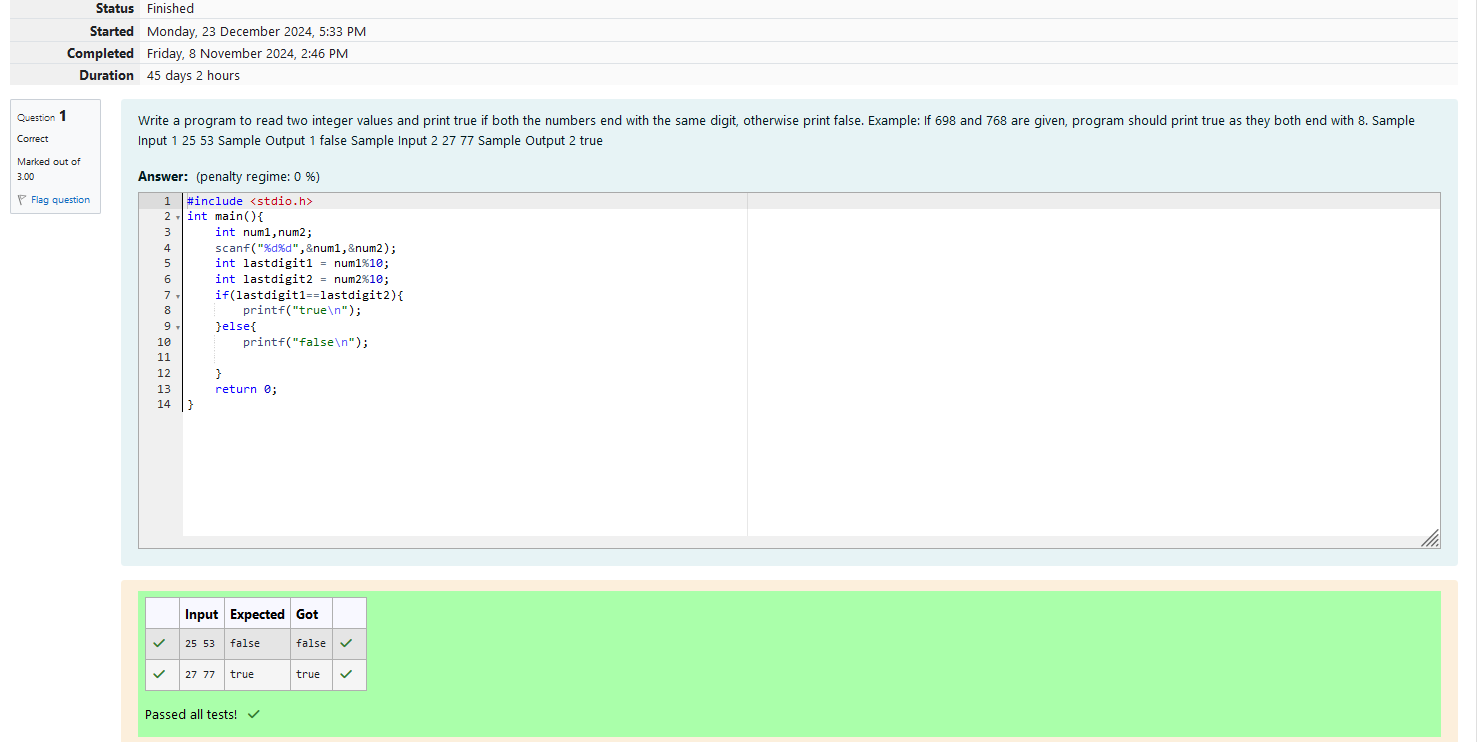
**Week 3**

**Roll No: 241401044**

**Name: Lakshanya.R.D**

**Week-03-01-Practice Session-Coding**

Q1) Write a program to read two integer values and print true if both the numbers end with the same digit, otherwise print false. Example: If 698 and 768 are given, program should print true as they both end with 8. Sample Input 1 25 53 Sample Output 1 false Sample Input 2 27 77 Sample Output 2 true



Q2) **Objective**

In this challenge, we're getting started with conditional statements.

**Task**

Given an integer, ***n***, perform the following conditional actions:

·         If ***n*** is odd, print Weird

·         If ***n*** is even and in the inclusive range of ***2*** to ***5***, print ***Not Weird***

·         If ***n*** is even and in the inclusive range of ***6*** to ***20***, print ***Weird***

·         If ***n*** is even and greater than ***20***, print ***Not Weird***

Complete the stub code provided in your editor to print whether or not ***n*** is weird.

**Input Format**

A single line containing a positive integer, ***n***.

**Constraints**

·         1 < n < 100

**Output Format**

Print Weird if the number is weird; otherwise, print Not Weird.

**Sample Input 0**

3

**Sample Output 0**

Weird

**Sample Input 1**

24

**Sample Output 1**

Not Weird

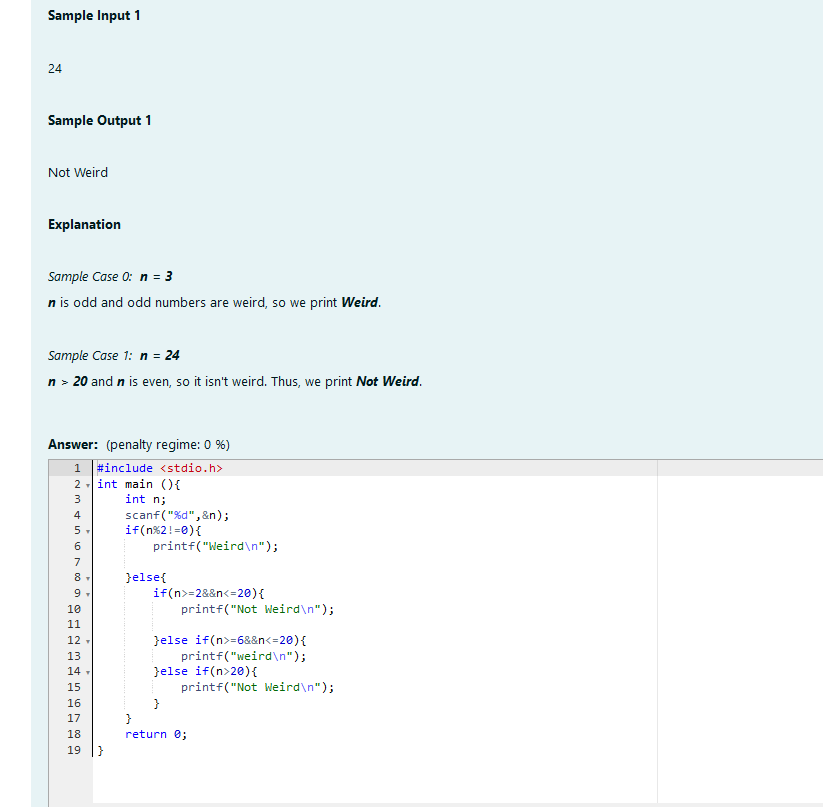
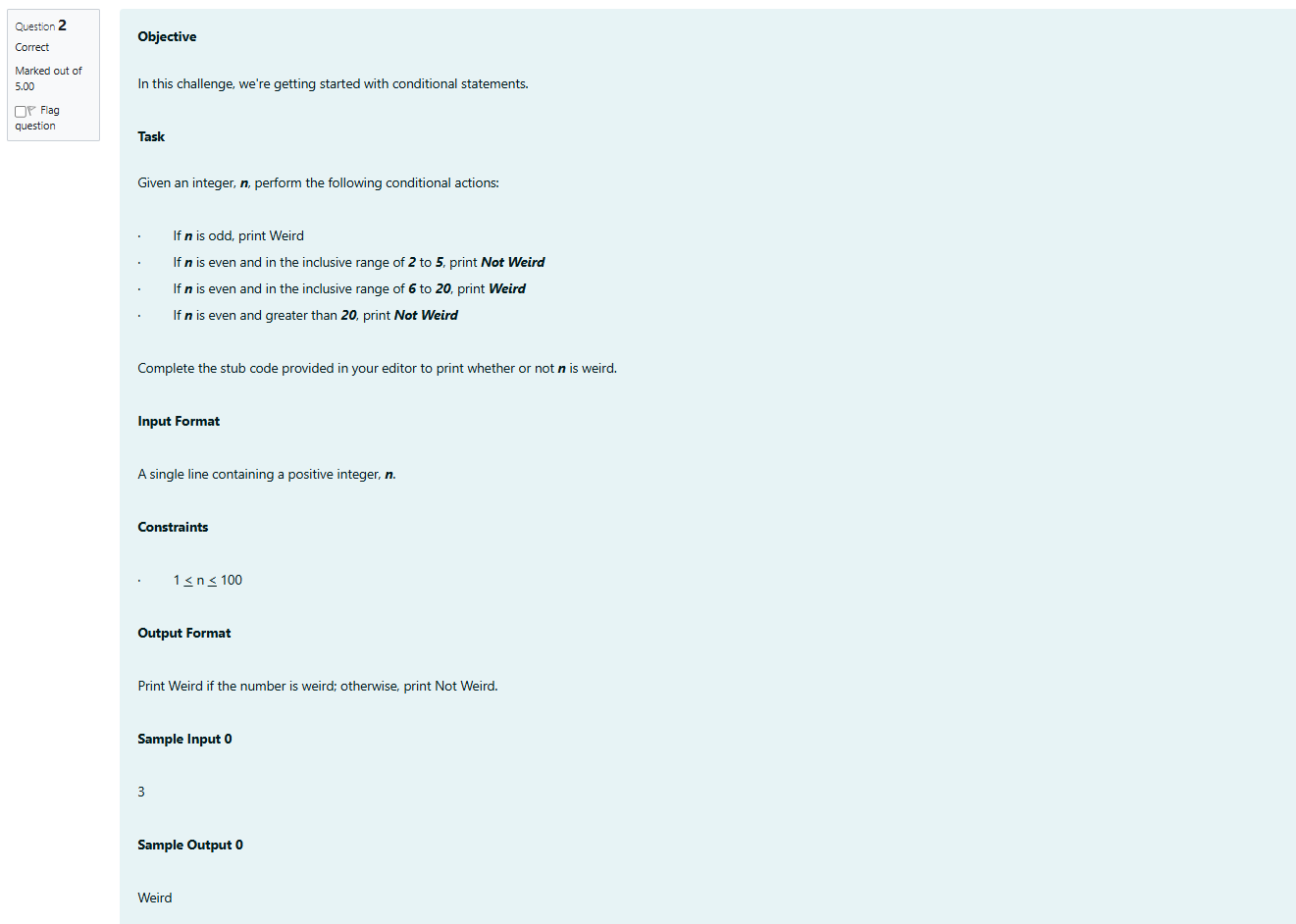
**Explanation**

*Sample Case 0:*  ***n = 3***

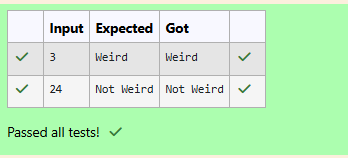
***n*** is odd and odd numbers are weird, so we print ***Weird***.

