got |   |
|---|-----------------------|----------|-----|---|
| ✓ | 8<br>3<br>4<br>2<br>6 | YES      | YES | ✓ |
| ✓ | 8<br>3<br>2<br>4<br>B | NO       | NO  | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 6** | Correct | Mark 1.00 out of 1.00

An automorphic number is a number whose square ends with the number itself.

For example, 5 is an automorphic number because  $5 \times 5 = 25$ . The last digit is 5 which same as the given number.

If it is an automorphic number display "Automorphic" else display "Not Automorphic".

Input Format:

Take a Integer from Keyboard

Output Format:

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

Example input:

5

Output:

Automorphic

Example input:

25

Output:

Automorphic

Example input:

7

Output:

Not Automorphic

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 c=0
3 a=n
4 while(n!=0):
5     r=n%10
6     n=n//10
7     c+=1
8     b=a*a
9     if(a%10**c==b%10**c):
10        print("Automorphic")
11    else:
12        print("Not Automorphic")
13
14
```

|   | Input | Expected        | Got             |   |
|---|-------|-----------------|-----------------|---|
| ✓ | 5     | Automorphic     | Automorphic     | ✓ |
| ✓ | 625   | Automorphic     | Automorphic     | ✓ |
| ✓ | 7     | Not Automorphic | Not Automorphic | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 7** | Correct | Mark 1.00 out of 1.00

A Number is said to be Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself. Write a program to print number is Disarium or not.

**Input Format:**

Single Integer Input from stdin.

**Output Format:**

Yes or No.

**Example Input:**

175

**Output:**

Yes

**Explanation**

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

**Example Input:**

123

**Output:**

No

**For example:**

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

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 c=0
3 s=0
4 a=n
5 num=n
6 • while(n!=0):
7     rem=n%10
8     n=n//10
9     c=c+1
10 • for i in range(c,0,-1):
11     b=a%10
12     a=a//10
13     s=s+b**i
14 • if(s==num):
15     print('Yes')
16 • else:
17     print('No')
```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 175   | Yes      | Yes | ✓ |
| ✓ | 123   | No       | No  | ✓ |
| ✓ | 89    | Yes      | Yes | ✓ |
| ✓ | 98    | No       | No  | ✓ |
| ✓ | 518   | Yes      | Yes | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 8** | Correct | Mark 1.00 out of 1.00

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

Assumption: The input number will be a positive integer number  $\geq 1$  and  $\leq 25000$ .

Some examples are as below.

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

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

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

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

**For example:**

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

**Answer:** (penalty regime: 0 %)

```

1 n=input()
2 c=0
3 for i in range(len(n)):
4     repeated=False
5     for j in range(len(n)):
6         if(i!=j and n[i]==n[j]):
7             repeated=True
8             break
9     else:
10         c=c+1
11 print(c)

```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 292   | 1        | 1   | ✓ |
| ✓ | 1015  | 2        | 2   | ✓ |
| ✓ | 108   | 3        | 3   | ✓ |

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 22    | 8        | 8   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 9** | Correct | Mark 1.00 out of 1.00

Given a positive integer  $a$ , find the smallest positive integer  $b$  whose multiplication of each digit equals to  $a$ .

If there is no answer or the answer is not fit in 32-bit signed integer, then return 0.

**Example 1**

Input:

48

Output:

68

**Example 2**

Input:

15

Output:

35

**For example:**

| Input | Result |
|-------|--------|
| 48    | 68     |
| 15    | 35     |

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 flag = False
3 for i in range(1,10):
4     for j in range(1,10):
5         if i*j==n:
6             print(i,j,sep=' ')
7             flag=True
8 if flag:
9     break

```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 48    | 68       | 68  | ✓ |
| ✓ | 15    | 35       | 35  | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 10** | Correct Mark 1.00 out of 1.00

Write python program to print the following pattern based on input size.

Input:

3

Output:

```
1
2 3
4 5 6
```

For example:

| Input | Result                                       |
|-------|--|
| 4     | 1         2 3         4 5 6         7 8 9 10 |

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 num=1
3 for i in range(1,n+1):
4     for _ in range(n-1):
5         print(end=" ")
6     for j in range(i):
7         print(num,end=" ")
8         num+=1
9     print()
10
11
```

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

|   | Input | Expected                      | Got                           |   |
|---|-------|-------------------------------|-------------------------------|---|
| ✓ | 4     | 1<br>2 3<br>4 5 6<br>7 8 9 10 | 1<br>2 3<br>4 5 6<br>7 8 9 10 | ✓ |
|   |       |                               |                               |   |
|   |       |                               |                               |   |
|   |       |                               |                               |   |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Started on** Monday, 25 August 2025, 9:14 AM

**State** Finished

**Completed on** Monday, 25 August 2025, 9:28 PM

**Time taken** 12 hours 14 mins

**Marks** 10.00/10.00

**Grade** 100.00 out of 100.00

**Question 1** | 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  $p^{\text{th}}$  element of the list, sorted ascending. If there is no  $p^{\text{th}}$  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<br>3 | 5      |
| 10<br>5 | 0      |
| 1<br>1  | 1      |

**Answer:** (penalty regime: 0 %)

```

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

```

|   | Input   | Expected | Got |   |
|---|---------|----------|-----|---|
| ✓ | 10<br>3 | 5        | 5   | ✓ |
| ✓ | 10<br>5 | 0        | 0   | ✓ |
| ✓ | 1<br>1  | 1        | 1   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 2** | Correct | Mark 1.00 out of 1.00

Given an integer `num`, repeatedly add all its digits until the result has only one digit, and return it.

**Example 1:**

**Input:** num = 38

**Output:** 2

**Explanation:** The process is

38 → 3 + 8 → 11

11 → 1 + 1 → 2

Since 2 has only one digit, return it.

**Example 2:**

**Input:** num = 0

**Output:** 0

**For example:**

| Input | Result |
|-------|--------|
| 38    | 2      |
| 0     | 0      |

**Answer:** (penalty regime: 0 %)

```

1 n=input()
2 s=0
3 while(len(n)>1):
4     num=int(n)
5     while(num>0):
6         r=num%10
7         s+=r
8         num//=10
9     n=str(s)
10    s=0
11 print(n)
12

```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 38    | 2        | 2   | ✓ |
| ✓ | 0     | 0        | 0   | ✓ |

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 11    | 2        | 2   | ✓ |
| ✓ | 50    | 5        | 5   | ✓ |
| ✓ | 81    | 9        | 9   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | Correct | Mark 1.00 out of 1.00

A **happy number** is a number defined by the following process:

- Starting with any positive integer, replace the number by the sum of the squares of its digits.
- Repeat the process until the number equals 1 (where it will stay), or it **loops endlessly in a cycle** which does not include 1.
- Those numbers for which this process **ends in 1** are happy.

Print `true` if  $n$  is a happy number, and `false` if not.

For example:

| Input | Result |
|-------|--------|
| 19    | True   |
| 2     | False  |

Answer: (penalty regime: 0 %)

```

1 n=input()
2 while(len(n)!=1):
3     sum=0
4     num=int(n)
5     while(num>0):
6         rem=num%10
7         sum+=rem*rem
8         num/=10
9     num=sum
10    sum=0
11    n=str(num)
12 if(n=="1"):
13     print("True")
14 else:
15     print("False")

```

|   | Input | Expected | Got   |   |
|---|-------|----------|-------|---|
| ✓ | 19    | True     | True  | ✓ |
| ✓ | 2     | False    | False | ✓ |
| ✓ | 82    | True     | True  | ✓ |
| ✓ | 16    | False    | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00 / 1.00

**Question 4** | Correct | Mark 1.00 out of 1.00

Write a program in Python to display a pyramid with "\*" as follows,

**For example:**

| Input | Result |
|-------|--------|
| 4     | *      |
|       | ***    |
|       | *****  |
|       | *****  |

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 v=1
3 for i in range(1,n+1):
4     for k in range(n-i):
5         print(" ",end="")
6     for j in range(2*i-1):
7         print("*",end="")
8     print()

```

|   | Input | Expected | Got   |   |
|---|-------|----------|-------|---|
| ✓ | 4     | *        | *     | ✓ |
|   |       | ***      | ***   |   |
|   |       | *****    | ***** |   |
|   |       | *****    | ***** |   |
| ✓ | 2     | *        | *     | ✓ |
|   |       | ***      | ***   |   |
| ✓ | 5     | *        | *     | ✓ |
|   |       | ***      | ***   |   |
|   |       | *****    | ***** |   |
|   |       | *****    | ***** |   |
|   |       | *****    | ***** |   |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 5** | Correct | Mark 1.00 out of 1.00

An **ugly number** is a *positive* integer which does not have a prime factor other than 2, 3, and 5.

Given an integer  $n$ , Print **True** if  $n$  is an **ugly number**, Otherwise Print **False**.

For example:

| Input | Result |
|-------|--------|
| 6     | True   |
| 14    | False  |

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 • while(n%2==0):
3     n/=2
4 • while(n%3==0):
5     n/=3
6 • while(n%5==0):
7     n/=5
8 print(n==1)

```

|   | Input | Expected | Got   |   |
|---|-------|----------|-------|---|
| ✓ | 6     | True     | True  | ✓ |
| ✓ | 14    | False    | False | ✓ |
| ✓ | 125   | True     | True  | ✓ |
| ✓ | 21    | False    | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 6** | Correct | Mark 1.00 out of 1.00

Given an integer  $n$ , return the number of trailing zeroes in  $n!$ .

