

Started on Saturday, 16 August 2025, 10:00 PM

State Finished

Completed on Saturday, 16 August 2025, 10:59 PM

Time taken 59 mins 15 secs

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 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 rem=(a-2000)%12
3 if(rem == 0):
4     ani="Dragon"
5 elif(rem == 1):
6     ani="Snake"
7 elif(rem == 2):
8     ani="Horse"
9 elif(rem == 3):
10    ani="Sheep"
11 elif(rem == 4):
12    ani="Monkey"
13 elif(rem == 5):
14    ani="Rooster"
15 elif(rem == 6):
16    ani="Dog"
17 elif(rem == 7):
18    ani="Pig"
19 elif(rem == 8):
20    ani="Rat"
21 elif(rem == 9):
22    ani="Ox"
23 elif(rem == 10):
24    ani="Tiger"
25 else:
26    ani="Hare"
27 print(a,"is the year of the",ani+".")

28
29
30
```

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

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

```

| | Input | Expected | Got | |
|---|--------------|------------------|------------------|---|
| ✓ | 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! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 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 c=0
2 for i in range(5):
3     a=int(input())
4     if(a%2!=0):
5         c+=1
6 print(c)
7

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 12 | 3 | 3 | ✓ |
| | 17 | | | |
| | 19 | | | |
| | 14 | | | |
| | 115 | | | |

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 15 | 2 | 2 | ✓ |
| | 0 | | | |
| | -12 | | | |
| | 19 | | | |
| | 28 | | | |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

IN / OUT

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

Input Format:

Input consists of 2 integers.

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

Output Format:

Output consists of the string "IN" or "OUT".

Sample Input and Output:

Input

8

3

Output

OUT

For example:

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

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=int(input())
3 if(b>=a/2):
4     print("IN")
5 else:
6     print("OUT")

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

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

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

Sample Input 1

60

60

60

Sample Output 1

That's a equilateral triangle

Sample Input 2

40

40

80

Sample Output 2

That's a isosceles triangle

Sample Input 3

50

60

70

Sample Output 3

That's a scalene triangle

For example:

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

Answer: (penalty regime: 0 %)

```

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

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00

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

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

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

Sample Test Cases

Test Case 1

Input

50

Output

100.00

Test Case 2

Input

300

Output

517.50

For example:

| Input | Result |
|--------|---------|
| 100.00 | 120.00 |
| 500 | 1035.00 |

Answer: (penalty regime: 0 %)

```

1 a=float(input())
2 if(a<=199):
3     b=a*1.20
4 elif(a>=200 and a<400):
5     b=a*1.50
6 elif(a>=400 and a<600):
7     b=a*1.80
8 else:
9     b=a*2.00
10 if b>400:
11     b+=b*0.15
12 if b<100:
13     b=100
14 print(format(b,".2f"))
15

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00

Most years have 365 days. However, the time required for the Earth to orbit the Sun is actually slightly more than that. As a result, an extra day, February 29, is included in some years to correct for this difference. Such years are referred to as leap years. The rules for determining whether or not a year is a leap year follow:

- Any year that is divisible by 400 is a leap year.
- Of the remaining years, any year that is divisible by 100 is not a leap year.
- Of the remaining years, any year that is divisible by 4 is a leap year.
- All other years are not leap years.

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

Sample Input 1

1900

Sample Output 1

1900 is not a leap year.

Sample Input 2

2000

Sample Output 2

2000 is a leap year.

Answer: (penalty regime: 0 %)

```

1 y=int(input())
2 if(y%400==0):
3     print(y,"is a leap year.")
4 elif(y%100==0):
5     print(y,"is not a leap year.")
6 elif(y%4==0):
7     print(y,"is a leap year.")
8 else:
9     print(y,"is not a leap year.")
```

| | Input | Expected | Got | |
|---|-------|--------------------------|--------------------------|---|
| ✓ | 1900 | 1900 is not a leap year. | 1900 is not a leap year. | ✓ |
| ✓ | 2000 | 2000 is a leap year. | 2000 is a leap year. | ✓ |
| ✓ | 2100 | 2100 is not a leap year. | 2100 is not a leap year. | ✓ |
| ✓ | 2020 | 2020 is a leap year. | 2020 is a leap year. | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00

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

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

Sample Input Format:

```
11
+
14
```

Sample Output Format:

```
25
```

Answer: (penalty regime: 0 %)

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

| | Input | Expected | Got | |
|---|----------------|----------|------|---|
| ✓ | 11 + 14 | 25 | 25 | ✓ |
| ✓ | 45 - 50 | -5 | -5 | ✓ |
| ✓ | 12 * 100 | 1200 | 1200 | ✓ |
| ✓ | 18 / 2 | 9.0 | 9.0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 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 n=int(input())
3 a=n+1
4 r=int(math.sqrt(a))
5 if(r*r==a):
6     print("Yes")
7 else:
8     print("No")

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 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 if(a>50 and b<0.7 and c>5600):
5     print(10)
6 elif(a>50 and b<0.7):
7     print(9)
8 elif(b<0.7 and c>5600):
9     print(8)
10 elif(a>50 and c>5600):
11     print(7)
12 elif(a>50 or b<0.7 or c>5600):
13     print(6)
14 else:
15     print(5)
16
```

| | Input | Expected | Got | |
|---|-------------------|----------|-----|---|
| ✓ | 53 0.6 5602 | 10 | 10 | ✓ |
| ✓ | 45 0 4500 | 6 | 6 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Saturday, 16 August 2025, 11:01 PM

State Finished

Completed on Saturday, 16 August 2025, 11:50 PM

Time taken 49 mins 12 secs

Marks 5.00/5.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 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 5.1 | 81.67 sq.units |
| 2 4.3 | 26.23 sq.units |
| 6.1 | |

Answer: (penalty regime: 0 %)