*Sample Case 1:*  ***n = 24***

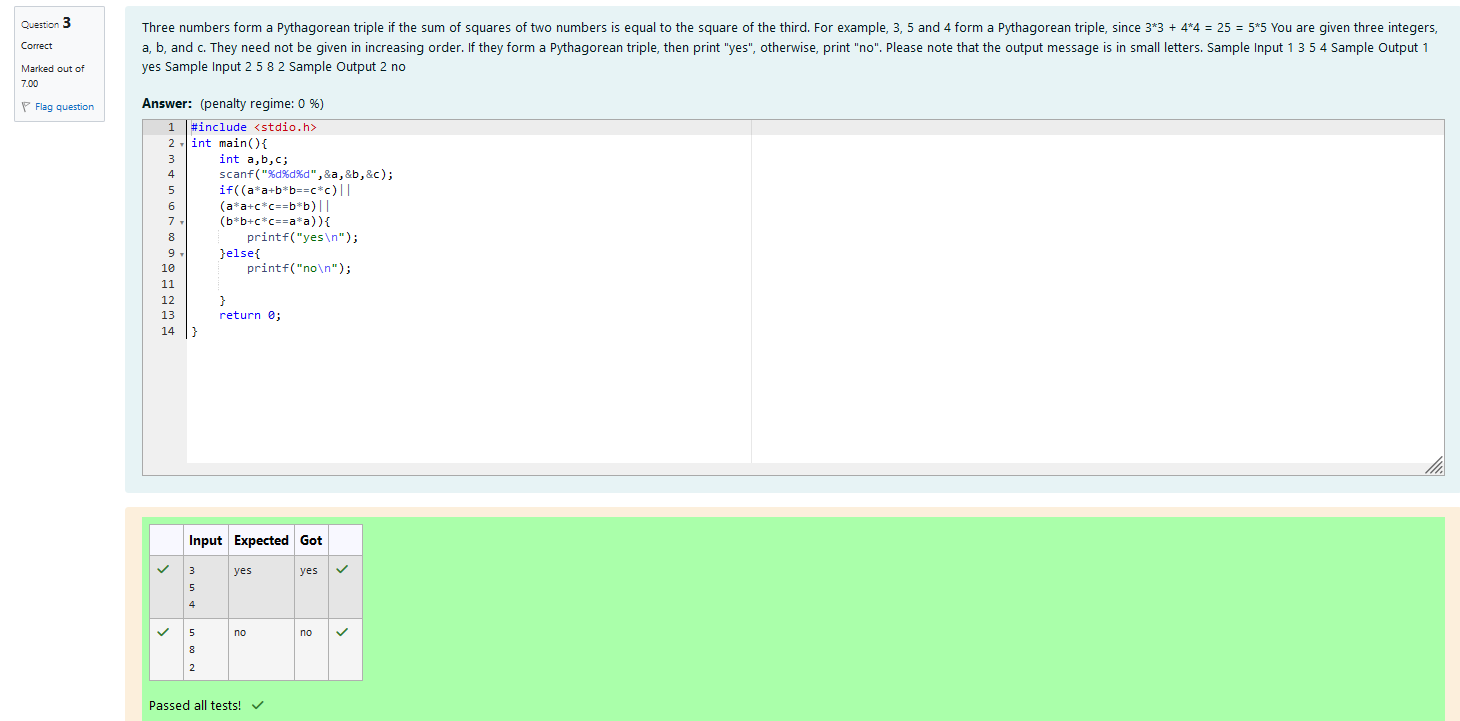
***n > 20*** and ***n*** is even, so it isn't weird. Thus, we print ***Not Weird***.



Output:



Q3)Three numbers form a Pythagorean triple if the sum of squares of two numbers is equal to the square of the third. For example, 3, 5 and 4 form a Pythagorean triple, since 3\*3 + 4\*4 = 25 = 5\*5 You are given three integers, a, b, and c. They need not be given in increasing order. If they form a Pythagorean triple, then print "yes", otherwise, print "no". Please note that the output message is in small letters. Sample Input 1 3 5 4 Sample Output 1 yes Sample Input 2 5 8 2 Sample Output 2 no



**Week-03-02-Practice Session-Coding**

Q1)Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to (and including) 10 sides. If a number of sides outside of this range is entered then your program should display an appropriate error message.

Sample Input 1

3

Sample Output 1

Triangle

Sample Input 2

7

Sample Output 2

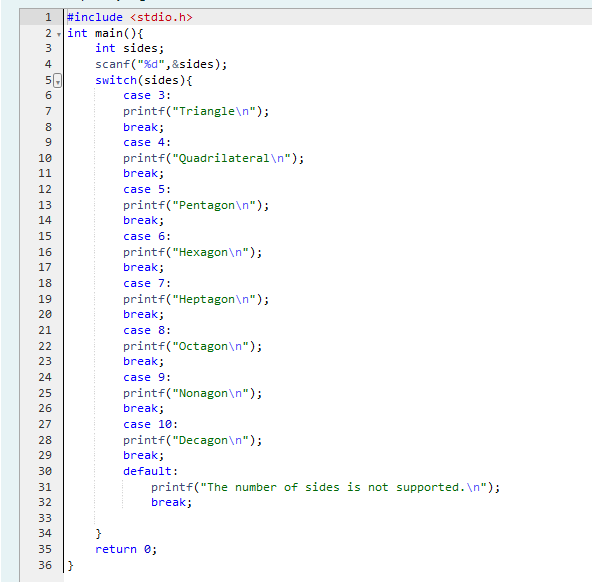
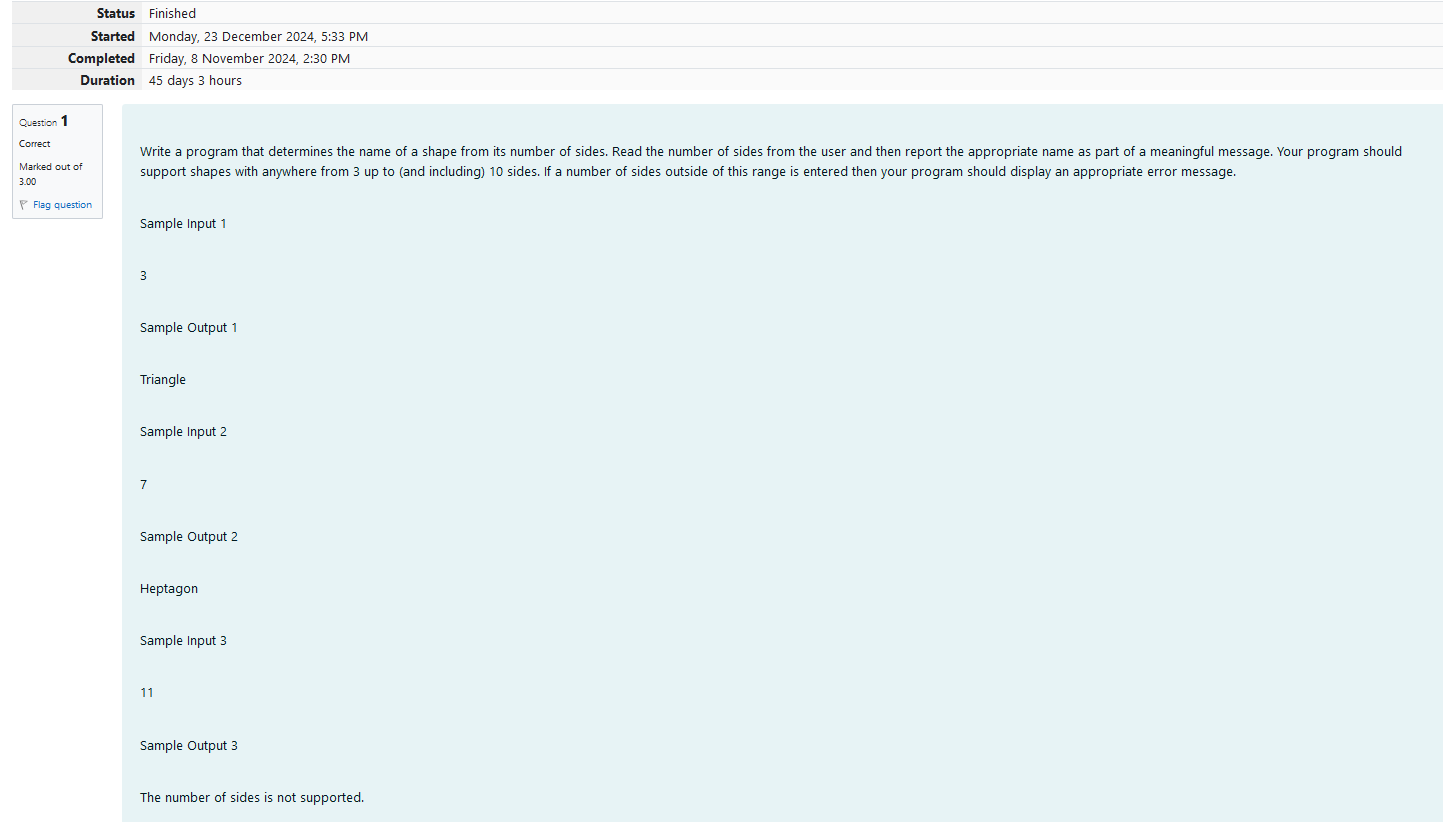
Heptagon

Sample Input 3

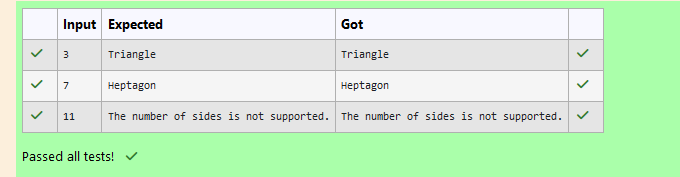
11

Sample Output 3

The number of sides is not supported.



Output:



Q2) 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

2004

Sample Output 1

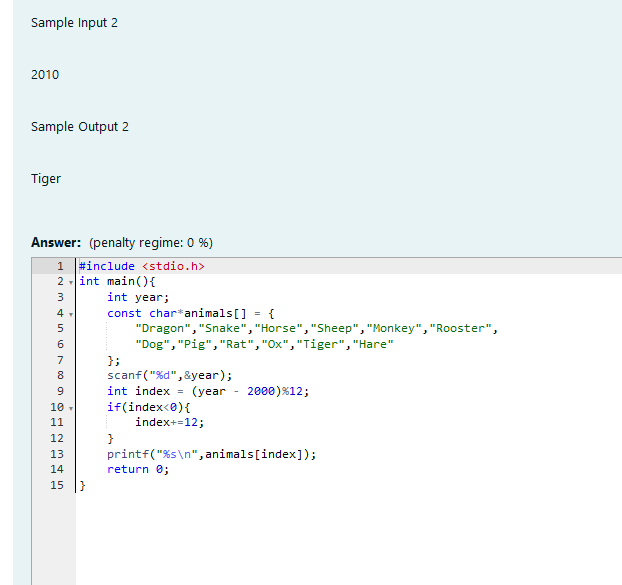
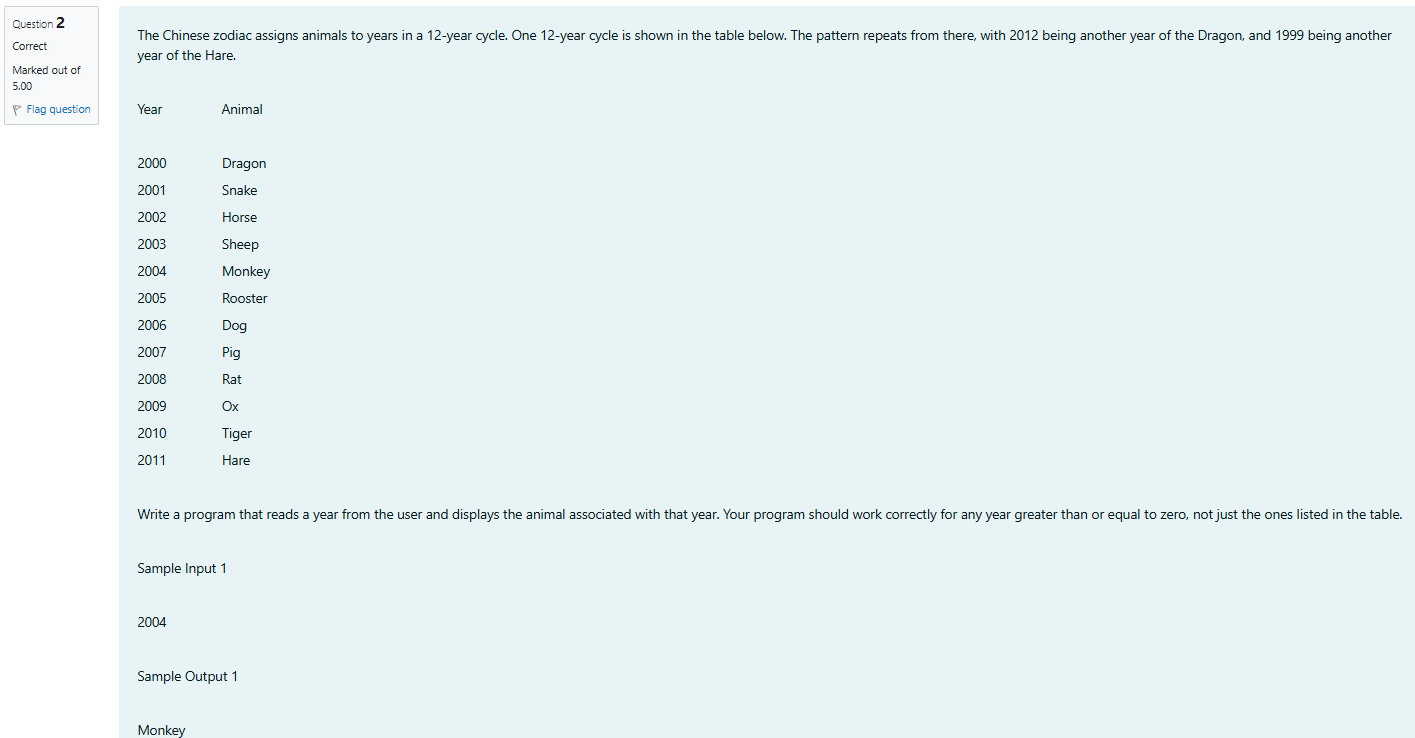
Monkey