Note that  $n! = n * (n - 1) * (n - 2) * \dots * 3 * 2 * 1$ .

**Example 1:**

**Input:**  $n = 3$

**Output:** 0

**Explanation:**  $3! = 6$ , no trailing zero.

**Example 2:**

**Input:**  $n = 5$

**Output:** 1

**Explanation:**  $5! = 120$ , one trailing zero.

**Example 3:**

**Input:**  $n = 0$

**Output:** 0

**Constraints:**

- $0 \leq n \leq 10^4$

**For example:**

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

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 fact=1
3 c=0
4 for i in range(2,n+1):
5     fact*=i
6 fact=str(fact)
7 for i in range(len(fact)-1,-1,-1):
8     if(fact[i]=='0'):
9         c+=1
10    else:
11        break
12 print(c)

```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 3     | 9        | 9   | ✓ |
| ✓ | 5     | 1        | 1   | ✓ |
| ✓ | 8     | 0        | 0   | ✓ |
| ✓ | 10    | 2        | 2   | ✓ |
| ✓ | 25    | 6        | 6   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 7** | Correct | Mark 1.00 out of 1.00

Given an integer `num`, return the number of digits in `num` that divide `num`.

An integer `val` divides `num` if `num % val == 0`.

**Example 1:**

```
Input: num = 7
Output: 1
Explanation: 7 divides itself, hence the answer is 1.
```

**Example 2:**

```
Input: num = 121
Output: 2
Explanation: 121 is divisible by 1, but not 2. Since 1 occurs twice as a digit, we return 2.
```

**Example 3:**

```
Input: num = 1248
Output: 4
Explanation: 1248 is divisible by all of its digits, hence the answer is 4.
```

**For example:**

| Input | Result |
|-------|--------|
| 7     | 1      |
| 121   | 2      |
| 1248  | 4      |

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 o=n
3 c=0
4 • while(n>0):
5     d=n%10
6     if(o%d==0):
7         c+=1
8     n//=10
9 print(c)
```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 7     | 1        | 1   | ✓ |
| ✓ | 121   | 2        | 2   | ✓ |
| ✓ | 1248  | 4        | 4   | ✓ |
| ✓ | 12    | 2        | 2   | ✓ |
| ✓ | 45    | 1        | 1   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 8** | Correct | Mark 1.00 out of 1.00

You are climbing a staircase. It takes  $n$  steps to reach the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

**Example 1:**

```
Input: n = 2
Output: 2
Explanation: There are two ways to climb to the top.
1. 1 step + 1 step
2. 2 steps
```

**Example 2:**

```
Input: n = 3
Output: 3
Explanation: There are three ways to climb to the top.
1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step
```

**Constraints:**

- $1 \leq n \leq 45$

**For example:**

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

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 c=0
3 a=1
4 b=1
5 • if(n==1):
6     print(1)
7 • if(n==0):
8     print(1)
9 • for i in range(2,n+1):
10    c=a+b
11    a,b=b,c
12    print(c)
```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | 2     | 2        | 2   | ✓ |
| ✓ | 3     | 3        | 3   | ✓ |
| ✓ | 4     | 5        | 5   | ✓ |
| ✓ | 5     | 8        | 8   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 9** | Correct | Mark 1.00 out of 1.00

A **perfect number** is a **positive integer** that is equal to the sum of its **positive divisors**, excluding the number itself. A **divisor** of an integer  $x$  is an integer that can divide  $x$  evenly.

Given an integer  $n$ , return **True** if  $n$  is a perfect number, otherwise return **False**.

**Example 1:**

```
Input: num = 28
Output: True
Explanation: 28 = 1 + 2 + 4 + 7 + 14
1, 2, 4, 7, and 14 are all divisors of 28.
```

**Example 2:**

```
Input: num = 7
Output: False
```

**Constraints:**

- $1 \leq \text{num} \leq 10^8$

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 s=0
3 for i in range(1,n):
4     if(n%i==0):
5         s+=i
6 if(s==n):
7     print("True")
8 else:
9     print("False")
```

|   | <b>Input</b> | <b>Expected</b> | <b>Got</b> |   |
|---|--------------|-----------------|------------|---|
| ✓ | 28           | True            | True       | ✓ |
| ✓ | 7            | False           | False      | ✓ |
| ✓ | 8128         | True            | True       | ✓ |
| ✓ | 496          | True            | True       | ✓ |

|   | Input | Expected | Got   |   |
|---|-------|----------|-------|---|
| ✓ | 500   | False    | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 10** | Correct Mark 1.00 out of 1.00

Given a positive integer  $n$ , write a function that returns the number of  set bits in its binary representation (also known as the [Hamming weight](#)):

**Example 1:****Input:**  $n = 11$ **Output:** 3**Explanation:**

The input binary string **1011** has a total of three set bits.

**Example 2:****Input:**  $n = 128$ **Output:** 1**Explanation:**

The input binary string **10000000** has a total of one set bit.

**Example 3:****Input:**  $n = 2147483645$ **Output:** 30**Explanation:**

The input binary string **1111111111111111111111111101** has a total of thirty set bits.

**For example:**

| Input | Result |
|-------|--------|
| 11    | 3      |
| 128   | 1      |

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 c=0
3 k=bin(n)[2:]
4 for i in k:
5     if(i=='1'):
6         c+=1
7 print(c)

```

|   | Input      | Expected | Got |   |
|---|------------|----------|-----|---|
| ✓ | 11         | 3        | 3   | ✓ |
| ✓ | 128        | 1        | 1   | ✓ |
| ✓ | 32         | 1        | 1   | ✓ |
| ✓ | 2147483645 | 38       | 38  | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Started on** Saturday, 30 August 2025, 6:35 PM

**State** Finished

**Completed on** Saturday, 30 August 2025, 7:33 PM

**Time taken** 58 mins 23 secs

**Marks** 5.00/5.00

**Grade** 100.00 out of 100.00

**Question 1** | 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     a=str(n)
3     b=0
4 •     for i in range(1,5):
5         b=b+int(a*i)
6     return b

```

|   | 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.



**Question 2** | 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     c=0
3     while n:
4         if(n%4==0):
5             c+=1
6             n=n-4
7         elif(n%3==0):
8             c+=1
9             n=n-3
10        elif(n%2==0):
11            c+=1
12            n=n-2
13        elif(n%1==0):
14            c+=1
15            n=n-1
16    return c
17
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | 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 •     for i in [2,3,5]:
3 •         while n%i==0:
4 •             n=n//i
5 •     if n==1:
6 •         return'ugly'
7 •     else:
8 •         return'not ugly'
9

```

|   | 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.

**Question 4** | 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

$1 \leq \text{orderValue} < 10e100000$

Input

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

Output

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

Example Input

578

Output

12

For example:

| Test                                       | Result |
|--|--------|
| <code>print(christmasDiscount(578))</code> | 12     |

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 • def christmasDiscount(n):
2     s=0
3     a=n
4 •     while a!=0:
5         r=a%10
6         a=a//10
7         c=0
8 •         for i in range(1,r+1):
9             if r%i==0:
10                 c=c+1
11 •         if c==2:
12             s=s+r
13     return(s)
14
```