```

1 c=int(input())
2 match c:
3     case 1:
4         r=float(input())
5         a=3.14*r*r
6         a=round(a+1e-8,2)
7         print("%.2f sq.units"%a)
8     case 2:
9         l=float(input())
10        b=float(input())
11        a=l*b
12        print(format(a,".2f")+" sq.units")
13    case 3:
14        p=float(input())
15        q=float(input())
16        a=0.5*p*q
17        print(format(a,".2f")+" sq.units")
18    case _:
19        print("Invalid choice")

```

| | Input | Expected | Got | |
|---|-----------------|-----------------|-----------------|---|
| ✓ | 1 5.1 | 81.67 sq.units | 81.67 sq.units | ✓ |
| ✓ | 2 4.3 6.1 | 26.23 sq.units | 26.23 sq.units | ✓ |
| ✓ | 3 3.7 5.6 | 10.36 sq.units | 10.36 sq.units | ✓ |
| ✓ | 4 | Invalid choice | Invalid choice | ✓ |
| ✓ | 1 9.5 | 283.39 sq.units | 283.39 sq.units | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 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 \leq 120 mm Hg and Diastolic \leq 80 mm Hg
3. Prehypertension: Systolic \leq 140 mm Hg and Diastolic \leq 90 mm Hg
4. Stage 1 Hypertension: Systolic \leq 160 mm Hg and Diastolic \leq 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 85 | Blood Pressure Category: Low Blood Pressure |
| 112 70 | Blood Pressure Category: Normal Blood Pressure |

Answer: (penalty regime: 0 %)

```

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

```

| | Input | Expected | Got | |
|---|--------------|--|--|---|
| ✓ | 50 85 | Blood Pressure Category: Low Blood Pressure | Blood Pressure Category: Low Blood Pressure | ✓ |
| ✓ | 112 70 | Blood Pressure Category: Normal Blood Pressure | Blood Pressure Category: Normal Blood Pressure | ✓ |
| ✓ | 135 86 | Blood Pressure Category: Prehypertension | Blood Pressure Category: Prehypertension | ✓ |
| ✓ | 145 98 | Blood Pressure Category: Stage 1 Hypertension | Blood Pressure Category: Stage 1 Hypertension | ✓ |
| ✓ | 170 110 | Blood Pressure Category: Stage 2 Hypertension | Blood Pressure Category: Stage 2 Hypertension | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 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 excellent | 500.0 |
| 31 good | 900.0 |
| 41 poor | 1100.0 |

Answer: (penalty regime: 0 %)

```

1 a=int(input())
2 b=input()
3 if(a>=18 and a<=30):
4     if(b=="excellent"):
5         print(500.0)
6     elif(b=="good"):
7         print(700.0)
8     else:
9         print(900.0)
10 elif(a>=31 and a<=50):
11     if(b=="excellent"):
12         print(700.0)
13     elif(b=="good"):
14         print(900.0)
15     else:
16         print(1100.0)
17
18
19

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 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: 70 |
| 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 b=0
2 for i in range(5):
3     a=int(input())
4     b=b+a
5 f=int(b/5)
6 print("Average score:",f)
7 if(f>=50):
8     print("The student has passed")
9 else:
10    print("The student has failed")
11

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 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 ld=n%10
3 rem=n//10
4 d=len(str(rem))
5 h=ld*(10**d)+rem
6 print(h)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Sunday, 24 August 2025, 10:22 AM

State Finished

Completed on Monday, 25 August 2025, 6:19 AM

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

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 a=int(input())
2 b=int(input())
3 c=0
4 for num in range(a,b+1):
5     if num>1:
6         ip=True
7         for i in range(2,int(num**0.5)+1):
8             if num%i==0:
9                 ip=False
10            break
11        if ip:
12            c+=1
13 print(c)
14
15

```

| | Input | Expected | Got | |
|---|------------|----------|-----|---|
| ✓ | 2 20 | 8 | 8 | ✓ |
| ✓ | 700 725 | 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 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     print(" ".join(str(num+j) for j in range(i))).center(2*n))
5     num+=i
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | 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 n=int(input())
3 s=math.sqrt(n)
4 k=int(s)+1
5 a=k*k
6 print(a)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | 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 a=input()
3 if(a=='W'):
4     for i in range(N):
5         row=""
6         for j in range(N):
7             if(i+j)%2==0:
8                 row+="W"
9             else:
10                row+="B"
11            print(row)
12 elif(a=='B'):
13     for i in range(N):
14         row=""
15         for j in range(N):
16             if(i+j)%2==0:
17                 row+="B"
18             else:
19                 row+="W"
20         print(row)
21
22
23
24
25

```

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

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

Assumption: The input number will be a positive integer number ≥ 1 and ≤ 25000 .

Some examples are as below.