Sample Input 2

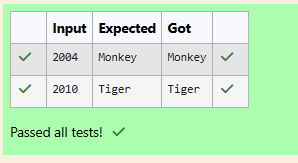
2010

Sample Output 2

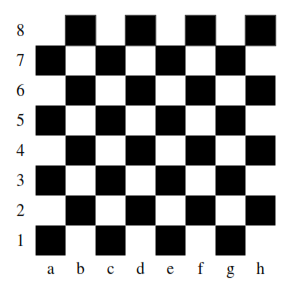
Tiger



Output:



Q3) Positions on a chess board are identiﬁed by a letter and a number. The letter identiﬁes the column, while the number identiﬁes the row:



Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square. Then use modular arithmetic to report the color of the square in that row. For example, if the user enters a1 then your program should report that the square is black. If the user enters d5 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

Sample Input 1

a 1

Sample Output 1

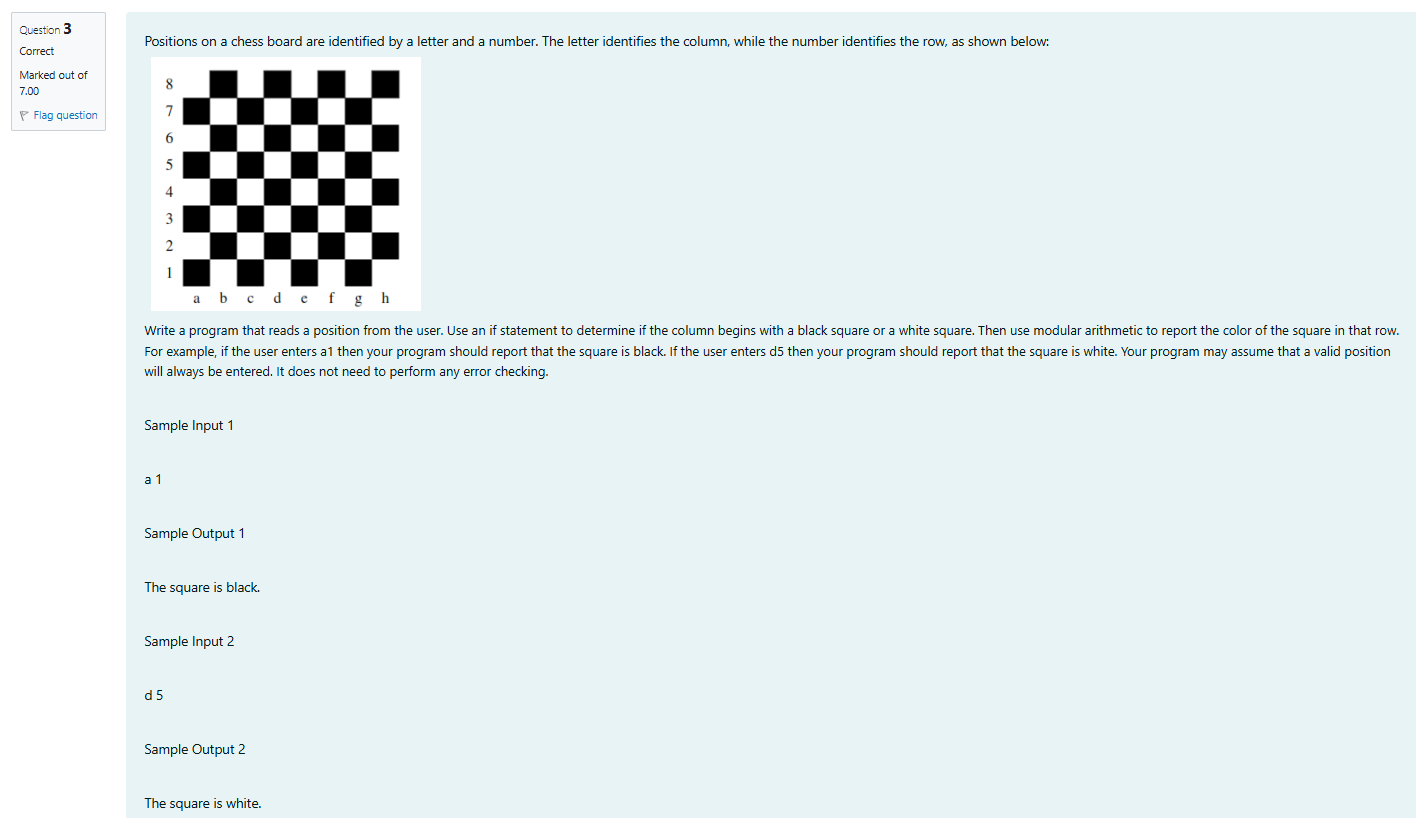
The square is black.

Sample Input 2

d 5

Sample Output 2

The square is white.





**Week-03-03-Practice Session-Coding**

Q1) Some data sets specify dates using the year and day of year rather than the year, month, and day of month. The day of year (DOY) is the sequential day number starting with day 1 on January 1st.

There are two calendars - one for normal years with 365 days, and one for leap years with 366 days. Leap years are divisible by 4. Centuries, like 1900, are not leap years unless they are divisible by 400. So, 2000 was a leap year.

To find the day of year number for a standard date, scan down the Jan column to find the day of month, then scan across to the appropriate month column and read the day of year number. Reverse the process to find the standard date for a given day of year.

Write a program to print the Day of Year of a given date, month and year.

Sample Input 1

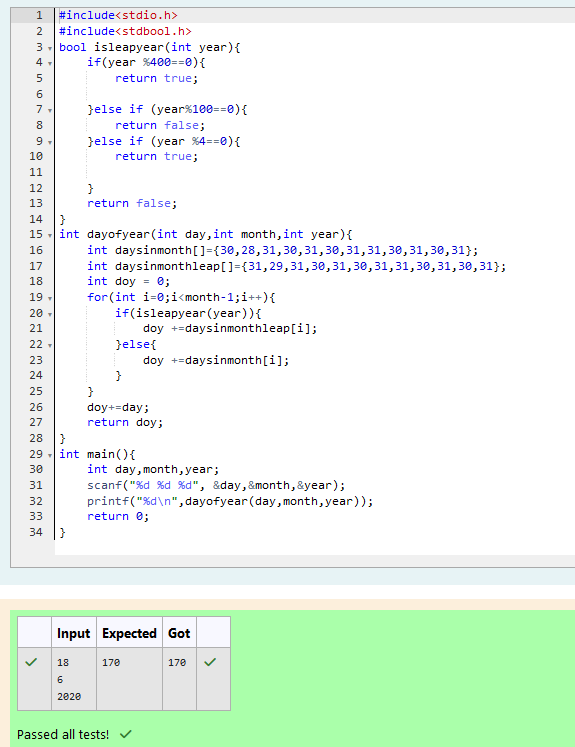
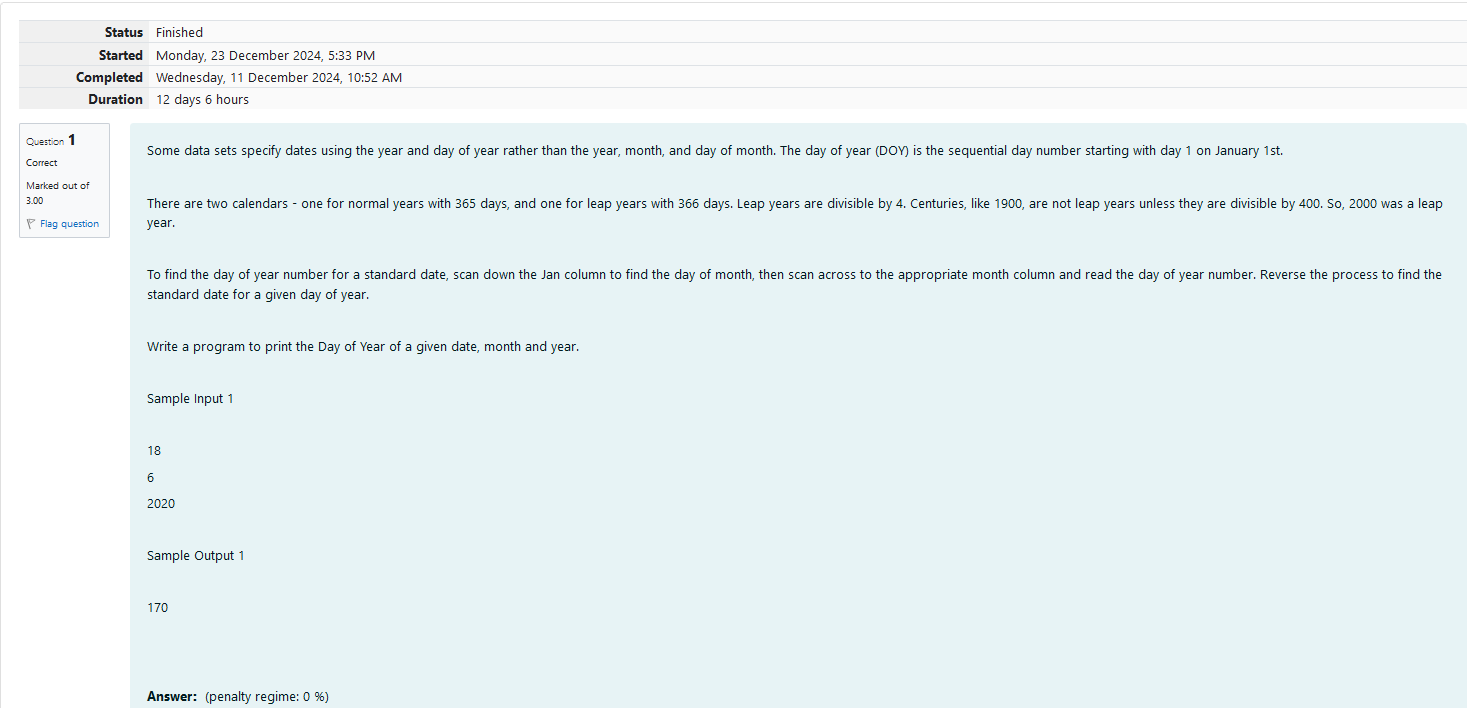
18

6

2020

Sample Output 1

170



Q2) Suppandi is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suppandi, is confused, he was never any good at math. And also, he is bad at remembering the names of shapes. Instead, you will be helping him [calculate the area](http://www.rajalakshmicolleges.org/moodle/mod/quiz/view.php?id=325) of shapes.

·         When he says rectangle he is actually referring to a square.

·         When he says square, he is actually referring to a triangle.

·         When he says triangle he is referring to a rectangle

·         And when he is confused, he just says something random. At this point, all you can do is say 0.

Help Suppandi by printing the correct answer in an integer.