| Test                              | Expected | Got  |
|-----------------------------------|----------|------|
| ✓ print(christmasDiscount(578))   | 12       | 12 ✓ |
| ✓ print(christmasDiscount(57))    | 12       | 12 ✓ |
| ✓ print(christmasDiscount(222))   | 6        | 6 ✓  |
| ✓ print(christmasDiscount(77777)) | 35       | 35 ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**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     l=0
4     r=len(n)-1
5 •     while(l<=r):
6         if(n[l]=='1' and n[r]=='1') or (n[l]=='0' and n[r]=='0') or\
7             (n[l]=='8' and n[r]=='8') or (n[l]=='6' and n[r]=='9') or (n[l]=='9' and n[r]=='6'):
8             return True
9 •         else:
10            return False
11

```

|   | Test                        | Expected | Got   |   |
|---|-----------------------------|----------|-------|---|
| ✓ | print(Strobogrammatic(69))  | True     | True  | ✓ |
| ✓ | print(Strobogrammatic(88))  | True     | True  | ✓ |
| ✓ | print(Strobogrammatic(962)) | False    | False | ✓ |
| ✓ | print(Strobogrammatic(66))  | False    | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Started on** Friday, 29 August 2025, 4:04 PM

**State** Finished

**Completed on** Saturday, 30 August 2025, 10:01 PM

**Time taken** 1 day 5 hours

**Marks** 5.00/5.00

**Grade** 100.00 out of 100.00

**Question 1** | Correct · Mark 1.00 out of 1.00

Given an integer number and you have to count the digits using recursion using Python program. In this program, you will be reading an integer number and counting the total digits, using a function `countDigits()` which will take a number as an argument and return the count after recursion process.

**Input Format:** The first and only line of the input contains a single integer  $n$

**Output Format:** Output a single line denoting the number of digits in  $n$ .

**For example:**

| Test                                 | Result |
|--------------------------------------|--------|
| <code>print(countDigits(800))</code> | 3      |

**Answer:** (penalty regime: 0 %)

[Reset answer](#)

```

1 • def countDigits(n):
2 •     if n<10:
3 •         return 1
4 •     else:
5 •         return 1 + countDigits(n//10)
6

```

|   | Test                                   | Expected | Got |   |
|---|--|----------|-----|---|
| ✓ | <code>print(countDigits(12345))</code> | 5        | 5   | ✓ |
| ✓ | <code>print(countDigits(800))</code>   | 3        | 3   | ✓ |

Passed all tests! ✓

[Correct](#)

Marks for this submission: 1.00/1.00.

**Question 2** | Correct | Mark 1.00 out of 1.00

Euclid was a Greek mathematician who lived approximately 2,300 years ago. His algorithm for computing the greatest common divisor of two positive integers,  $a$  and  $b$ , is both efficient and recursive. It is outlined below:

```
If b is 0 then
    return a
Else
    Set c equal to the remainder when a is divided by b
    Return the greatest common divisor of b and c
```

Write a Recursive function that implements Euclid's algorithm and uses it to determine the greatest common divisor of two integers entered by the user. Test your program with some very large integers. The result will be computed quickly, even for huge numbers consisting of hundreds of digits, because Euclid's algorithm is extremely efficient.

**For example:**

| Test                  | Result |
|-----------------------|--------|
| print(gcd(8, 12))     | 4      |
| print(gcd(720, 1000)) | 40     |

**Answer:** (penalty regime: 0 %)

[Reset answer](#)

```
1 • def gcd(a,b):
2 •     if b==0:
3 •         return a
4 •     else:
5 •         return gcd(b,a%b)
```

|   | Test                  | Expected | Got |   |
|---|-----------------------|----------|-----|---|
| ✓ | print(gcd(8, 12))     | 4        | 4   | ✓ |
| ✓ | print(gcd(720, 1000)) | 40       | 40  | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | Correct | Mark 1.00 out of 1.00

Complete the recursive function to return Binary Equivalent of an Integer using Recursion.

Sample Test Cases

Test Case 1

Input

10

Output

1010

Test Case 2

Input

257

Output

100000001

For example:

| Test                    | Result    |
|-------------------------|-----------|
| print(binayNumber(10))  | 1010      |
| print(binayNumber(257)) | 100000001 |

Answer: (penalty regime: 0 %)

Reset answer

```
1 • def binayNumber(n):  
2 •     if n<2:  
3 •         return str(n)  
4 •     else:  
5 •         return binayNumber(n//2)+str(n%2)  
6 •
```

| Test                      | Expected  | Got       |   |
|---------------------------|-----------|-----------|---|
| ✓ print(binayNumber(10))  | 1010      | 1010      | ✓ |
| ✓ print(binayNumber(257)) | 100000001 | 100000001 | ✓ |

Passed all tests! ✓

Correct!

Marks for this submission: 1.00/1.00.

**Question 4** | Correct | Mark 1.00 out of 1.00

Complete a Recursive Function to find if a given number N can be expressed as a sum of two prime numbers.

Note: YOU MUST OPTIMIZE the logic to find whether a number is prime or not, as very large prime numbers are provided as input. If the logic is not optimized your program will NOT get executed within the given time limit.

**Input Format:**

First line contains number N.

**Output Format:**

Return either yes or no..

**Boundary Conditions / Constraints:**

$3 \leq N \leq 10^9$

**Example Input/Output 1:****Input:**

20

**Output:**

yes

**Input:**

23

**Output:**

no

**Explanation:**

20 can be expressed as 17+3

23 cannot be expressed as sum of two primes

**For example:**

| Test                                  | Result |
|---------------------------------------|--------|
| <code>print(checkPrimeSum(20))</code> | yes    |
| <code>print(checkPrimeSum(23))</code> | no     |

**Answer:** (penalty regime: 0 %)

[Reset answer](#)

```

1 • def isPrime(num, divisor=2):
2 •     if num<2:
3 •         return False
4 •     if divisor * divisor > num:
5 •         return True
6 •     if num%divisor ==0:
7 •         return False
8 •     return isPrime(num, divisor+1)
9
10 • def checkPrimeSum(n,i=2):
11 •     if i > n//2:

```

```
12     return "no"
13 •   if isPrime(i) and isPrime(n-i):
14       return "yes"
15   return checkPrimeSum(n,i+1)
```

|   | Test                     | Expected | Got |   |
|---|--------------------------|----------|-----|---|
| ✓ | print(checkPrimeSum(20)) | yes      | yes | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 5** | Correct | Mark 1.00 out of 1.00

The notion of a palindrome was introduced previously. In this exercise you will write a recursive function that determines whether or not a string is a palindrome. The empty string is a palindrome, as is any string containing only one character. Any longer string is a palindrome if its first and last characters match, and if the string formed by removing the first and last characters is also a palindrome.

Write a program that reads a string from the user and uses your recursive function to determine whether or not it is a palindrome. Then your program should display an appropriate message for the user.

Sample Input

malayalam

Sample Output

That was a palindrome!

Sample Input

madan

Sample Output

That is not a palindrome.

**Answer:** (penalty regime: 0 %)

[Reset answer](#)

```

1 • def isPalindrome(s):
2     # Base case: The empty string is a palindrome. So is a string containing only 1 character.
3 •     if len(s) <= 1:
4         return True
5 •     if(s[0]==s[-1]):
6         return(isPalindrome(s[1:-1]))
7
8     # Recursive case: The string is a palindrome only if the first and last characters match, and
9     # the rest of the string is a palindrome
10
11
12 # Check whether or not a string entered by the user is a palindrome
13 # Read the string from the user
14 line=input()
15 # Check its status and display the result
16 • if isPalindrome(line):
17     print("That was a palindrome!")
18 • else:
19     print("That is not a palindrome. ")
20
21
22

```

|   | Input     | Expected                  | Got                       |   |
|---|-----------|---------------------------|---------------------------|---|
| ✓ | malayalam | That was a palindrome!    | That was a palindrome!    | ✓ |
| ✓ | madan     | That is not a palindrome. | That is not a palindrome. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Started on** Monday, 8 September 2025, 10:14 PM

**State** Finished

**Completed on** Tuesday, 9 September 2025, 6:11 PM

**Time taken** 19 hours 57 mins

**Marks** 10.00/10.00

**Grade** 100.00 out of 100.00

**Question 1** | 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 s2=input().strip()
2 s1=input().strip()
3 index=s2.find(s1)
4 print(index)
```

|   | Input                    | Expected | Got |   |
|---|--------------------------|----------|-----|---|
| ✓ | thistest123string<br>123 | 8        | 8   | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00;

**Question 2** | 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<br>rajalakshmi<br>arvijayakumar |

**Answer:** (penalty regime: 0 %)

```
1 s=input().strip()
2 username,rest=s.split("@",1)
3 parts=rest.split(".")
4 domain=parts[0]
5 extension=".join(parts[1:])
6 print(extension)
7 print(domain)
8 print(username)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | 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** = "|12iz4n":

Return true.

**Example 2:****Input**

apple

a2e

**Output**

false

**Explanation**

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

Return false.

**Answer:** (penalty regime: 0 %)