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

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

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

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

For example:

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

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 freq=[0]*10
3 s=str(n)
4 for a in s:
5     b=int(a)
6     freq[b]+=1
7 d=0
8 for c in freq:
9     if c==1:
10        d+=1
11 print(d)

```

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

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 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 import math
2 n=int(input())
3 c=0
4 for i in range(1,n+1):
5     if math.gcd(i,n)==1:
6         c+=1
7 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 7 | Correct Mark 1.00 out of 1.00

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

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

Input Format:

x_1 -position of kangaroo1
 v_1 -Speed of kangaroo1
 x_2 -position of kangaroo2
 v_2 -Speed of kangaroo2
 k -jumps

Output Format:

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

For example:

| Input | Result |
|-------|--------|
| 0 | 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")

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | 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 s=0
3 ns=str(n)
4 for i in range(len(ns)):
5     d=int(ns[i])
6     s+=d**i
7 if(s==n):
8     print("Yes")
9 else:
10    print("No")

```

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

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 90 | No | No | ✓ |
| ✓ | 518 | Yes | Yes | ✓ |

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 a=int(input())
2 if a<10:
3     print(a)
4 else:
5     s=""
6     for d in range(9,1,-1):
7         while a%d==0:
8             s=str(d)+s
9             a/=d
10    if a>1:
11        print(0)
12    else:
13        ans=int(s)
14        print(ans if ans<= 2**31-1 else 0)
15

```

| | 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

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

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

If 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 a=len(str(n))
3 p=10**a
4 sn=n*n
5 if sn%p==n:
6     print("Automorphic")
7 else:
8     print("Not Automorphic")

```

| | 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.

Started on Monday, 25 August 2025, 7:57 PM

State Finished

Completed on Monday, 25 August 2025, 11:39 PM

Time taken 3 hours 42 mins

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | 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 if n<0:
3     print(False)
4 else:
5     while n%2==0:
6         n=n//2
7     while n%3==0:
8         n=n//3
9     while n%5==0:
10        n=n//5
11    if n==1:
12        print(True)
13    else:
14        print(False)

```

| | 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 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=int(input())
2 while n>=10:
3     temp=n
4     sumdi=0
5     while temp>0:
6         digit=temp%10
7         sumdi=sumdi+digit
8         temp=temp//10
9     n=sumdi
10 print(n)

```

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

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 <= n <= 45**

For example:

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

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 if n==1:
3     print(1)
4 else:
5     a=1
6     b=2
7     i=3
8 while i<=n:
9     c=a+b
10    a=b
11    b=c
12    i+=1
13 print(b)

```

| | 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 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^{th} element of the list, sorted ascending. If there is no p^{th} element, return 0.

Example

$n = 20$

$p = 3$

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

Constraints

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^9$

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

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

Sample Case 0**Sample Input 0**

10

3

Sample Output 0

5

Explanation 0

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

Sample Case 1**Sample Input 1**

10

5

Sample Output 1

0

Explanation 1

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

Sample Case 2**Sample Input 2**

1

1

Sample Output 2

1

Explanation 2

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

For example:

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

| Input | Result |
|-------|--------|
| 1 | 1 |
| 1 | |

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 p=int(input())
3 i=1
4 c=0
5 ans=0
6 while i<=n:
7     if n%i==0:
8         c+=1
9     if c==p:
10        ans=i
11        break
12    i+=1
13 if ans!=0:
14     print(ans)
15 else:
16     print(0)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | 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 num=n
3 c=0
4 while n>0:
5     digit=n%10
6     if digit!=0:
7         if num%digit==0:
8             c=c+1
9     n=n//10
10 print(c)

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 7 | 1 | 1 | ✓ |
| ✓ | 121 | 2 | 2 | ✓ |

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | 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 for i in bin(n)[2:]:
4     if i=='1':
5         c+=1
6 print(c)

```

| | Input | Expected | Got | |
|---|------------|----------|-----|---|
| ✓ | 11 | 3 | 3 | ✓ |
| ✓ | 128 | 1 | 1 | ✓ |
| ✓ | 32 | 1 | 1 | ✓ |
| ✓ | 2147483645 | 30 | 30 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | 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 <= num <= 108`

Answer: (penalty regime: 0 %)

```
1 num=int(input())
2 if num<=0:
3     print(False)
4 else:
5     tot=0
6     i=1
7 while i<num:
8     if num%i==0:
9         tot=tot+i
10    i+=1
11 if tot==num:
12     print(True)
13 else:
14     print(False)
```

| | Input | Expected | Got | |
|---|-------|----------|-------|---|
| ✓ | 28 | True | True | ✓ |
| ✓ | 7 | False | False | ✓ |
| ✓ | 8128 | True | True | ✓ |
| ✓ | 496 | True | True | ✓ |
| ✓ | 500 | False | False | ✓ |

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

For example:

| Input | Result |
|-------|----------------------------------|
| 4 | <pre> * *** **** ***** </pre> |