Input Format

·         Name of shape (always in upper case R à Rectangle, S à Square, T à Triangle)

·         Length of 1 side

·         Length of other side

Note: In case of triangle, you can consider the sides as height and length of base

Output Format

·         Print the area of the shape.

Sample Input 1

T

10

20

Sample Output 1

200

Sample Input 2

S

30

40

Sample Output 2

600

Sample Input 3

R

10

10

Sample Output 3

100

Sample Input 4

G

8

8

Sample Output 4

0

Sample Input

C

9

10

Sample Output 4

0

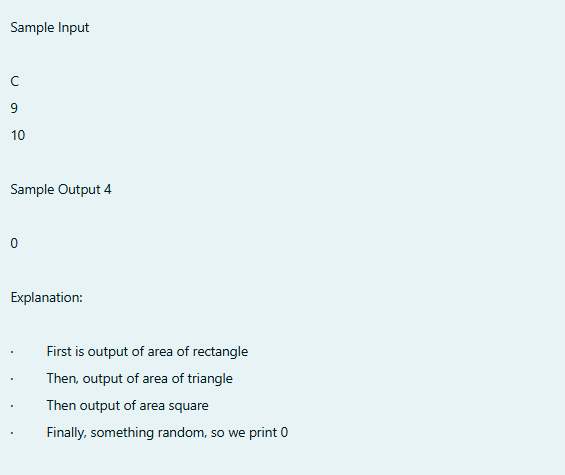
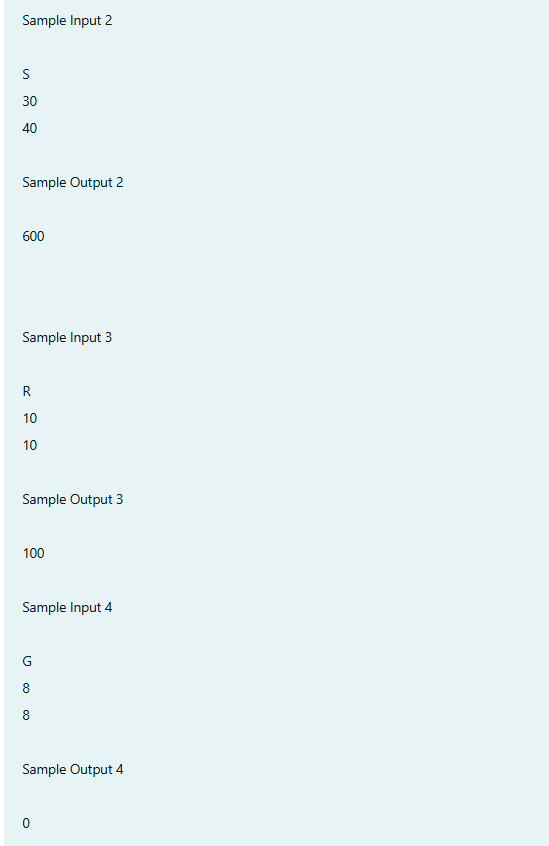
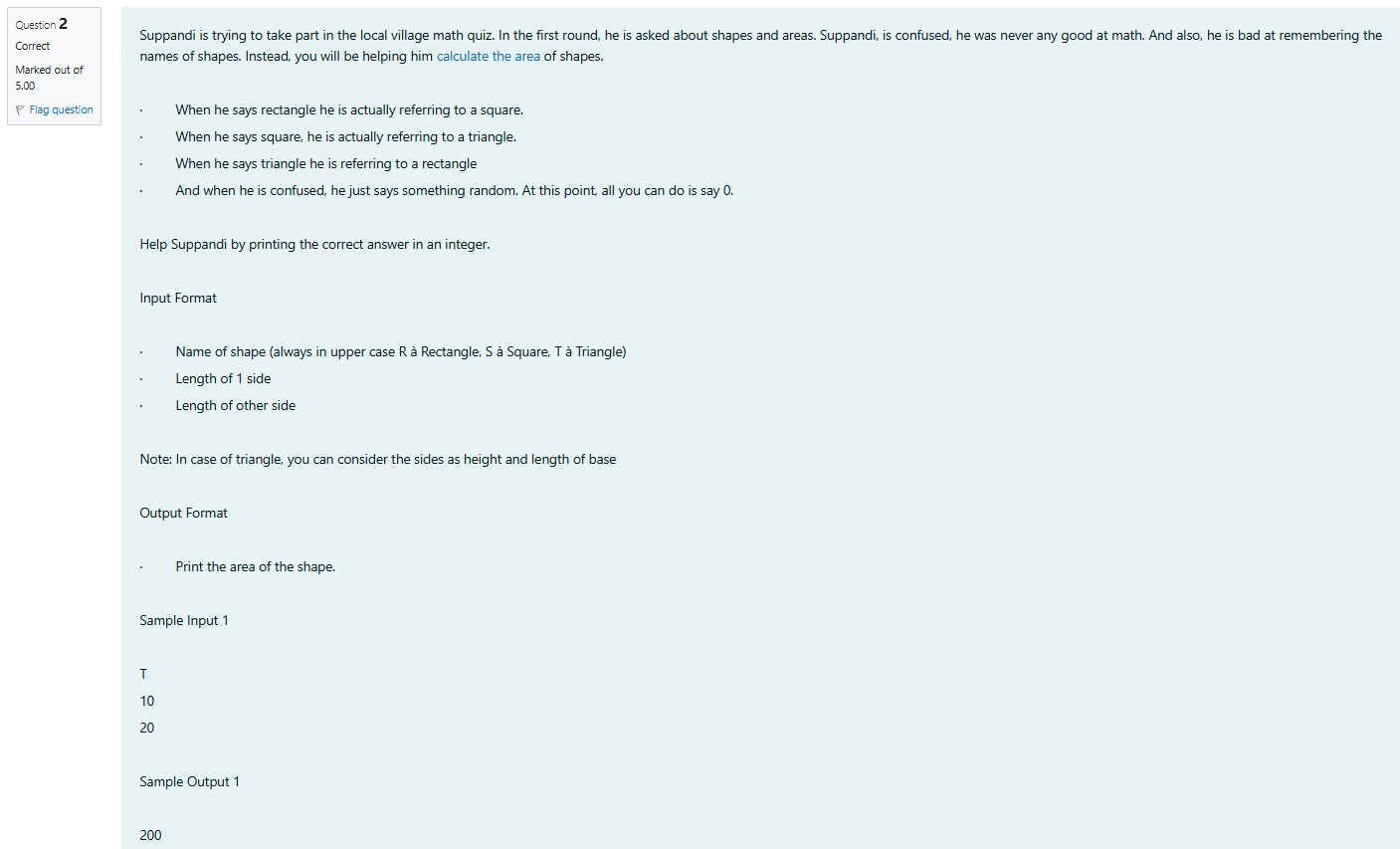
Explanation:

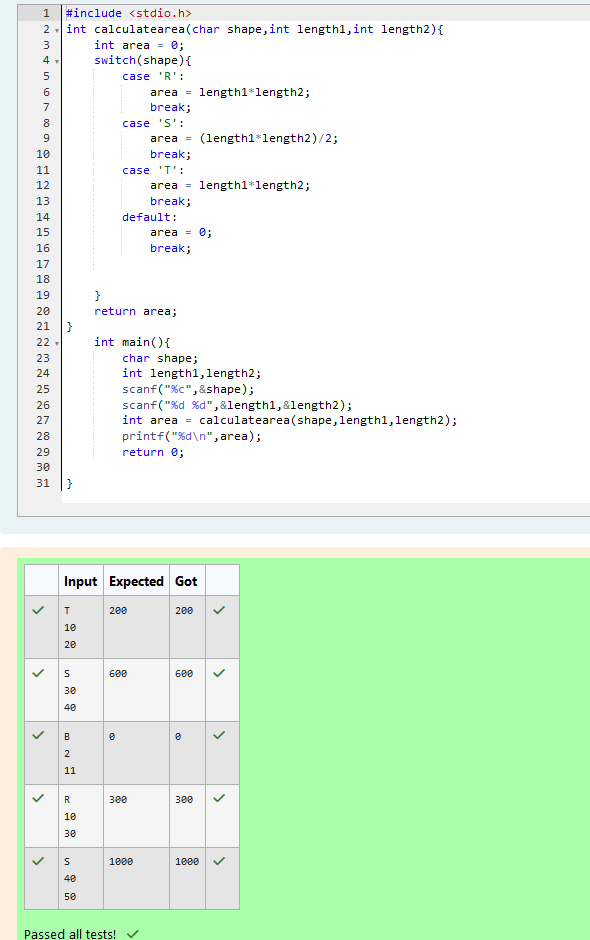
·         First is output of area of rectangle

·         Then, output of area of triangle

·         Then output of area square

·         Finally, something random, so we print 0





Q3) Superman is planning a journey to his home planet. It is very important for him to know which day he arrives there. They don't follow the 7-day week like us. Instead, they follow a 10-day week with the following days: Day Number Name of Day 1 Sunday 2 Monday 3 Tuesday 4 Wednesday 5 Thursday 6 Friday 7 Saturday 8 Kryptonday 9 Coluday 10 Daxamday Here are the rules of the calendar: • The calendar starts with Sunday always. • It has only 296 days. After the 296th day, it goes back to Sunday. You begin your journey on a Sunday and will reach after n. You have to tell on which day you will arrive when you reach there.

Input format: •

 Contain a number n (0 < n)

Output format: Print the name of the day you are arriving on

Example Input

7

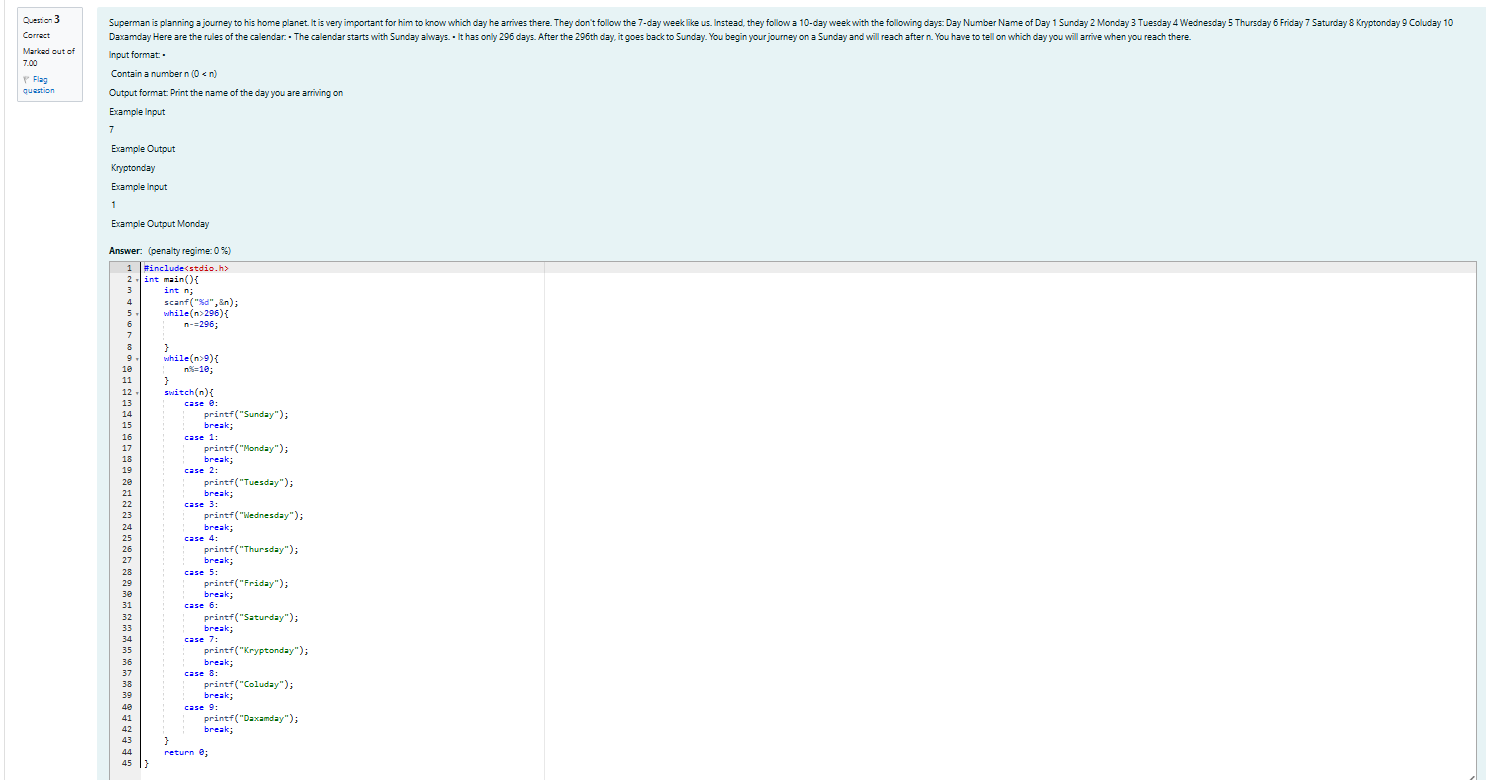
 Example Output

 Kryptonday

 Example Input

 1

 Example Output Monday



Output:



## **Array Applications**

Q1)Write a program to find the largest and second largest elements with in the elements of the given one dimensional array.  
  