```

1 w=input()
2 abb=input()
3 i=0
4 j=0
5 n=len(w)
6 m=len(abb)
7 valid=True
8 while i<n and j<m:
9     if abb[j].isalpha():
10        if w[i]!=abb[j]:
11            valid=False
12            break
13        i+=1
14        j+=1
15    else:
16        if abb[j]=='0':
17            valid=False
18            break
19        num=0
20        while j<m and abb[j].isdigit():
21            num=num*10+int(abb[j])
22            j+=1
23        i+=num
24        if i>n:
25            valid=False
26            break
27 if i==n and j==m and valid:
28     print('true')
29 else:
30     print('false')
```

|   | Input                            | Expected | Got   |   |
|---|----------------------------------|----------|-------|---|
| ✓ | internationalization<br>i112iz4n | true     | true  | ✓ |
| ✓ | apple<br>a2e                     | false    | false | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

**Question 4** | 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","range","return","stru
2 w=input().strip()
3 print(f"{w} is a keyword" if w in keywords else f"{w} 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.

**Question 5** | 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**

aabbbbcccccc

**Answer:** (penalty regime: 0 %)

```

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

```

|   | Input   | Expected            | Got                 |   |
|---|---------|---------------------|---------------------|---|
| ✓ | a2b4c6  | aabbbbcccccc        | aabbbbcccccc        | ✓ |
| ✓ | a12b3d4 | aaaaaaaaaaaabbddddd | aaaaaaaaaaaabbddddd | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 6** | 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^
lucky^y ^charr^ms
ora^^nge^^^^
```

Output:

```
Hey google
luckycharms
-1
```

**Answer:** (penalty regime: 0 %)

```
1 def process_string(s):
2     stack=[]
3     for ch in s:
4         if ch=='^':
5             if stack:
6                 stack.pop()
7             else:
8                 stack.append(ch)
9     return ''.join(stack)if stack else "-1"
10 n=int(input())
11 for _ in range(n):
12     line=input().strip()
13     print(process_string(line))
```

|   | Input  | Expected                        | Got                             |   |
|---|--|---------------------------------|---------------------------------|---|
| ✓ | 3<br>Hey ^ goooo^^glee^<br>lucky^y ^charr^ms<br>bra^ng^e^^^^ | Hey google<br>luckycharms<br>-1 | Hey google<br>luckycharms<br>-1 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 7** | 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 s=input()
2 s=''.join(ch.lower() for ch in s if ch.isalnum())
3 print(1 if s==s[::-1]else 0)
```

|   | 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

**Question 8** | 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     paris={')':'(', '}':'{', ']':'['}
4
5     for ch in s:
6         if ch in "([{":
7             stack.append(ch)
8         else:
9             if not stack or stack[-1] !=paris[ch]:
10                 return "false"
11             stack.pop()
12     return "true" if not stack else "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.

**Question 9** | 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 | s=input()  
2 | print(s[::-1])
```

|   | <b>Input</b> | <b>Expected</b> | <b>Got</b> |   |
|---|--------------|-----------------|------------|---|
| ✓ | hello        | olleh           | olleh      | ✓ |
| ✓ | Hannah       | hannaH          | hannaH     | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**Question 10** | 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:

$1 \leq \text{sentence.length} \leq 1000$

sentence consists of lowercase English letters;

For example:

| Test  | Result |
|---|--------|
| <code>print(checkPangram('thequickbrownfoxjumpsoverthelazydog'))</code> | true   |
| <code>print(checkPangram('arvijayakumar'))</code>                       | false  |

Answer: (penalty regime: 0 %)

Reset answer

```
1 • def checkPangram(s):
2     return "true" if set(s) >= set("abcdefghijklmnopqrstuvwxyz") else "false"
```

|   | 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

**Started on** Tuesday, 9 September 2025, 6:19 PM

**State** Finished

**Completed on** Tuesday, 9 September 2025, 7:08 PM

**Time taken** 48 mins 49 secs

**Marks** 10.00/10.00

**Grade** 100.00 out of 100.00

**Question 1** | Correct · Mark 1.00 out of 1.00

Given an integer  $n$ , return a 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:**

$0 \rightarrow 0$

$1 \rightarrow 1$

$2 \rightarrow 10$

**Example2:**

**Input:**  $n = 5$

**Output:**  $[0, 1, 1, 2, 1, 2]$

**Explanation:**

$0 \rightarrow 0$

$1 \rightarrow 1$

$2 \rightarrow 10$

$3 \rightarrow 11$

$4 \rightarrow 100$

$5 \rightarrow 101$

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     ans=[]
3 •     for i in range(n+1):
4         ans.append(bin(i).count('1'))
5     return ans

```

| 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.

**Question 2** | 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  $A[\text{startIndex} \dots \text{endIndex}]$  (`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 range_update(arr_size,operations):
2     arr=[0]*arr_size
3     for op in operations:
4         start,end,inc=op
5         arr[start]+=inc
6         if end+1<arr_size:
7             arr[end+1]-=inc
8         for i in range(1,arr_size):
9             arr[i]+=arr[i-1]
10    return arr
11 arr_size=int(input())
12 k=int(input())
```

```
13 operations=[]
14 for _ in range(k):
15     operations.append(tuple(map(int,input().split())))
16 result=range_update(arr_size,operations)
17 print(*result)
```

|   | Input                              | Expected   | Got        |   |
|---|------------------------------------|------------|------------|---|
| ✓ | 5<br>3<br>1 3 2<br>2 4 3<br>8 2 -2 | -2 0 3 5 3 | -2 0 3 5 3 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | 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     | True   |
| 6     |        |
| 5     |        |
| 4     |        |
| 3     |        |

Answer: (penalty regime: 0 %)

```

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

```

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

|   | Input                 | Expected | Got   |   |
|---|-----------------------|----------|-------|---|
| ✓ | 4<br>9<br>6<br>4<br>2 | True     | True  | ✓ |
| ✓ | 3<br>2<br>1<br>4      | False    | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**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) and then return the  $p^{\text{th}}$  element of the list, sorted ascending. If there is no  $p^{\text{th}}$  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<br>3 | 5      |
| 10<br>5 | 0      |
| 1<br>1  | 1      |

**Answer:** (penalty regime: 0 %)

```

1 import math
2 def get_pth_factors(n,p):
3     factors=[]
4     for i in range(1,int(math.isqrt(n))+1):
5         if n%i==0:
6             factors.append(i)
7             if i!=n//i:
8                 factors.append(n//i)
9     factors.sort()
10    if p<=len(factors):
11        return factors[p-1]
12    else:
13        return 0
14 n=int(input())
15 p=int(input())
16 print(get_pth_factors(n,p))

```

|   | Input   | Expected | Got |   |
|---|---------|----------|-----|---|
| ✓ | 10<br>3 | 5        | 5   | ✓ |
| ✓ | 10<br>5 | 0        | 0   | ✓ |
| ✓ | 1<br>1  | 1        | 1   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 5** | 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     | 1      |
| 3     |        |
| 1     |        |
| 3     |        |
| 5     |        |
| 4     |        |

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

**Answer:** (penalty regime: 0 %)

```

1 def pair_with(arr,k):
2     n=len(arr)
3     i,j=0,1
4     while i<n and j<n:
5         diff=arr[j]-arr[i]
6         if diff==k and i!=j:
7             return 1
8         elif diff <k:
9             j+=1
10        else:
11            i+=1
12            if i==j:
13                j+=1
14    return 0
15 T=int(input())
16 for _ in range(T):
17     N=int(input())
18     arr=[]
19     for _ in range(N):
20         arr.append(int(input()))
21     k=int(input())
22     print(pair_with(arr,k))

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

**Question 6** | 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 n=int(input())  
2 arr=[]  
3 for _ in range(n):  
4     arr.append(int(input()))  
5 freq={}  
6 for num in arr:  
7     if num in freq:  
8         freq[num]+=1  
9     else:  
10        freq[num]=1  
11 printed=set()  
12 for num in arr:  
13     if num not in printed:  
14         print(f"{num} occurs {freq[num]} times")  
15         printed.add(num)
```

|   | 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    |                   |                   |   |
|   | 48    |                   |                   |   |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

**Question 7** | 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 3 4 |
| 1     |         |
| 2     |         |
| 2     |         |
| 3     |         |
| 4     |         |