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 for i in range(1,n+1):
3     print(" "* (n-i) + "*"*(2*i-1))

```

| | Input | Expected | Got | |
|---|-------|--|--|---|
| ✓ | 4 | <pre> * *** **** ***** </pre> | <pre> * *** **** ***** </pre> | ✓ |
| ✓ | 2 | <pre> * *** </pre> | <pre> * *** </pre> | ✓ |
| ✓ | 5 | <pre> * *** **** ***** ****** </pre> | <pre> * *** **** ***** ****** </pre> | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | 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 c=0
3 while n>0:
4     n=n//5
5     c+=n
6 print(c)

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 3 | 0 | 0 | ✓ |

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | 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=int(input())
2 while n!=1 and n!=4:
3     temp=n
4     sum=0
5     while temp>0:
6         digit=temp%10
7         sum=sum+digit*digit
8         temp=temp//10
9     n=sum
10 if n==1:
11     print(True)
12 else:
13     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.

Started on Tuesday, 26 August 2025, 12:45 PM

State Finished

Completed on Sunday, 31 August 2025, 11:34 PM

Time taken 5 days 10 hours

Marks 4.00/5.00

Grade **80.00** out of 100.00

Question 1 | Incorrect Mark 0.00 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     s=str(n)
3     r=""
4     for c in s[::-1]:
5         if c=='0':
6             r+="0"
7         elif c=='1':
8             r+="1"
9         elif c=='6':
10            r+="9"
11        elif c=='8':
12            r+="8"
13        elif c=='9':
14            r+="6"
15        else:
16            r="x"
17            break
18

```

```
19 if(r==n):  
20     return False  
21 else:  
22     return True  
23  
24  
25  
26  
27  
28  
29
```

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

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

Show differences 

Incorrect

Marks for this submission: 0.00/1.00.

Question 2 | 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     while(n%2==0):
3         n=n//2
4     while(n%3==0):
5         n=n//3
6     while(n%5==0):
7         n=n//5
8     if(n==1):
9         return("ugly")
10    else:
11        return("not ugly")
12
13

```

| | 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 3 | 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 |
|----------------------------------|--------|
| <code>print(Summation(8))</code> | 9872 |

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 import math
2 def Summation(n):
3     s=str(n)
4     total=int(s)+int(s*2)+int(s*3)+int(s*4)
5     return total
6
7
8
9
10
11
12
13

```

| | Test | Expected | Got | |
|---|-----------------------------------|----------|----------|---|
| ✓ | <code>print(Summation(8))</code> | 9872 | 9872 | ✓ |
| ✓ | <code>print(Summation(10))</code> | 10203040 | 10203040 | ✓ |

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     c=0
3     while(n!=0):
4         b=n%10
5         if(b==2 or b==3 or b==5 or b==7):
6             c+=b
7         n=n//10
8     return(c)
9

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | 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     coins=[4,3,2,1]
3     count=0
4     for i in coins:
5         count+=n//i
6         n%=i
7     return count
8 
```

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

Passed all tests! ✓

[Correct](#)

Marks for this submission: 1.00/1.00.

Started on Tuesday, 9 September 2025, 1:28 AM

State Finished

Completed on Tuesday, 9 September 2025, 11:32 PM

Time taken 22 hours 4 mins

Marks 9.00/10.00

Grade **90.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

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

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

Example 1:

Input:

thequickbrownfoxjumpsoverthelazydog

Output:

true

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

Example 2:

Input:

arvijayakumar

Output: false

Constraints:

1 <= sentence.length <= 1000

sentence consists of lowercase English letters.

For example:

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

Answer: (penalty regime: 0 %)

Reset answer

```
1 def checkPangram(s):
2     for i in "abcdefghijklmnopqrstuvwxyz":
3         if i not in s:
4             return "false"
5     return "true"
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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 rajalakshmi arvijayakumar |

Answer: (penalty regime: 0 %)

```
1 s=input()
2 username, remaining = s.split("@",1)
3 domain, extention = remaining.split(".",1)
4 print(extention)
5 print(domain)
6 print(username)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | 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 r=s[::-1]  
3 print(r)
```

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

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 kw=["break", "case", "continue", "default", "defer", "else", "for", "func", "goto", "if", "map", "range", "return"]
2 s=input()
3 if s in kw:
4     print(s,"is a keyword")
5 else:
6     print(s,"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

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 c=""
3 for i in s:
4     if i.isalnum():
5         c+=i.lower()
6 if c==c[::-1]:
7     print(1)
8 else:
9     print(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 6 | 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

aabbbbccccc

Answer: (penalty regime: 0 %)

```

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

```

| | Input | Expected | Got | |
|---|---------|-----------------|-----------------|---|
| ✓ | a2b4c6 | aabbbbccccc | aabbbbccccc | ✓ |
| ✓ | a12b3d4 | aaaaaaaaaaabbdd | aaaaaaaaaaabbdd | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00

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

Boundary Condition(s):

$1 \leq N \leq 100$

$1 \leq \text{Length of each string} \leq 100$

Input Format:

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

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

Output Format:

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

Example Input/Output 1:

Input:

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

Output:

```
Hey google
luckycharms
-1
```

Answer: (penalty regime: 0 %)

```
1 a=int(input())
2 for i in range(a):
3     s=input()
4     v=""
5     for j in s:
6         if j == "^":
7             v=v[:-1]
8         else:
9             v=v+j
10    if(v==""):
11        print(-1)
12    else:
13        print(v)
14
15
16
17
18
19
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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     st=''
3     for i in s:
4         if i in "([{":
5             st+=i
6         else:
7             if len(st)==0:
8                 return "false"
9             j=st[-1]
10            st=st[:-1]
11            if j+i not in "()-{}-[]":
12                return "false"
13        if(len(st)==0):
14            return "true"
15        else:
16            return "false"
17
18
19

```

| | 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 | Incorrect Mark 0.00 out of 1.00

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

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

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

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

Note:

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

Example 1:**Input**

internationalization

i12iz4n

Output

true

Explanation

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

Return true.