For example, if the user gives the **input** as:

5

Next, the program should print the messages one by one on the console.

if the user gives the **input** as:

10

50

30

20

25

then the program should **print** the result as:

The largest element of the array = 50

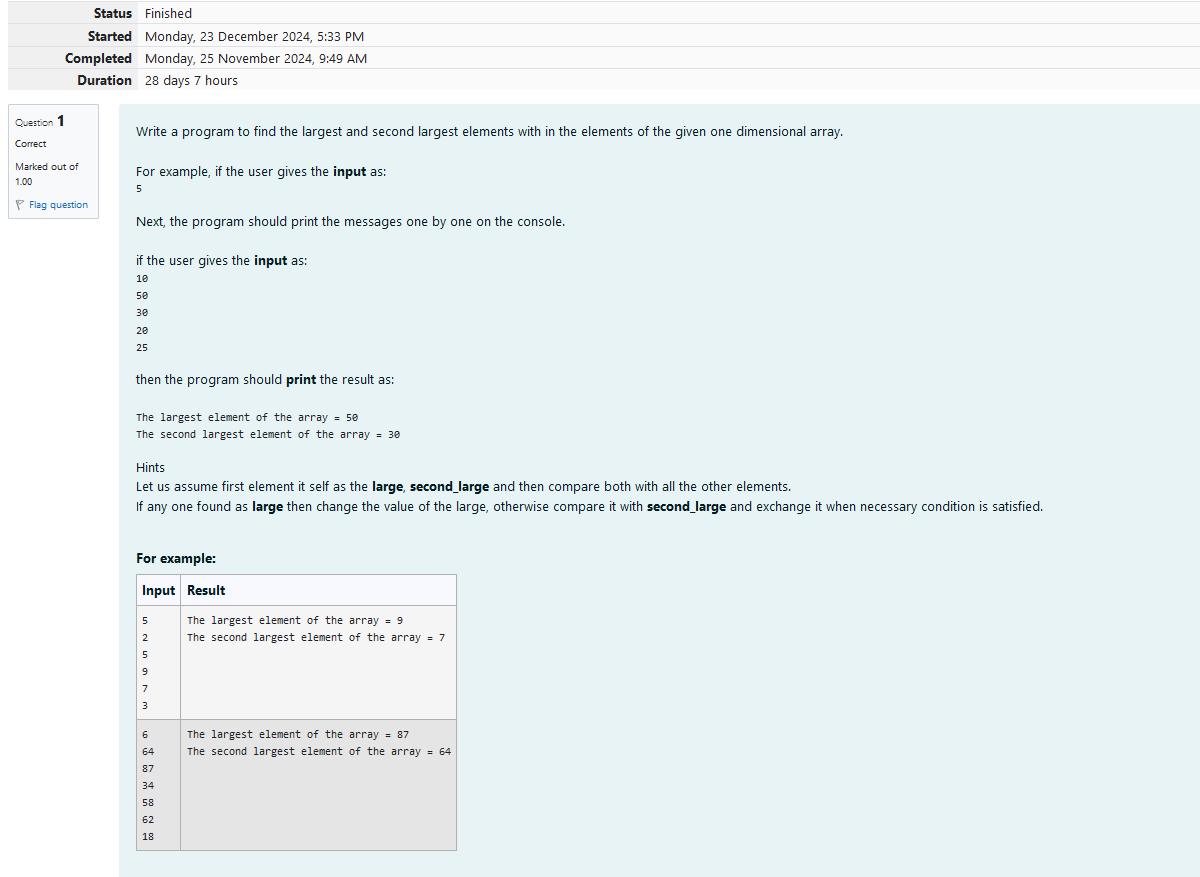
The second largest element of the array = 30

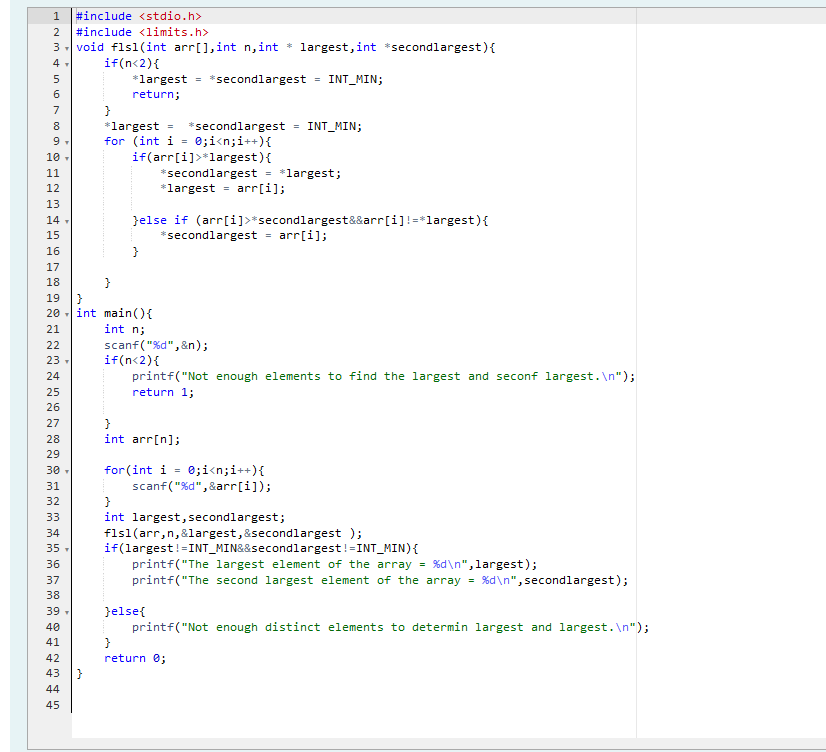
Hints

Let us assume first element it self as the **large**, **second\_large** and then compare both with all the other elements.  
If any one found as **large** then change the value of the large, otherwise compare it with **second\_large** and exchange it when necessary condition is satisfied.

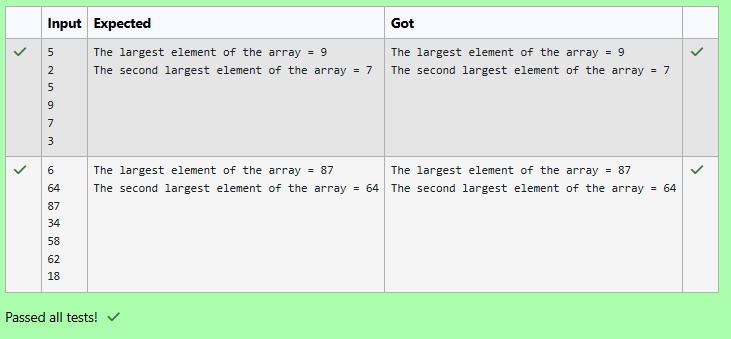
For example:

| Input | Result |
| --- | --- |
| 5  2  5  9  7  3 | The largest element of the array = 9  The second largest element of the array = 7 |
| 6  64  87  34  58  62  18 | The largest element of the array = 87  The second largest element of the array = 64 |





Output:



Q2) Write a program to find the minimum and second minimum elements with in the elements of one dimensional array.

**Constraints:**

* 1 <= N <= 103
* 1 <= Elements of the array <= 106

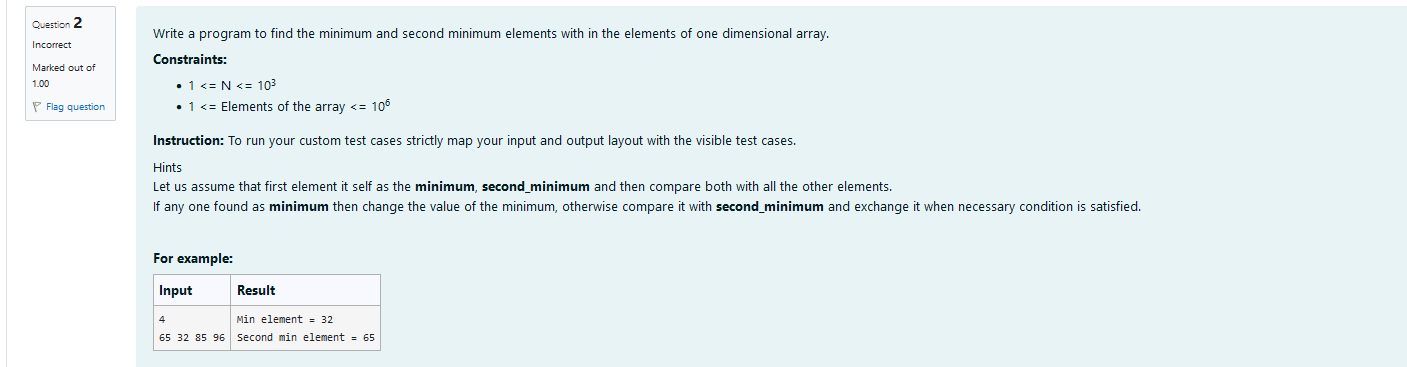
**﻿Instruction:** To run your custom test cases strictly map your input and output layout with the visible test cases.

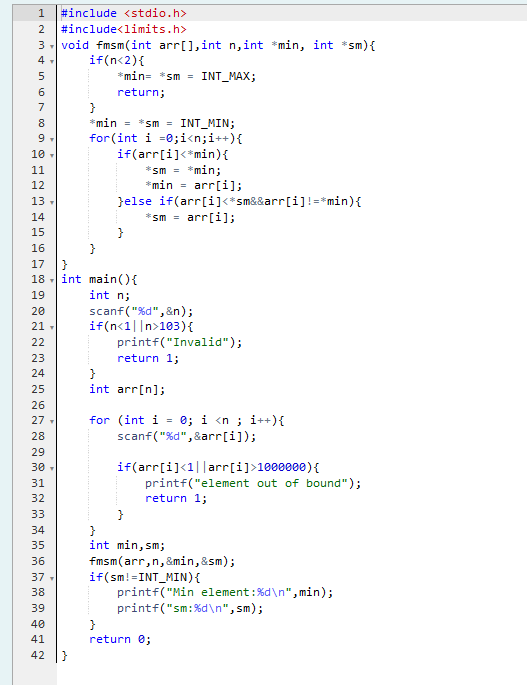
Hints

Let us assume that first element it self as the **minimum**, **second\_minimum** and then compare both with all the other elements.  
If any one found as **minimum** then change the value of the minimum, otherwise compare it with **second\_minimum** and exchange it when necessary condition is satisfied.