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

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 arr=[int(input()) for _ in range(n)]
3 seen=set()
4 distinct=[]
5 for num in arr:
6     if num not in seen:
7         distinct.append(num)
8         seen.add(num)
9 print(*distinct)

```

|   | Input                           | Expected | Got     |   |
|---|---------------------------------|----------|---------|---|
| ✓ | 5<br>1<br>2<br>2<br>3<br>4      | 1 2 3 4  | 1 2 3 4 | ✓ |
| ✓ | 6<br>1<br>1<br>2<br>2<br>3<br>3 | 1 2 3    | 1 2 3   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



**Question 8** | 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*  $\text{maxes}[i]$  representing the total number of elements  $\text{nums}[j]$  satisfying  $\text{nums}[j] \leq \text{maxes}[i]$  where  $0 \leq j < n$  and  $0 \leq i < m$ , in the given order.

The program has the following:

$\text{nums}[\text{nums}[0], \dots, \text{nums}[n-1]]$ : first array of positive integers  
 $\text{maxes}[\text{maxes}[0], \dots, \text{maxes}[n-1]]$ : second array of positive integers

**Constraints**

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

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

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

The next  $m$  lines each contain an integer describing  $\text{maxes}[i]$  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  $n = 4$ ,  $\text{nums} = [1, 4, 2, 4]$ ,  $m = 2$ , and  $\text{maxes} = [3, 5]$ .

1. For  $\text{maxes}[0] = 3$ , we have 2 elements in  $\text{nums}$  ( $\text{nums}[0] = 1$  and  $\text{nums}[2] = 2$ ) that are  $\leq \text{maxes}[0]$ .
2. For  $\text{maxes}[1] = 5$ , we have 4 elements in  $\text{nums}$  ( $\text{nums}[0] = 1$ ,  $\text{nums}[1] = 4$ ,  $\text{nums}[2] = 2$ , and  $\text{nums}[3] = 4$ ) that are  $\leq \text{maxes}[1]$ .

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,  $n = 5$ ,  $\text{nums} = [2, 10, 5, 4, 8]$ ,  $m = 4$ , and  $\text{maxes} = [3, 1, 7, 8]$ .

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

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

**Answer:** (penalty regime: 0 %)

```
1 import bisect
2 def count_less_equal(nums,maxes):
3     nums.sort()
4     result=[]
5     for x in maxes:
6         count=bisect.bisect_right(nums,x)
7         result.append(count)
8     return result
9 n=int(input())
10 nums=[]
11 for _ in range(n):
12     nums.append(int(input()))
13 m=int(input())
14 maxes=[]
```

```
15 • for _ in range(m):
16     maxes.append(int(input()))
17 output=count_less_equal(nums,maxes)
18 • for count in output:
19     print(count)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 9** | 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):**

$1 \leq N \leq 10^4$

$-99999 \leq \text{Array Element Value} \leq 99999$

**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=input().strip().split()
2 N=int(N_K[0])
3 K=int(N_K[1])
4 nums=list(map(int,input().strip().split()))
5 result=11
```

```
6 for i in range(0,N,K):
7     group=nums[i:i+K]
8     group.sort(reverse=True)
9     result.extend(group)
10 print(*result)
```

|   | Input                       | Expected             | Got                  |   |
|---|-----------------------------|----------------------|----------------------|---|
| ✓ | 7 3<br>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.

**Question 10** | 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:**

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

**Answer:** (penalty regime: 0 %)

```
1 n,m=map(int,input().split())  
2 mat=[]  
3 for _ in range(n):  
4     mat.append(list(map(int,input().split())))  
5 common=set(mat[0])  
6 for i in range(1,n):  
7     common &= set(mat[i])  
8 if common:  
9     print(min(common))  
10 else:  
11     print(-1)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Started on** Monday, 22 September 2025, 6:58 PM

**State** Finished

**Completed on** Wednesday, 24 September 2025, 9:00 PM

**Time taken** 2 days 2 hours

**Marks** 10.00/10.00

**Grade** 100.00 out of 100.00

**Question 1** | 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 n=input()
2 s=set(n)
3 if s.issubset({'0','1'}):
4     print("Yes")
5 else:
6     print("No")

```

|   | 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.



**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     c=0
4     while nums:
5         if nums[0]==min(nums):
6             nums=nums[1:]
7         else:
8             nums=nums[1:]+(nums[0],)
9         c+=1
10    return c
11    pass

```

| 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.

**Question 3** | 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:**

55 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<br>3 | 1      |
| 1,2<br>8       | 0      |

**Answer:** (penalty regime: 0 %)

```

1 t=tuple(map(int,input().split(',')))
2 k=int(input())
3 pairs=set()
4 for i in range(len(t)):
5     for j in range(i+1,len(t)):
6         if t[i]+t[j]==k:
7             pairs.add(tuple(sorted((t[i],t[j]))))
8 print(len(pairs))
9

```

|   | Input             | Expected | Got |   |
|---|-------------------|----------|-----|---|
| ✓ | 5,6,5,7,7,8<br>13 | 2        | 2   | ✓ |
| ✓ | 1,2,1,2,5<br>3    | 1        | 1   | ✓ |

|   | Input     | Expected | Got |   |
|---|-----------|----------|-----|---|
| ✓ | 1, 2<br>8 | 8        | 8   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 4** | 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 a=input()
2 a=[int(x) for x in a.split()]
3 seen=set()
4 for n in a:
5     if n in seen:
6         print(n)
7         break
8     seen.add(n)
9

```

|   | 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.

**Question 5** | 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 = "AAAAAACCCCCAAAAACCCCCAAAAAGGGTTT"

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

**Example 2:**

**Input:** s = "AAAAAAAAAAAAAA"

**Output:** ["AAAAAAAAAA"]

**For example:**

| Input                            | Result                   |
|----------------------------------|--------------------------|
| AAAAAACCCCCAAAAACCCCCAAAAAGGGTTT | AAAAACCCCC<br>CCCCCAAAAA |

**Answer:** (penalty regime: 0 %)

```

1 s=input()
2 result=[]
3 for i in range(len(s)-9):
4     seq=s[i:i+10]
5     if seq in s[i+1:] and seq not in result:
6         result.append(seq)
7 for i in result:
8     print(i)

```

|   | Input                           | Expected                | Got                     |   |
|---|---------------------------------|-------------------------|-------------------------|---|
| ✓ | AAAAACCCCCAAAAACCCCCAAAAAGGGTTT | AAAAACCCC<br>CCCCAAAAAA | AAAAACCCC<br>CCCCAAAAAA | ✓ |
| ✓ | AAAAAAAAAAAAAA                  | AAAAAAAAAA              | AAAAAAAAAA              | ✓ |

Passed all tests! ✓

Correct

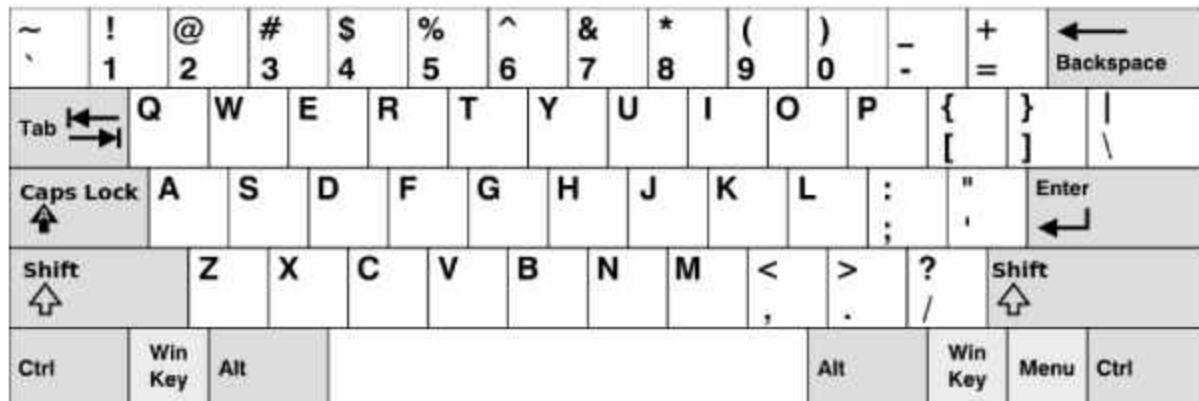
Marks for this submission: 1.00/1.00.

**Question 6** | 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".

**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      | Alaska |
| Hello  | Dad    |
| Alaska |        |
| Dad    |        |
| Peace  |        |
| 2      | adsfd  |
| adsfd  | afd    |
| afd    |        |

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 l=[]
3 for i in range(n):
```

```

4     l.append(input())
5     s1='qwertyuiop'
6     s2='asdfghjkl'
7     s3='zxcvbnm'
8     result=[]
9     for i in l:
10        w=i.lower()
11        if all(c in s1 for c in w) or all(c in s2 for c in w) or all(c in s3 for c in w):
12            result.append(i)
13    if result:
14        for i in result:
15            print(i)
16    else:
17        print('No words')
18