Example 2:**Input**

apple

a2e

Output

false

Explanation

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

Return false.

Answer: (penalty regime: 0 %)

```

1 def valab(s,ab):
2     i=j=0
3     while i<len(s) and j<len(ab):
4         if ab[i].isdigit():
5             if ab[j]=='0':
6                 return False
7             num=0
8             while j<len(ab) and ab[j].isdigit():
9                 num= num*10 + int(ab[j])
10                j+=1
11            i+=num
12        else:
13            if i>=len(s) or s[i]!=ab[j]:
14                return False
15            i+=1
16            j+=1
17    return i==len(s) and j==len(ab)
18
19

```

| | Input | Expected | |
|---|---------------------------------|----------|---|
| ✗ | internationalization i12iz4n | true | ✗ |
| ✗ | apple a2e | false | ✗ |

Some hidden test cases failed, too.

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

Incorrect

Marks for this submission: 0.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

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

Sample Input 1

thistest123string

123

Sample Output 1

8

Answer: (penalty regime: 0 %)

```
1 a=input()
2 b=input()
3 if b in a:
4     print(a.index(b))
5 else:
6     print(-1)
7
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Tuesday, 23 September 2025, 12:25 PM

State Finished

Completed on Thursday, 25 September 2025, 11:57 PM

Time taken 2 days 11 hours

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

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

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

Example 1:

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

Output:

1

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

For example:

| Input | Result |
|--|--------|
| hello world ad | 1 |
| Faculty Upskilling in Python Programming ak | 2 |

Answer: (penalty regime: 0 %)

```

1 t=input()
2 broken=input()
3 broken_set=set(broken.lower())
4 words=t.split()
5 typec=0
6 for word in words:
7     can_type=True
8     for char in word.lower():
9         if char in broken_set:
10             can_type=False
11             break
12     if can_type:
13         typec+=1
14 print(typec)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

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

Examples:

Input: t = (5, 6, 5, 7, 7, 8), K = 13

Output: 2

Explanation:

Pairs with sum K(= 13) are {(5, 8), (6, 7), (6, 7)}.

Therefore, distinct pairs with sum K(= 13) are { (5, 8), (6, 7) }.

Therefore, the required output is 2.

For example:

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

Answer: (penalty regime: 0 %)

```

1 s=input()
2 t=tuple(int(x) for x in s.split(','))
3 k=int(input())
4 dist=set()
5 seen=set()
6 for num in t:
7     comp=k-num
8     if comp in seen:
9         pair=tuple(sorted((num,comp)))
10        dist.add(pair)
11    seen.add(num)
12 print(len(dist))

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

Check if a set is a subset of another set.

Example:

Sample Input1:

mango apple

mango orange

mango

output1:

yes

set3 is subset of set1 and set2

input2:

mango orange

banana orange

grapes

output2:

no

For example:

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

Answer: (penalty regime: 0 %)

```

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

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | 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     k=int(input())
5     s.add(k)
6 print(*s)

```

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

Passed all tests! ✓

Correct

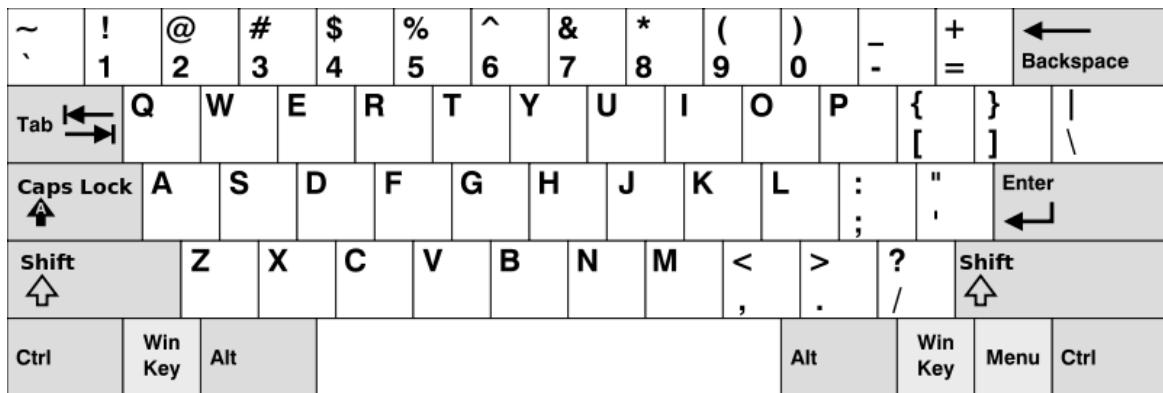
Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Given an array 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 words=[input() for _ in range(n)]
3 row1=set("qwertyuiop")
4 row2=set("asdfghjkl")
5 row3=set("zxcvbnm")
6 result=[]
7 for i in words:
8     lower_word=set(i.lower())
9     if lower_word<=row1 or lower_word<=row2 or lower_word<=row3:
10         result.append(i)