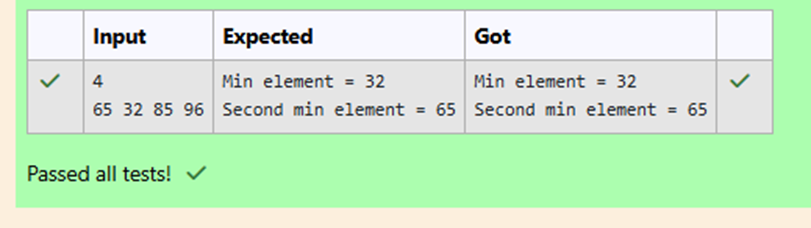
For example:

| Input | Result |
| --- | --- |
| 4  65 32 85 96 | Min element = 32  Second min element = 65 |





Output:



Q3)Write a program to read a student n subjects marks in an array and find the total, average of the marks.  
  
For example, if the user gives the **input** as:

3

Next, the program should print the messages one by one on the console.

if the user gives the **input** as:

75

80

85

then the program should **print** the result as:

The total marks = 240

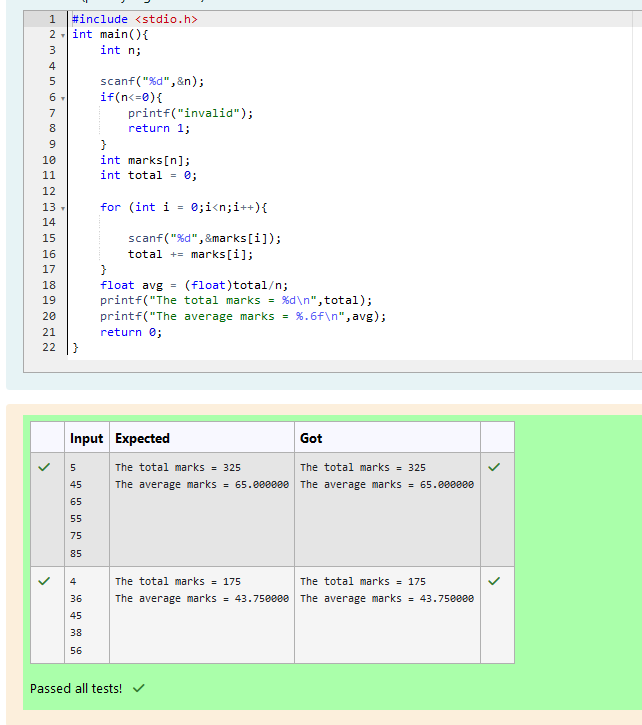
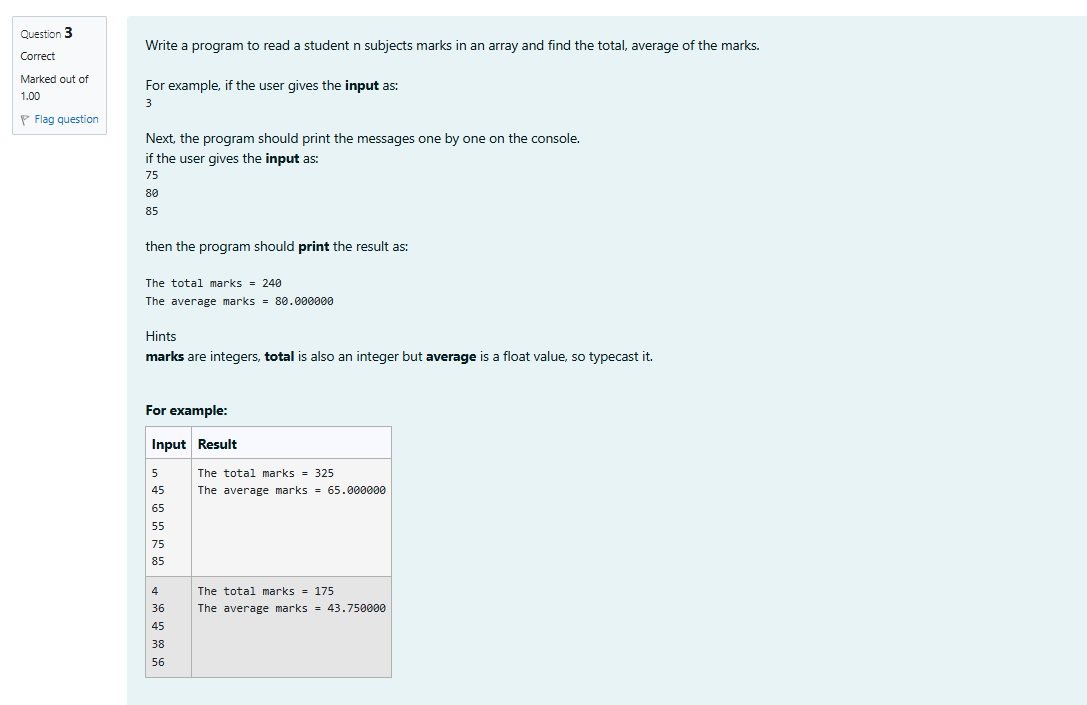
The average marks = 80.000000

Hints

**marks** are integers, **total** is also an integer but **average** is a float value, so typecast it.

For example:

| Input | Result |
| --- | --- |
| 5  45  65  55  75  85 | The total marks = 325  The average marks = 65.000000 |
| 4  36  45  38  56 | The total marks = 175  The average marks = 43.750000 |

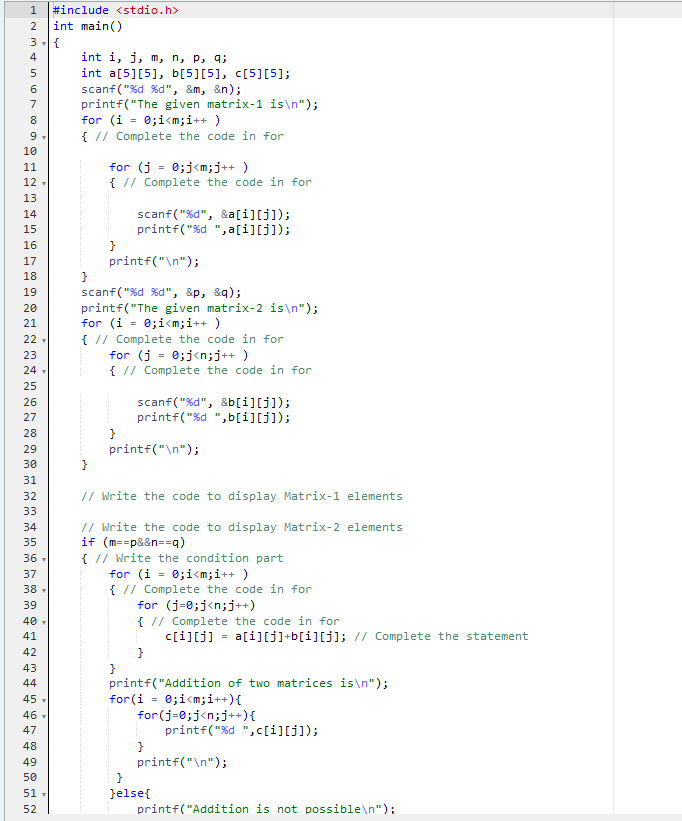
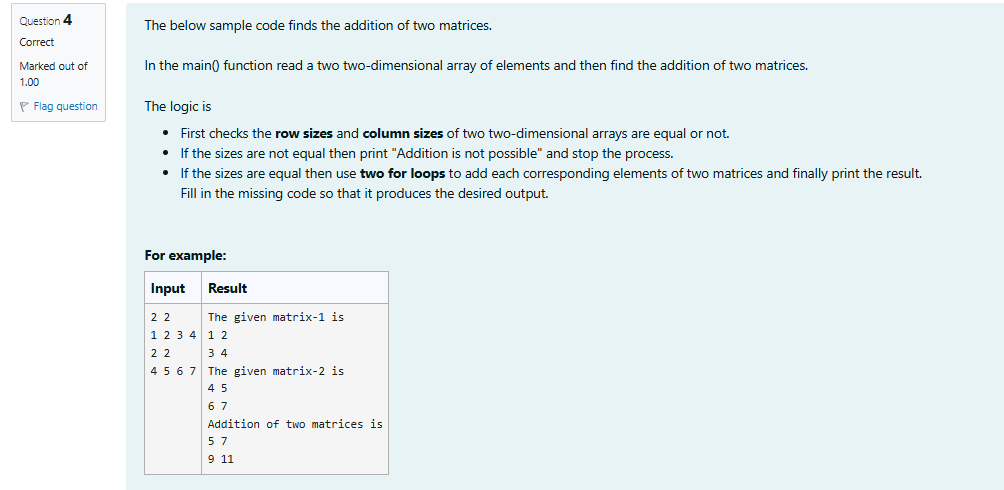


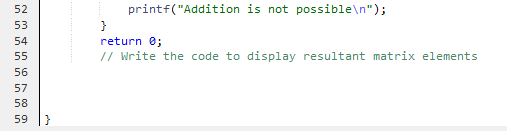
Q4)The below sample code finds the addition of two matrices.  
  
In the main() function read a two two-dimensional array of elements and then find the addition of two matrices.  
  
The logic is

* First checks the **row sizes** and **column sizes** of two two-dimensional arrays are equal or not.
* If the sizes are not equal then print "Addition is not possible" and stop the process.
* If the sizes are equal then use **two for loops** to add each corresponding elements of two matrices and finally print the result.Fill in the missing code so that it produces the desired output.

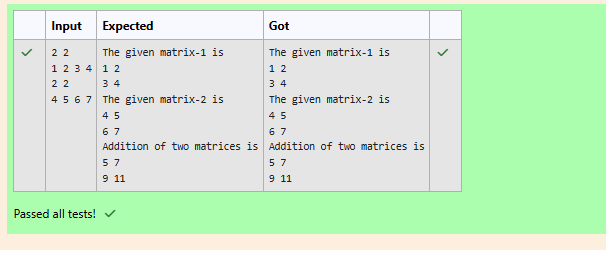
For example:

| Input | Result |
| --- | --- |
| 2 2  1 2 3 4  2 2  4 5 6 7 | The given matrix-1 is  1 2  3 4  The given matrix-2 is  4 5  6 7  Addition of two matrices is  5 7  9 11 |





Ouput:



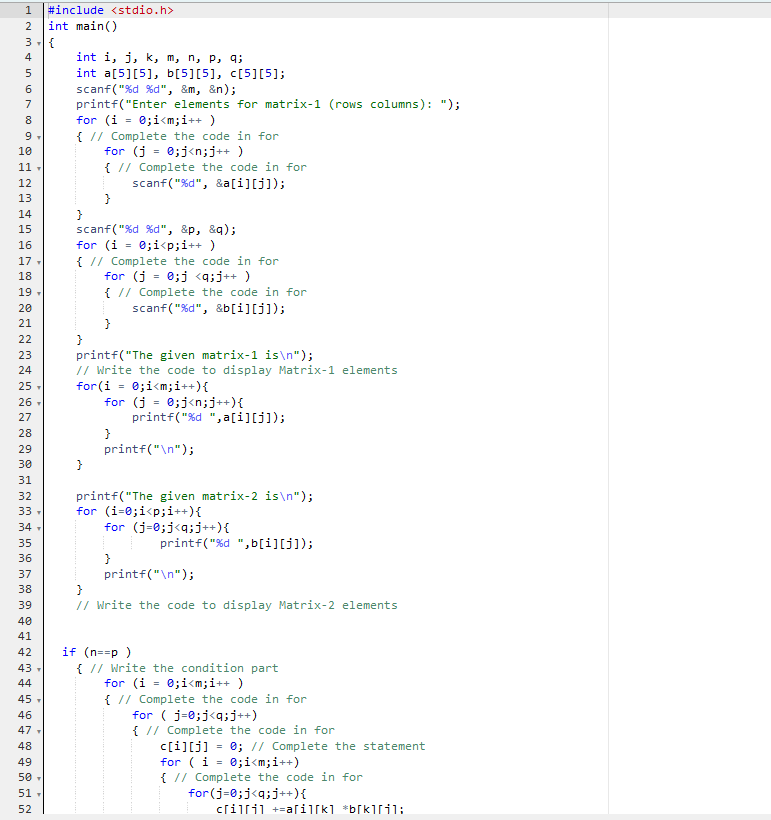
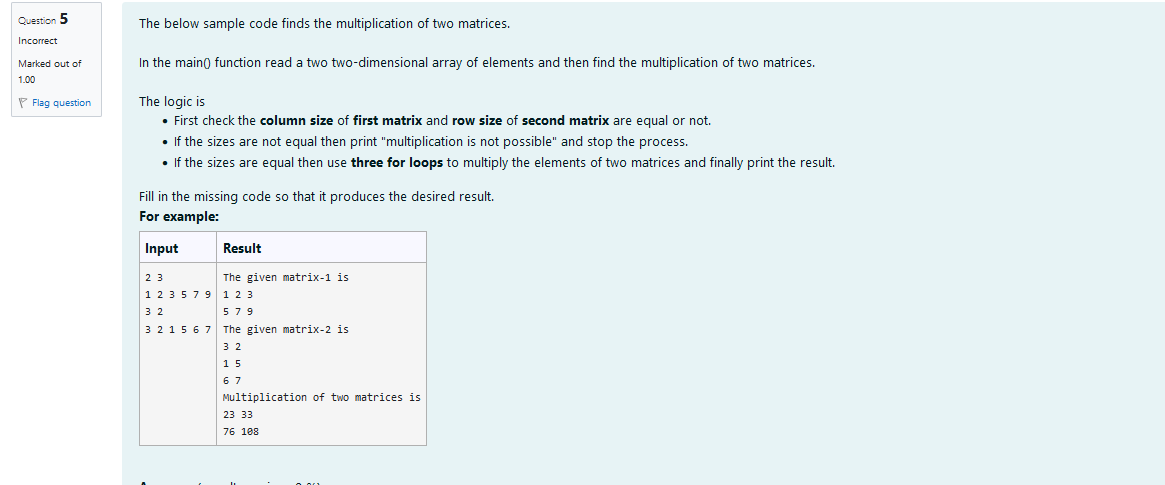
Q5) The below sample code finds the multiplication of two matrices.  
  