```

|   | <b>Input</b>                         | <b>Expected</b> | <b>Got</b>    |   |
|---|--------------------------------------|-----------------|---------------|---|
| ✓ | 4<br>Hello<br>Alaska<br>Dad<br>Peace | Alaska<br>Dad   | Alaska<br>Dad | ✓ |
| ✓ | 1<br>omk                             | No words        | No words      | ✓ |
| ✓ | 2<br>adsfd<br>afd                    | adsfd<br>afd    | adsfd<br>afd  | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 7** | 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 5 10           |
| 1 2 8 6 5 | 3                |
| 2 6 8 10  |                  |
| 5 5       | NO SUCH ELEMENTS |
| 1 2 3 4 5 |                  |
| 1 2 3 4 5 |                  |

**Answer:** (penalty regime: 0 %)

```
1 n,m=map(int,input().split())
2 a1=list(map(int,input().split()))
3 a2=list(map(int,input().split()))
4 c=set(a1)^set(a2)
5 z=len(c)
6 if c:
7     print(' '.join(map(str,c)))
8     print(z)
9 else:
10    print("NO SUCH ELEMENTS")
```

|   | Input                         | Expected                             | Got                                  |   |
|---|-------------------------------|--------------------------------------|--------------------------------------|---|
| ✓ | 5 4<br>1 2 8 6 5<br>2 6 8 10  | 1 5 10<br>3<br>3                     | 1 5 10<br>3                          | ✓ |
| ✓ | 3 3<br>10 10 10<br>10 11 12   | 11 12<br>2<br>2                      | 11 12<br>2                           | ✓ |
| ✓ | 5 5<br>1 2 3 4 5<br>1 2 3 4 5 | NO SUCH ELEMENTS<br>NO SUCH ELEMENTS | NO SUCH ELEMENTS<br>NO SUCH ELEMENTS | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 8** | 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<br>ad                              | 1      |
| Faculty Upskilling in Python Programming<br>ak | 2      |

Answer: (penalty regime: 0 %)

```

1 w=input()
2 w=w.replace(' ','')
3 w=set(w)
4 w=str(w)
5 b=set(input())
6 c=0
7 for ch in w:
8     if ch in b:
9         c+=1
10 print(c)

```

|   | Input               | Expected | Got |   |
|---|---------------------|----------|-----|---|
| ✓ | hello world<br>ad   | 1        | 1   | ✓ |
| ✓ | Welcome to REC<br>R | 1        | 1   | ✓ |

|   | Input  | Expected | Got |  |
|---|--|----------|-----|--|
| ✓ | Faculty Upskilling in Python Programming<br>ak | 2        | 2 ✓ |  |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 9** | 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 3 4 |
| 1     |         |
| 2     |         |
| 2     |         |
| 3     |         |
| 4     |         |

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 s=set()
3 for i in range (n):
4     p=int(input())
5     s.add(p)
6 print(*s)
7
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 10** | 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<br>mango orange<br>mango    | yes<br>set3 is subset of set1 and set2 |
| 2    | mango orange<br>banana orange<br>grapes | No                                     |

**Answer:** (penalty regime: 0 %)

```

1 s1=set(input())
2 s2=set(input())
3 s3=set(input())
4 if s3.issubset(s1) or s3.issubset(s2):
5     print("yes")
6     print("set3 is subset of set1 and set2\n")
7 else:
8     print("No")
9

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Started on** Tuesday, 7 October 2025, 8:34 PM

**State** Finished

**Completed on** Tuesday, 7 October 2025, 9:25 PM

**Time taken** 50 mins 14 secs

**Marks** 5.00/5.00

**Grade** 100.00 out of 100.00

**Question 1** | 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**

$1 \leq \text{num} \leq 4294967295$

**Sample Input 1**

26

**Sample Output 1**

Z

For example:

| Test                                | Result |
|-------------------------------------|--------|
| <code>print(excelNumber(26))</code> | Z      |

**Answer:** (penalty regime: 0 %)

[Reset answer](#)

```
1 • def excelNumber(n):  
2     result=""  
3 •     while n>0:  
4         n-=1  
5         result=chr(n%26+ord('A'))+result  
6         n/=26  
7     return result
```

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

Passed all tests! ✓

Correct!

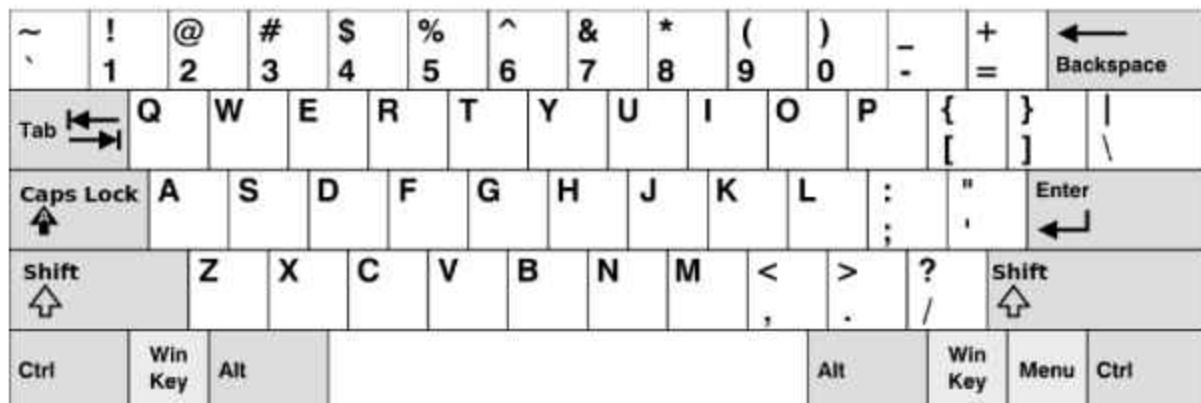
Marks for this submission: 1.00/1.00.

**Question 2** 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".

**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      | Alaska |
| Hello  | Dad    |
| Alaska |        |
| Dad    |        |
| Peace  |        |
| 2      | adsdf  |
| adsfd  | afd    |
| afdf   |        |

**Answer:** (penalty regime: 0 %)

```
1 n=int(input())
2 l=[]
3 for i in range(n):
```

```

4     l.append(input())
5     s1='qwertyuiop'
6     s2='asdfghjkl'
7     s3='zxcvbnm'
8     result=[]
9     for i in l:
10        w=i.lower()
11        if all(c in s1 for c in w) or all(c in s2 for c in w) or all(c in s3 for c in w):
12            result.append(i)
13    if result:
14        for i in result:
15            print(i)
16    else:
17        print('No words')

```

|   | <b>Input</b>                         | <b>Expected</b> | <b>Got</b>    |   |
|---|--------------------------------------|-----------------|---------------|---|
| ✓ | 4<br>Hello<br>Alaska<br>Dad<br>Peace | Alaska<br>Dad   | Alaska<br>Dad | ✓ |
| ✓ | 1<br>omk                             | No words        | No words      | ✓ |
| ✓ | 2<br>adsfd<br>afd<br>afd             | adsfd<br>afd    | adsfd<br>afd  | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | 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<br>ad                              | 1      |
| Faculty Upskilling in Python Programming<br>ak | 2      |

Answer: (penalty regime: 0 %)

```

1 w=input()
2 w=w.replace(' ','')
3 w=str(set(w))
4 b=set(input())
5 c=0
6 for ch in w:
7     if ch in b:
8         c+=1
9 print(c)

```

|   | Input               | Expected | Got |   |
|---|---------------------|----------|-----|---|
| ✓ | hello world<br>ad   | 1        | 1   | ✓ |
| ✓ | Welcome to REC<br>R | 1        | 1   | ✓ |

|   | Input  | Expected | Got |  |
|---|--|----------|-----|--|
| ✓ | Faculty Upskilling in Python Programming<br>ak | 2        | 2 ✓ |  |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 4** | Correct | Mark 1.00 out of 1.00

The program must accept a string S which represents a series of keystrokes as the input. There are two special keystrokes denoted by "<sup>z</sup>" for undo operation and "<sup>y</sup>" for redo operation. The undo operation removes the last typed character and redo operations reverses the immediately done last undo operations. The program must accept the keystrokes and perform all the operations (undo and redo). Finally, the program must print the string after applying all the operations as the output.

**Boundary Condition(s):**

1 <= Length of string <= 1000

**Input Format:**

The first line contains the string S.

**Output Format:**

The first line contains the modified string.

**Example Input/Output 1:****Input:**

missel<sup>z</sup><sup>z</sup><sup>z</sup>pell<sup>z</sup><sup>z</sup><sup>y</sup>t

**Output:**

misspelt

**Explanation:**

The characters **missel** are followed by three undo operations which deletes last three characters.

So the string **missel** becomes **mis**.

The three undo operations are followed by one redo operation which reverses the last undo operation.

So the string **mis** becomes **miss**.

Then the characters **pell** are entered which are followed by two undo operations so the last two characters are removed.

So the string becomes **misspe**.

Then a redo operation is applied which reverses the last undo operation and so the string **misspe** becomes **misspel**.

The characters are followed by **t** so the final string is misspelt.

**Answer: (penalty regime: 0 %)**

```
1 s=input()
2 n=len(s)
3 l=| |
```

```
4 k=[]
5 i=0
6 while(i<n):
7     if(s[i:i+2]=='^z'):
8         if(l):
9             k.append(l.pop())
10        i+=2
11    elif s[i:i+2]=='^y':
12        if k:
13            l.append(k.pop())
14        i+=2
15    else:
16        l.append(s[i])
17        i+=1
18 print(''.join(l))
```

|   | Input                      | Expected | Got      |   |
|---|----------------------------|----------|----------|---|
| ✓ | missel^z^z^z^typell^z^z^yt | misspelt | misspelt | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 5** | 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=input().split()
2 r=[""]*len(s)
3 for word in s:
4     po=int(word[-1])-1
5     r[po]=word[:-1]
6 print(" ".join(r))

```

|   | 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.