```

```
11 if not result:  
12     print("No words")  
13 else:  
14     print(*result,sep='\n')
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | 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 s=input()
2 chars=set(s)
3 if chars.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 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=input()
2 n1,n2=map(int,n.split())
3 arr1n=input()
4 arr1=set(map(int,arr1n.split()))
5 arr2n=input()
6 arr2=set(map(int,arr2n.split()))
7 ce=arr1.intersection(arr2)
8 nons=sorted(list(arr1.symmetric_difference(arr2)))
9 if nons:
10     print(*nons)
11     print(len(nons))
12 else:
13     print("NO SUCH ELEMENTS")
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00

Given an array of integers `nums` containing $n + 1$ integers where each integer is in the range $[1, n]$ inclusive. There is only **one repeated number** in `nums`, return *this repeated number*. Solve the problem using set.

Example 1:

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

Output: 2

Example 2:

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

Output: 3

For example:

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

Answer: (penalty regime: 0 %)

```

1 num=input()
2 nums=[int(x) for x in num.split()]
3 seen=set()
4 for n in nums:
5     if n in seen:
6         print(n)
7         break
8     seen.add(n)

```

| | 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 9 | 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     nums=list(nums)
3     count=0
4     while nums:
5         if nums[0]==min(nums):
6             nums.pop(0)
7         else:
8             nums.append(nums.pop(0))
9         count+=1
10    return count

```

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

| | Test | Expected | Got | |
|---|--|----------|-----|---|
| ✓ | 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 10 | 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 = "AAAAACCCCCAAAAACCCCCAAAAAGGGTTT"

Output: ["AAAAACCCC", "CCCCAAAAA"]

Example 2:

Input: s = "AAAAAAAAAAAAAA"

Output: ["AAAAAAAAAA"]

For example:

| Input | Result |
|---------------------------------|------------------------|
| AAAAACCCCCAAAAACCCCCAAAAAGGGTTT | AAAAACCCC CCCCAAAAA |

Answer: (penalty regime: 0 %)

```

1 s=input()
2 n=10
3 result=[]
4 seen=set()
5 for i in range(len(s)-n+1):
6     substring=s[i:i+n]
7     if substring in seen:
8         if substring not in result:
9             result.append(substring)
10    else:
11        seen.add(substring)
12 print(*result, sep='\n')

```

| | Input | Expected | Got | |
|---|---------------------------------|------------------------|------------------------|---|
| ✓ | AAAAACCCCCAAAAACCCCCAAAAAGGGTTT | AAAAACCCC CCCCAAAAA | AAAAACCCC CCCCAAAAA | ✓ |
| ✓ | AAAAAAAAAAAAAA | AAAAAAAAAA | AAAAAAAAAA | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Tuesday, 7 October 2025, 10:37 PM

State Finished

Completed on Tuesday, 7 October 2025, 11:01 PM

Time taken 24 mins 9 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 |
|------------------------|---------------|
| print(excelNumber(26)) | Z |

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 def excelNumber(n):
2     l=tuple("ABCDEFGHIJKLMNOPQRSTUVWXYZ")
3     r=""
4     while n>0:
5         n-=1
6         r=l[n%26]+r
7         n//=26
8     return r

```

| | 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

The program must accept a string S which represents a series of keystrokes as the input. There are two special keystrokes denoted by "^z" for undo operation and "y" 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):

$1 \leq \text{Length of string} \leq 1000$

Input Format:

The first line contains the string S.

Output Format:

The first line contains the modified string.

Example Input/Output 1:

Input:

missel^z^z^z^ypell^z^z^yt

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().strip()
2 t=[];u=[]
3 i=0
4 while i<len(s):
5     if s[i]=='^' and i+1<len(s):
6         if s[i+1]=='z':
7             if t:
8                 u.append(t.pop())
9             elif s[i+1]=='y':
10                if u:

```

```
11             t.append(u.pop())
12             i+=2
13 v     else:
14         t.append(s[i])
15         u.clear()
16         i+=1
17 print("".join(t))
```

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

Passed all tests! ✓

Correct

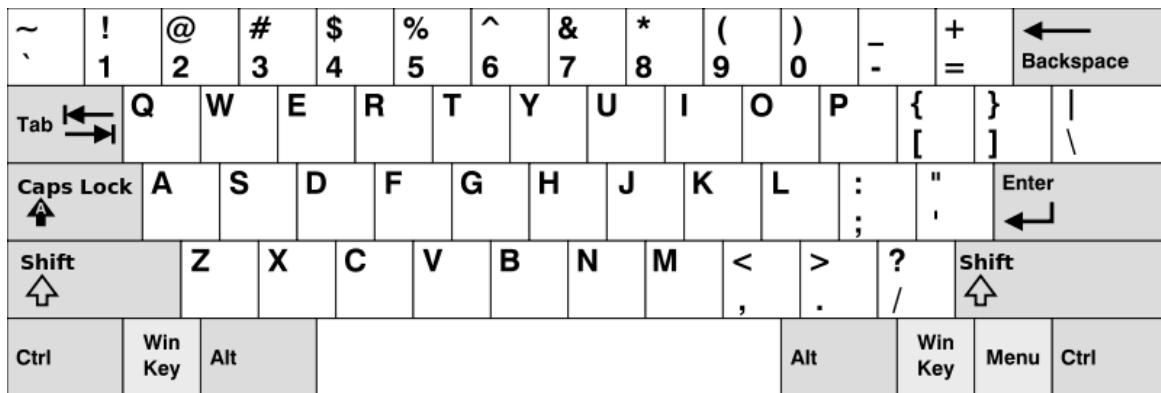
Marks for this submission: 1.00/1.00.