In the main() function read a two two-dimensional array of elements and then find the multiplication of two matrices.  
  
The logic is

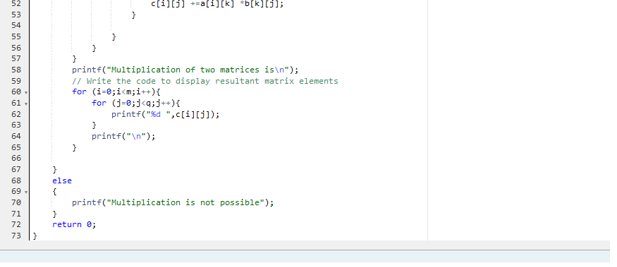
* First check the **column size** of **first matrix** and **row size** of **second matrix** are equal or not.
* If the sizes are not equal then print "multiplication is not possible" and stop the process.
* If the sizes are equal then use **three for loops** to multiply the elements of two matrices and finally print the result.

Fill in the missing code so that it produces the desired result.

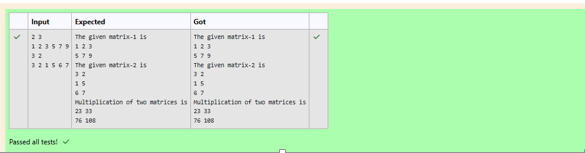
For example:

| Input | Result |
| --- | --- |
| 2 3  1 2 3 5 7 9  3 2  3 2 1 5 6 7 | The given matrix-1 is  1 2 3  5 7 9  The given matrix-2 is  3 2  1 5  6 7  Multiplication of two matrices is  23 33  76 108 |





Output:

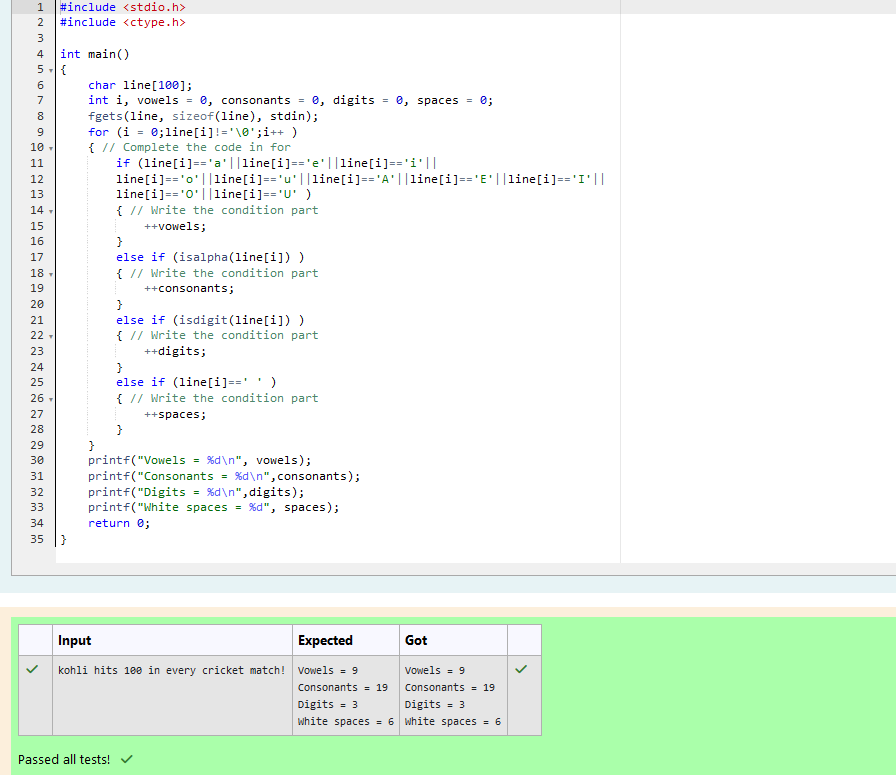
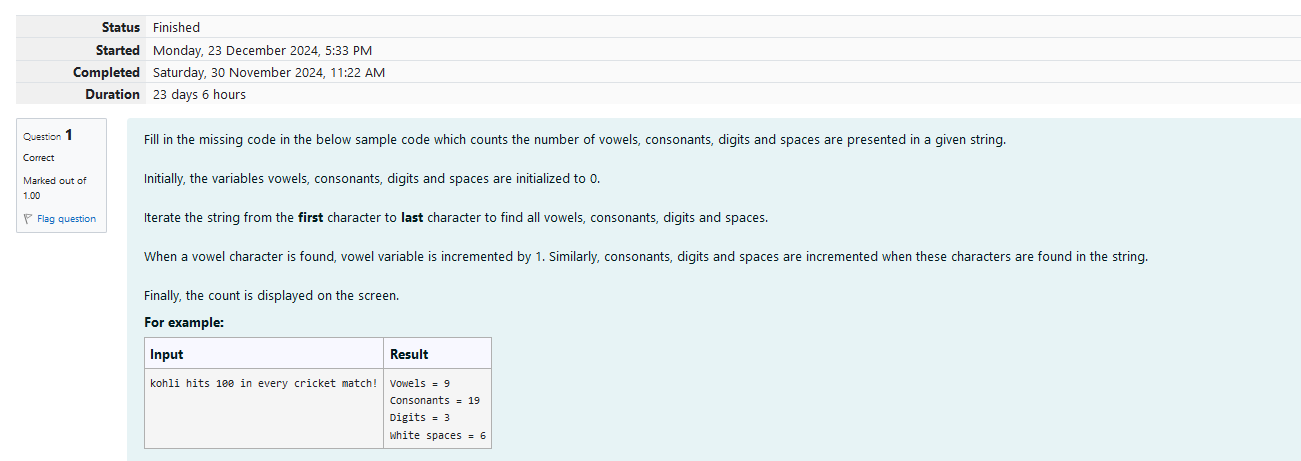


## **Problem solving with Strings**

Q1) Fill in the missing code in the below sample code which counts the number of vowels, consonants, digits and spaces are presented in a given string.  
  
Initially, the variables vowels, consonants, digits and spaces are initialized to 0.  
  
Iterate the string from the **first** character to **last** character to find all vowels, consonants, digits and spaces.  
  
When a vowel character is found, vowel variable is incremented by 1. Similarly, consonants, digits and spaces are incremented when these characters are found in the string.  
  
Finally, the count is displayed on the screen.

For example:

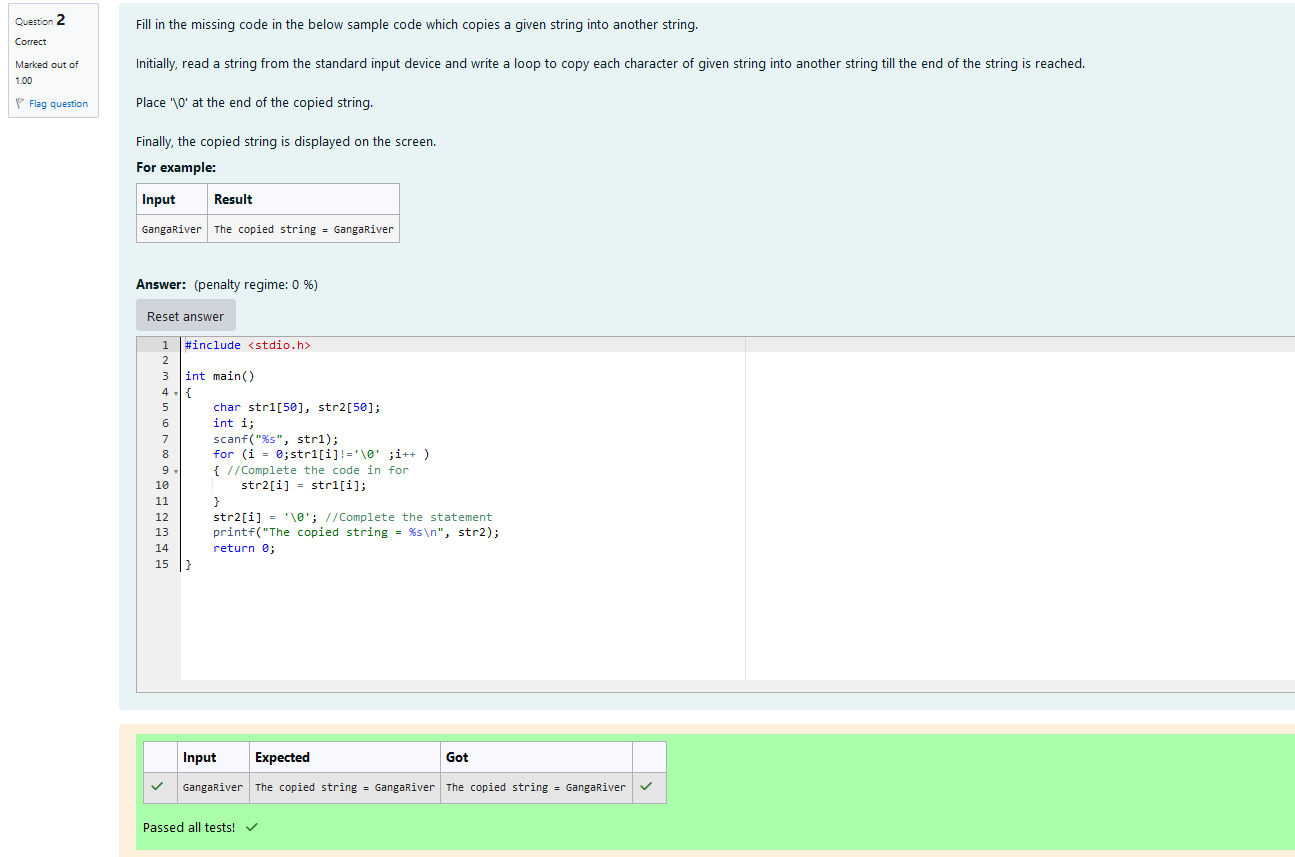
| Input | Result |
| --- | --- |
| kohli hits 100 in every cricket match! | Vowels = 9  Consonants = 19  Digits = 3  White spaces = 6 |



Q2) Fill in the missing code in the below sample code which copies a given string into another string.  
  