**Started on** Tuesday, 7 October 2025, 12:37 PM

**State** Finished

**Completed on** Thursday, 9 October 2025, 10:41 PM

**Time taken** 2 days 10 hours

**Marks** 10.00/10.00

**Grade** 100.00 out of 100.00

**Question 1** | 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=input().split()
2 r=[""]*len(s)
3 for word in s:
4     po=int(word[-1])-1
5     r[po]=word[:-1]
6 print(" ".join(r))

```

|   | 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.

**Question 2** | 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**

$1 \leq \text{num} \leq 4294967295$

**Sample Input 1**

26

**Sample Output 1**

Z

For example:

| Test                                | Result |
|-------------------------------------|--------|
| <code>print(excelNumber(26))</code> | Z      |

**Answer:** (penalty regime: 0 %)

[Reset answer](#)

```
1 • def excelNumber(n):
2     result=''
3 •     while n>0:
4         n-=1
5         result=chr((n%26)+ord('A'))+result
6         n/=26
7     return result
8
```

| 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.

**Question 3** | 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 p={"A":1,"E":1,"I":1,"L":1,"N":1,"O":1,"R":1,"S":1,"T":1,"U":1,
2 "D":2,"G":2,
3 "B":3,"C":3,"H":3,"P":3,
4 "F":4,"H":4,"V":4,"W":4,"Y":4,
5 "K":5,
6 "J":8,"X":8,
7 "Q":10,"Z":10
8 }
9 word=input().upper()
10 score=0
11 for ch in word:
12     score+=p.get(ch,0)
13 print(word,"is worth",score,"points.")
14

```

|   | 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.

**Question 4** | 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     # Your implementation here  
3     lower=set()  
4     upper=set()  
5     for ch in word:  
6         if ch.islower():  
7             lower.add(ch)  
8         elif ch.isupper():  
9             upper.add(ch.lower())  
10    return len(lower&upper)  
11    pass
```

| 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.

**Question 5** | 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 s1 and s2, return a list of all the uncommon words. You may return the answer in any order.

Example 1:

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

Output: ["sweet", "sour"]

Example 2:

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

Output: ["banana"]

Constraints:

$1 \leq s1.length, s2.length \leq 200$

s1 and s2 consist of lowercase English letters and spaces.

s1 and s2 do not have leading or trailing spaces.

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

Note:

Use dictionary to solve the problem

For example:

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

Answer: (penalty regime: 0 %)

```

1 s1=input()
2 s2=input()
3 d={}
4 for w in (s1+" "+s2).split():
5     d[w]=d.get(w,0)+1
6 print(" ".join([w for w,c in d.items() if c==1]))
7

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 6** | 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 d={}
3 for i in range(n):
4     s=input().split()
5     d[s[0]]=list(map(int,s[1:4]))
6 lname=[]
7 hname=[]
8 maxass=[]
9 minlab=[]
10 la=101
```

```

11 ha=-1
12 ma=-1
13 ml=101
14 for i,j in d.items():
15     avg=sum(j)/3
16     t,a,l=j
17     if(avg>ha):
18         ha=avg
19         hname=[i]
20     elif avg==ha:
21         hname.append(i)
22     if(avg<la):
23         la=avg
24         lname=[i]
25     elif avg==la:
26         lname.append(i)
27     if a>ma:
28         ma=a
29         maxass=[i]
30     elif(a==ma):
31         maxass.append(i)
32     if(i<ml):
33         ml=i
34         minlab=[i]
35     elif l==ml:
36         minlab.append(i)
37 print(''.join(sorted(hname)))
38 print(''.join(sorted(maxass)))
39 print(''.join(sorted(minlab)))
40 print(''.join(sorted(lname)))
41
42
43

```

|   | <b>Input</b>  | <b>Expected</b>                      | <b>Got</b>                           |   |
|---|---|--------------------------------------|--------------------------------------|---|
| ✓ | 4<br>James 67 89 56<br>Lalith 89 45 45<br>Ram 89 89 89<br>Sita 70 70 70 | Ram<br>James Ram<br>Lalith<br>Lalith | Ram<br>James Ram<br>Lalith<br>Lalith | ✓ |
| ✓ | 3<br>Raja 95 67 90<br>Aarav 89 90 90<br>ShadhanA 95 95 91               | ShadhanA<br>ShadhanA<br>Aarav Raja   | ShadhanA<br>ShadhanA<br>Aarav Raja   | ✓ |
|   |   | Raja                                 | Raja                                 |   |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 7** | 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 17  |
| Gfg 6 7 4  | Best 18 |
| Best 7 6 5 |         |

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 d={}
3 for _ in range(n):
4     parts=input().split()
5     key=parts[0]
6     values=list(map(int,parts[1:]))
7     d[key]=values
8 d_sum={k:sum(v) for k,v in d.items()}
9 for k,v in sorted(d_sum.items(),key=lambda x:x[1]):
10    print(k,v)

```

|   | Input                        | Expected          | Got               |   |
|---|------------------------------|-------------------|-------------------|---|
| ✓ | 2<br>Gfg 6 7 4<br>Best 7 6 5 | Gfg 17<br>Best 18 | Gfg 17<br>Best 18 | ✓ |
| ✓ | 2<br>Gfg 6 6<br>Best 5 5     | Best 10<br>Gfg 12 | Best 10<br>Gfg 12 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 8** | 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<br>e = 1<br>i = 0<br>o = 2<br>u = 0 |
| Python      | a = 0<br>e = 0<br>i = 0<br>o = 1<br>u = 0 |

**Answer:** (penalty regime: 0 %)

```
1 text=input()
2 text=text.lower()
3 vowels="aeiou"
4 for v in vowels:
5     print(v,"=",text.count(v))
6
7
8
9
```

|   | Input                      | Expected                                  | Got                                       |   |
|---|----------------------------|---|---|---|
| ✓ | Hello World                | a = 0<br>e = 1<br>i = 0<br>o = 2<br>u = 0 | a = 0<br>e = 1<br>i = 0<br>o = 2<br>u = 0 | ✓ |
| ✓ | AEIOU aeio u               | a = 2<br>e = 2<br>i = 2<br>o = 2<br>u = 2 | a = 2<br>e = 2<br>i = 2<br>o = 2<br>u = 2 | ✓ |
| ✓ | Python                     | a = 0<br>e = 0<br>i = 0<br>o = 1<br>u = 0 | a = 0<br>e = 0<br>i = 0<br>o = 1<br>u = 0 | ✓ |
| ✓ | abcdefghijklmnopqrstuvwxyz | a = 1<br>e = 1<br>i = 1<br>o = 1<br>u = 1 | a = 1<br>e = 1<br>i = 1<br>o = 1<br>u = 1 | ✓ |
| ✓ | 12345!@#\$%^&*AEIOU        | a = 1<br>e = 1<br>i = 1<br>o = 1<br>u = 1 | a = 1<br>e = 1<br>i = 1<br>o = 1<br>u = 1 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 9** | 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:

@#\$%^.^&amp;

For example:

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

Answer: (penalty regime: 0 %)

```

1 n=input()
2 d={'0':'!','1':'@','2':'#','3':'$','4':'%','5':'^','6':'&','7':'*','8':">>','9': '<'}
3 for i in n:
4     if i!=.:
5         print(d[i],end='')
6     elif i==.:
7         print('.',end='')
8 if '.' not in n:
9     print('!.!!')

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 10** | 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  
Johny  
Jamie  
Jamie  
Johny  
Jack  
Johny  
Johny  
Jackie

**Sample Output:**

Johny

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

**Answer:** (penalty regime: 0 %)