Question 3 | 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 w=[input().strip() for _ in range(n)]
3 s1=set("qwertyuiop")
4 s2=set("asdfghjkl")
5 s3=set("zxcvbnm")
6 l=[]
7 for i in w:
8     s=set(i.lower())
9     if s.issubset(s1) or s.issubset(s2) or s.issubset(s3):
10         l.append(i)

```

```
11 v if l:  
12 v     for _ in l:  
13         print(_)  
14 v else:  
15     print("No words")
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | 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 t=tuple((w[:-1],int(w[-1]))for w in s)
3 t=sorted(t,key=lambda x: x[1])
4 print(' '.join(w for w,_ in t))
```

| | 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 5 | Correct Mark 1.00 out of 1.00

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

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

Example 1:

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

Output:

1

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

For example:

| Input | Result |
|--|--------|
| hello world ad | 1 |
| Faculty Upskilling in Python Programming ak | 2 |

Answer: (penalty regime: 0 %)

```

1 t=input().strip()
2 b=set(input().strip().lower())
3 c=0
4 for i in t.split():
5     if b.isdisjoint(i.lower()):
6         c+=1
7 print(c)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Tuesday, 7 October 2025, 12:39 PM

State Finished

Completed on Friday, 10 October 2025, 1:30 AM

Time taken 2 days 12 hours

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | 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 | |
|---|---|----------|-----|---|
| ✓ | <code>print(count_special_letters("AaBbCcDdEe"))</code> | 5 | 5 | ✓ |

| | Test | Expected | Got | |
|---|---------------------------------------|----------|-----|---|
| ✓ | 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 2 | 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 Johny get maximum votes. Since John is alphabetically smaller, we print it. Use dictionary to solve the above problem

Sample Input:

10
 John
 John
 Johny
 Jamie
 Jamie
 Johny
 Jack
 Johny
 Johny
 Jackie

Sample Output:

Johny

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

Answer: (penalty regime: 0 %)

```

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

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

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

Description:

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

Program Requirements:**Input:**

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

Output:

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

The output should list each vowel followed by its count.

Example:

Consider the following example for better understanding:

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

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

For example:

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

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

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

Input

n: Float data type which reads amount to send

Output

s: : String data type which displays symbols

Sample Testcase 1

Input

10000

Output

@!!!!.!!

Sample Testcase2

1234.56

Output

@#\$%.^&

For example:

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

Answer: (penalty regime: 0 %)

```

1 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('!.!!')
10

```

| | Input | Expected | Got | |
|---|---------|-----------|-----------|---|
| ✓ | 1345.23 | @\$%^.#\$ | @\$%^.#\$ | ✓ |

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

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()
2 d={}
3 for i in s.split():
4     d[i[-1]]=i[:-1]
5 for i in sorted(d):
6     print(d[i],end=' ')
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | 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]))

```

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

| | Input | Expected | Got | |
|---|-----------------------|----------|--------|---|
| ✓ | apple apple banana | banana | banana | ✓ |

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 Gfg 6 7 4 Best 7 6 5 | Gfg 17 Best 18 | Gfg 17 Best 18 | ✓ |
| ✓ | 2 Gfg 6 6 Best 5 5 | Best 10 Gfg 12 | Best 10 Gfg 12 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00

Given a number, convert it into corresponding alphabet.

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

Input Format

Input is an integer

Output Format

Print the alphabets

Constraints

$1 \leq num \leq 4294967295$

Sample Input 1

26

Sample Output 1

Z

For example:

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

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 def excelNumber(n):
2     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 9 | Correct Mark 1.00 out of 1.00

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

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

Note:

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

Sample input:

```
4
James 67 89 56
Lalith 89 45 45
Ram 89 89 89
Sita 70 70 70
```

Sample Output:

```
Ram
James Ram
Lalith
Lalith
```

For example:

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

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 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         lname.append(t)
20         hname.append(a)
21         maxass.append(l)
22         minlab.append(j)
23 print(*lname)
24 print(*hname)
25 print(*maxass)
26 print(*minlab)
```

```

10     ha=avg
11     hname=[i]
12 v   elif avg==ha:
13     hname.append(i)
14 v   if(avg<la):
15     la=avg
16     lname=[i]
17 v   elif avg==la:
18     lname.append(i)
19 v   if a>ma:
20     ma=a
21     maxass=[i]
22 v   elif(a==ma):
23     maxass.append(i)
24 v   if(l<ml):
25     ml=l
26     minlab=[i]
27 v   elif l==ml:
28     minlab.append(i)
29
30 print(' '.join(sorted(hname)))
31 print(' '.join(sorted(maxass)))
32 print(' '.join(sorted(minlab)))
33 print(' '.join(sorted(lname)))
34
35
36
37
38
39
40
41
42
43
44
45

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

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

Points Letters

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

2 D and G

3 B, C, M and P

4 F, H, V, W and Y

5 K

8 J and X

10 Q and Z

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

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

[Sample Input](#)

REC

[Sample Output](#)

REC is worth 5 points.

For example:

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

Answer: (penalty regime: 0 %)

```

1 a={"A":1,"E":1,"I":1,"L":1,"N":1,"O":1,"R":1,"S":1,"T":1,"U":1,"D":2,"G":2,"B":3,"C":3,"M":3,"P":3,"F":4,"H":4,"V":4}
2 word=input().upper()
3 score=0
4 for ch in word:
5     score+=a.get(ch,0)
6 print(word,"is worth",score,"points.")
7

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Started on Tuesday, 14 October 2025, 11:02 PM

State Finished

Completed on Tuesday, 14 October 2025, 11:27 PM

Time taken 24 mins 40 secs

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

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

Description:**1. Input:**

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

2. Output:

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

Example:**• Input File Content:**

"Line one.

Line two.

Line three.

Line four."

3

Inserted line..

Output:

Line one.

Line two.

Inserted line.

Line three.

Line four.

For example:

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

Answer: (penalty regime: 0 %)

```

1 s=input()
2 n=int(input())
3 ins=input()
4 s=open(s,'r')
5 d=s.readlines()
6 c=0
7 with open('output.txt','w') as file:
8     for i in d:
9         c+=1
10    if c==n:
11        file.write(ins+"\n")
12        file.write(i)
13    else:
14        file.write(i)
15    if n>len(d):
16        file.write("\n"+ins)
17

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

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

Description:**1. Input:**

- Source file and destination file names.

2. Output:

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

For example:

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

Answer: (penalty regime: 0 %)

```

1 s=input()
2 d=input()
3 try:
4     with open(s,'r') as src:
5         data = src.read()
6     with open(d,'w') as des:
7         des.write(data)
8 except Exception as e:
9     print("An error occurred:",e)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

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

Description:**1. Input:**

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

2. Output:

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

Example:**• Input File Content:**

"Line one.

Line two.

Line three.

Line four."

3

Output:

Line one.

Line two.

eerht eniL.

Line four.

For example:

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

Answer: (penalty regime: 0 %)

```

1 s=input()
2 n=int(input())
3 s1=open(s,'r')
4 data=s1.readlines()
5 s1.close()
6 c=0
7 with open('output.txt','w') as file:
8     for i in data:
9         c+=1
10    if c==n:
11        i=i.strip()
12        file.write(i[-2:0:-1]+i[0]+i[-1]+"\n")
13    else:
14        file.write(i)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

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

Description:**1. Input:**

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

2. Output:

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

Example:**• Input File Content:**

"Line one.

Line two.

Line three.

Line four."

2

Updated line two.

Output:

Line one.

Line three.

Line four.

For example:

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

Answer: (penalty regime: 0 %)

```

1 a=input()
2 n=int(input())
3 a=open(a, 'r')
4 data=a.readlines()
5 c=0
6 with open('output.txt', 'w') as file:
7     for i in data:
8         c+=1
9     if c==n:
10         pass
11     else:
12         file.write(i)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

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

Description:**1. Input:**

- String as input.

2. Output:

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

Example:**• Input File Content:**

```
apple orange apple banana apple orange
```

Output:

```
apple: 3
orange: 2
banana: 1
```

For example:

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

Answer: (penalty regime: 0 %)

```
1 import string
2 a=input().lower()
3 dic={}
4 trans=str.maketrans(' ',' ',string.punctuation)
5 a=a.translate(trans)
6 with open('output.txt','w') as file:
7     a=list(map(str,a.split()))
8     for i in a:
9         if i in dic:
10             dic[i]+=1
11         elif i not in dic:
12             dic[i]=1
13     di=list(sorted(dic.items()))
14     for j in di:
15         file.write(j[0].lower()+":"+str(j[1])+"\n")
```

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

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00

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

Description:**1. Input:**

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

2. Output:

- The total number of words in the file.

For example:

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

Answer: (penalty regime: 0 %)

```

1 s=input()
2 s1=open(s, 'r')
3 file=s1.read()
4 c=0
5 s2=file.split()
6 for i in s2:
7     c+=1
8 print("Total words:",len(s2))

```

| | 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 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 s=input()
2 s1=open(s,'r')
3 s2=s1.read().split()
4 maxi=0
5 lw=''
6 for i in s2:
7     if maxi<len(i):
8         maxi=len(i)
9         lw=i
10 print("Longest word:",lw)

```

| | 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

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

Description:**1. Input:**

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

2. Output:

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

Example:**• Input File Content:**

"Line one.

Line two.

Line three.

Line four."

2

Updated line two.

Output:

Line one.

Updated line two.

Line three.

Line four.

For example:

| Test | Input | Result |
|---|--|---|
| <pre>with open('output.txt', 'r') as file: text = file.read() print(text)</pre> | <input type="text" value="input1.txt"/> 2 <input type="text" value="Updated line two."/> | Line one. Updated line two. Line three. Line four. |

Answer: (penalty regime: 0 %)

```

1 s=input()
2 n=int(input())
3 us=input()
4 data=open(s,'r')
5 data=data.readlines()
6 c=0
7 with open('output.txt','w') as file:
8   for i in data:
9     c+=1
10  if c==n:
11    file.write(us+"\n")
12  else:
13    file.write(i)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 out of 1.00

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

Description:**1. Input:**

- A text file containing multiple words.

2. Output:

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

For example:

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

Answer: (penalty regime: 0 %)

```

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

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

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

Description:**1. Input:**

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

2. Output:

- The content of the specified line.

input1.txt:

Line one.

Line two.

Line three.

Line four.

For example:

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

Answer: (penalty regime: 0 %)

```

1 s=input()
2 n=int(input())
3 s1=open(s,'r')
4 file=s1.readlines()
5 c=0
6 for i in file:
7     c+=1
8 if c==n:
9     print(i)
10    break

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