```

1 n=int(input())
2 votes=[input() for _ in range(n)]
3 count={}
4 for v in votes: count[v]=count.get(v,0)+1
5 print(sorted([k for k,c in count.items() if c==max(count.values()))])[0])

```

|   | Input  | Expected | Got     |  |
|---|--|----------|---------|--|
| ✓ | 10<br>John<br>John<br>Johny<br>Jamie<br>Jamie<br>Johny<br>Jack<br>Johny<br>Johny<br>Jackie | Johny    | Johny ✓ |  |
| ✓ | 6<br>Ida<br>Ida<br>Ida<br>Kiruba<br>Kiruba<br>Kiruba                                       | Ida      | Ida ✓   |  |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Started on** Friday, 10 October 2025, 3:54 PM

**State** Finished

**Completed on** Monday, 13 October 2025, 10:50 PM

**Time taken** 3 days 6 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 Python program to count the frequency of each word in a given text file.

**Description:****1. Input:**

- o String as input.

**2. Output:**

- o 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                                  |
|---|---|---|
| <pre>with open('output.txt', 'r') as file:     text = file.read()     print(text)</pre> | <pre>apple orange apple banana apple orange</pre> | <pre>apple: 3 banana: 1 orange: 2</pre> |

**Answer:** (penalty regime: 0 %)

```

1 k=input()
2 s=""
3 l=[]
4 for i in k:
5     if(i not in ".!?"):
6         s+=i
7 s=s.lower()
8 l=s.split()
9
10 d={}
11 for i in l:
12     if i not in d:
13         d[i]=1
14     else:
15         d[i]+=1
16 d=dict(sorted(d.items()))
17 with open('output.txt','w')as f:
18     for i,j in d.items():
19         f.write(i)
20         f.write(':')
21         f.write(' ')
22         f.write(str(j))
23         f.write("\n")
24

```

| Test   | Input  | Expected   | Got  |   |
|--|--|--|--|---|
| ✓ with open('output.txt', 'r') as file:<br>text = file.read()<br>print(text) | apple orange apple banana apple orange                               | apple: 3<br>banana: 1<br>orange: 2   | apple: 3<br>banana: 1<br>orange: 2   | ✓ |
| ✓ with open('output.txt', 'r') as file:<br>text = file.read()<br>print(text) | Hello world! Hello everyone. Welcome to the world of<br>programming. | everyone: 1<br>hello: 2<br>of: 1<br>programming:<br>i<br>the: 1<br>to: 1<br>welcome: 1<br>world: 2 | everyone: 1<br>hello: 2<br>of: 1<br>programming:<br>i<br>the: 1<br>to: 1<br>welcome: 1<br>world: 2 | ✓ |
| ✓ with open('output.txt', 'r') as file:<br>text = file.read()<br>print(text) | One fish two fish Red fish blue fish                                 | blue: 1<br>fish: 4<br>one: 1<br>red: 1<br>two: 1   | blue: 1<br>fish: 4<br>one: 1<br>red: 1<br>two: 1   | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 2** | 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:**

- o A text file containing multiple words.

**2. Output:**

- o 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:<br>text = file.read()<br>print(text) | input1.txt | madam<br>arora<br>malayalam |

**Answer:** (penalty regime: 0 %)

```

1 n=input()
2 with open(n,'r')as f:
3     s=f.read().split()
4 with open('output.txt','w')as f1:
5     for i in s:
6         if(i==i[::-1]):
7             f1.write(i)
8             f1.write('\n')

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 3** | 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:**

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

**2. Output:**

- o 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:<br>text = file.read()<br>print(text) | input1.txt<br>2 | Line one.<br>Updated line two.<br>Line three.<br>Line four. |

**Answer:** (penalty regime: 0 %)

```

1 n=input()
2 n1=int(input())
3 a=input()
4 with open(n,'r')as f:
5     s=f.readlines()
6     m=n1-1
7     if 0<=m<len(s):
8         s[m]=a+'\n'
9 with open('output.txt','w')as f1:
10    f1.writelines(s)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 4** | 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:<br>text = file.read()<br>print(text)<br>3 | input1.txt | Line one.<br>Line two.<br>eerht enil.<br>Line four. |

**Answer:** (penalty regime: 0 %)

```

1 a=input()
2 n=int(input())
3 with open(a,'r')as f:
4     s=f.readlines()
5     s[n-1]=s[n-1][:-2][::-1]+'\'\n'
6 with open("output.txt",'w')as f1:
7     f1.writelines(s)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 5** | 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:**

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

**2. Output:**

- o The content of the specified line.

input1.txt:

Line one.

Line two.

Line three.

Line four.

**For example:**

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

**Answer:** (penalty regime: 0 %)

```

1: n=input()
2: pos=int(input())
3: f=open(n,'r')
4: s=f.readlines()
5: print(s[pos-1])

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 6** | 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:**

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

**2. Output:**

- o 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:<br>text = file.read()<br>print(text) | input1.txt<br>2 | Line one.<br>Line three.<br>Line four. |

**Answer:** (penalty regime: 0 %)

```

1 n=input()
2 n1=int(input())
3 with open(n,'r')as f:
4     a=f.readlines()
5     a.pop(n1-1)
6 with open('output.txt','w')as f1:
7     f1.writelines(a)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

**Question 7** | 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: containing |

**Answer:** (penalty regime: 0 %)

```

1 n=input()
2 f=open(n,'r')
3 s=f.read().split()
4 l=''
5 for i in s:
6     if( len(i)>len(l)):
7         l=i
8 print("Longest word:",l)
9 f.close()
```

|   | Input      | Expected   | Got  |   |
|---|------------|--|--|---|
| ✓ | input1.txt | Longest word: containing                         | Longest word: containing                         | ✓ |
| ✓ | 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.

**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:**

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

**2. Output:**

- o The total number of words in the file.

**For example:**

| Input      | Result          |
|------------|-----------------|
| input2.txt | Total words: 14 |
| input3.txt | Total words: 15 |

**Answer:** (penalty regime: 0 %)

```

1 n=input()
2 f=open(n,'r')
3 s=f.read().split()
4 c=0
5 for i in s:
6     c+=1
7 print("Total words:",c)

```

|   | Input      | Expected        | Got             |   |
|---|------------|-----------------|-----------------|---|
| ✓ | input1.txt | Total words: 6  | Total words: 6  | ✓ |
| ✓ | input2.txt | Total words: 14 | Total words: 14 | ✓ |
| ✓ | input3.txt | Total words: 15 | Total words: 15 | ✓ |

Passed all tests! ✓

**CORRECT**

Marks for this submission: 1.00/1.00.

**Question 9** | 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:**

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

**2. Output:**

- o 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:<br>text = file.read()<br>print(text)<br>3<br>Inserted line. | input1.txt | Line one.<br>Line two.<br>Inserted line.<br>Line three.<br>Line four. |

**Answer:** (penalty regime: 0 %)

```
1 txt=input()  
2 f=open(txt,'r')  
3 r=f.readlines()  
4 r[-1]+='\n'  
5 pos=int(input())  
6 val=input()  
7 r.insert(pos-1,val)  
8 r.insert(pos,'\n')  
9 with open('output.txt','w') as f1:  
10     f1.writelines(r)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**Question 10** | 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   |
|--|---------------------------|--|
| <pre>with open('output1.txt', 'r') as file:     text = file.read()     print(text)</pre> | input1.txt<br>output1.txt | This is the source file.<br>It contains multiple lines of text.<br>Here is another line. |

**Answer:** (penalty regime: 0 %)

```
1 n1=input()
2 n2=input()
3 with open(n1,'r')as f:
4     text=f.read()
5 with open(n2,'w')as f1:
6     f1.write(text)
```

|   | Test   | Input                     | Expected   | Got  |   |
|---|--|---------------------------|--|--|---|
| ✓ | <pre>with open('output1.txt', 'r') as file:     text = file.read()     print(text)</pre> | input1.txt<br>output1.txt | This is the source file.<br>It contains multiple lines of         text.<br>Here is another line.           | This is the source file.<br>It contains multiple lines of         text.<br>Here is another line.           | ✓ |
| ✓ | <pre>with open('output2.txt', 'r') as file:     text = file.read()     print(text)</pre> | input2.txt<br>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. | ✓ |

| Test  | Input      | Expected                     | Got          |   |
|---|------------|------------------------------|--------------|---|
| ✓ with open('output3.txt', 'r') as file:<br>text = file.read()<br>print(text) | input3.txt | Single line.<br>output3.txt. | Single line. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.