Initially, read a string from the standard input device and write a loop to copy each character of given string into another string till the end of the string is reached.  
  
Place '\0' at the end of the copied string.  
  
Finally, the copied string is displayed on the screen.

For example:

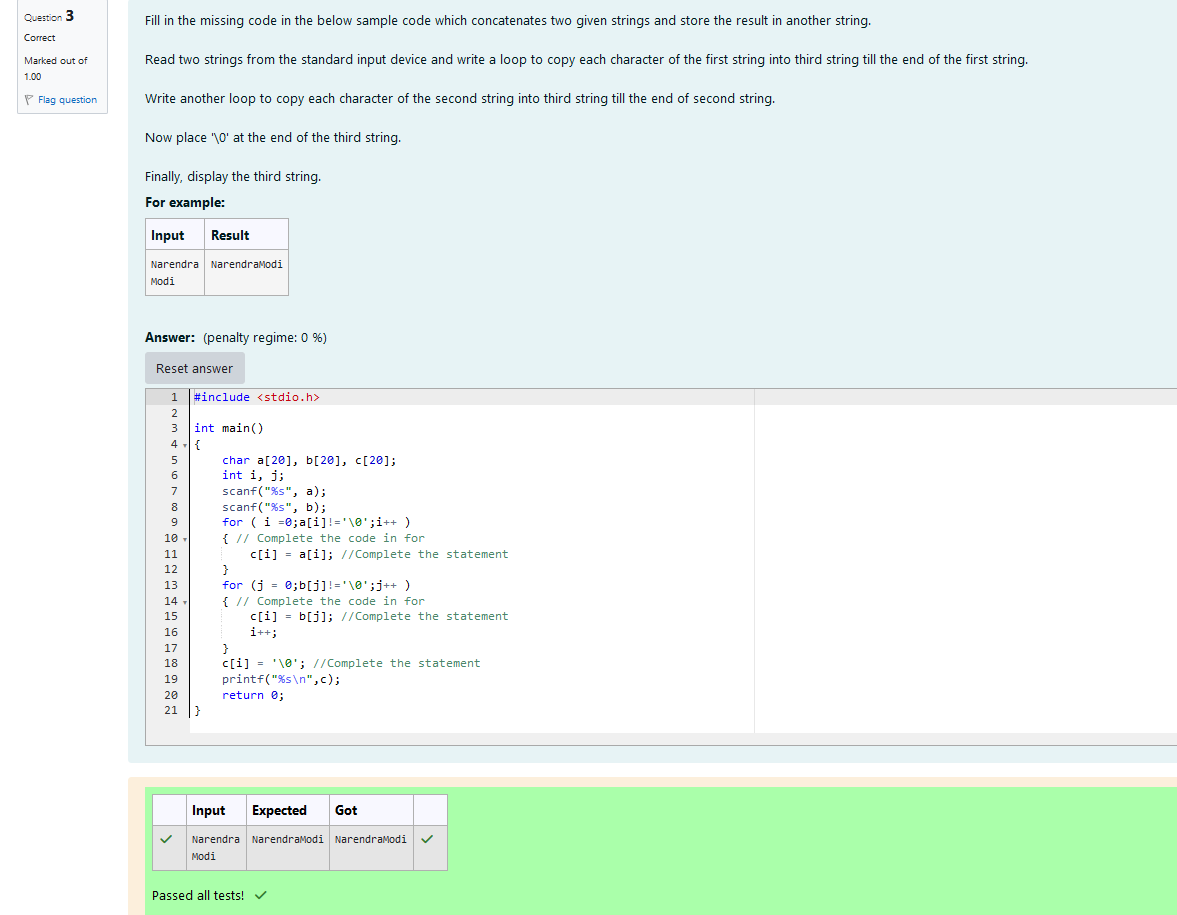
| Input | Result |
| --- | --- |
| GangaRiver | The copied string = GangaRiver |



Q3) Fill in the missing code in the below sample code which concatenates two given strings and store the result in another string.  
  
Read two strings from the standard input device and write a loop to copy each character of the first string into third string till the end of the first string.  
  
Write another loop to copy each character of the second string into third string till the end of second string.  
  
Now place '\0' at the end of the third string.  
  
Finally, display the third string.

For example:

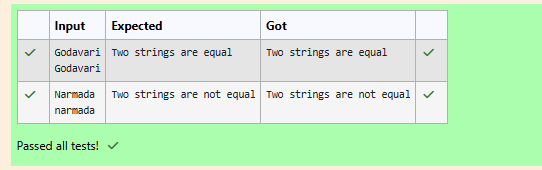
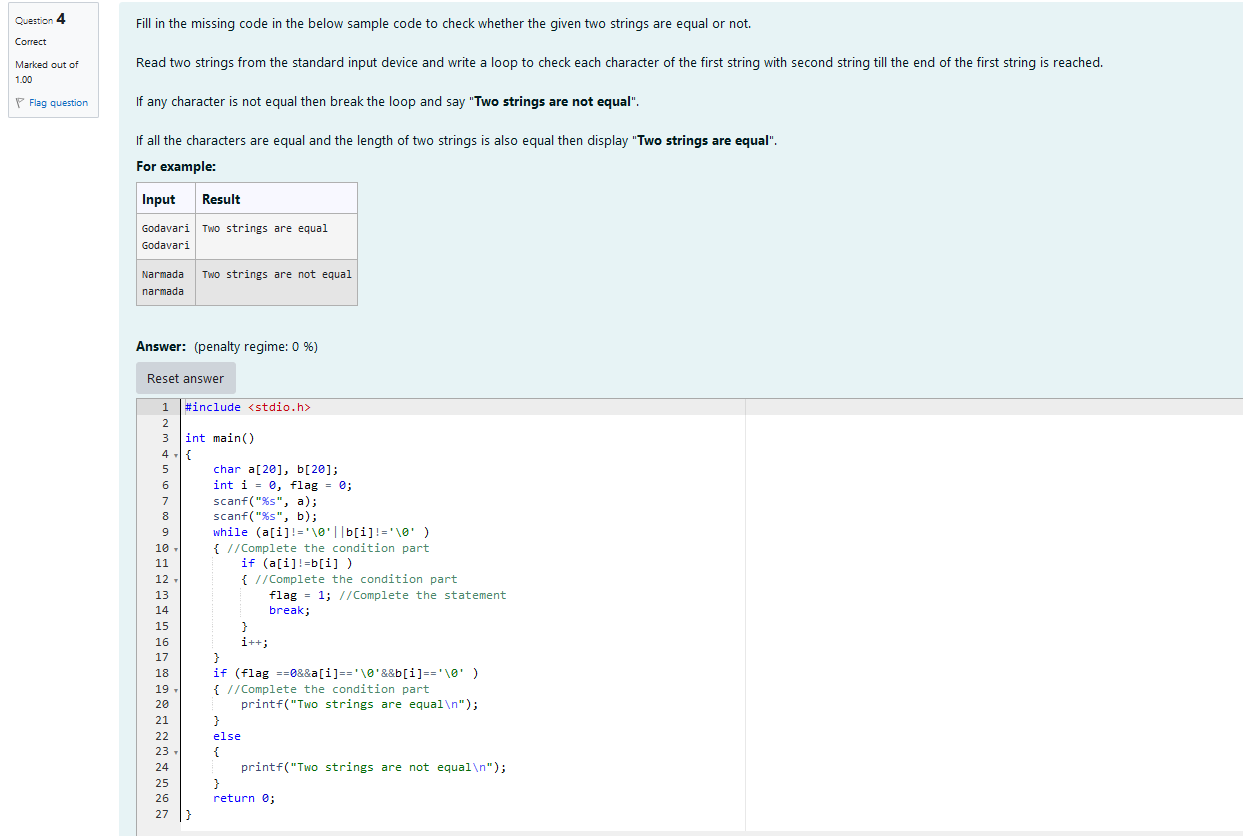
| Input | Result |
| --- | --- |
| Narendra  Modi | NarendraModi |



Q4)Fill in the missing code in the below sample code to check whether the given two strings are equal or not.  
  
Read two strings from the standard input device and write a loop to check each character of the first string with second string till the end of the first string is reached.  
  
If any character is not equal then break the loop and say "**Two strings are not equal**".  
  
If all the characters are equal and the length of two strings is also equal then display "**Two strings are equal**".

For example:

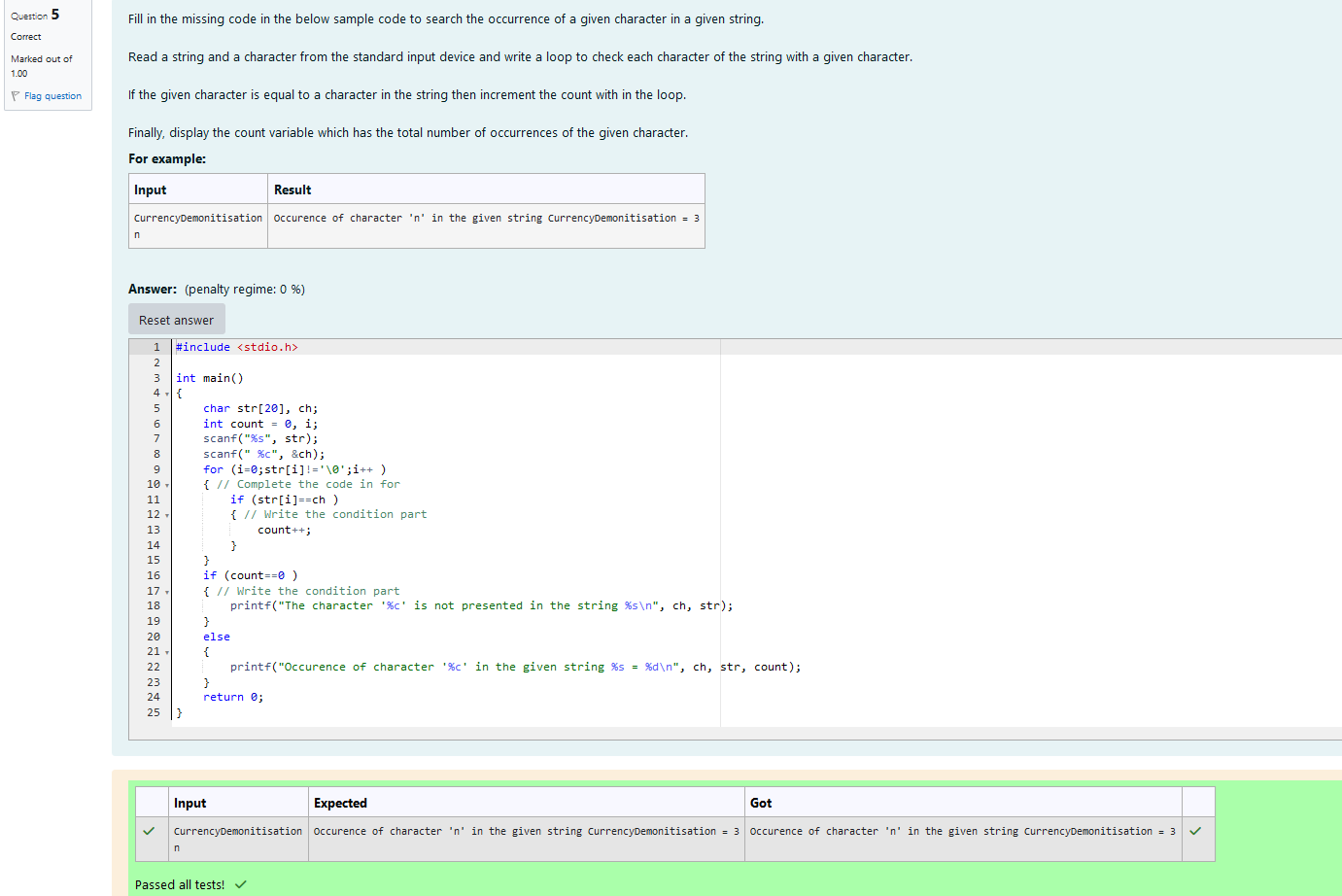
| Input | Result |
| --- | --- |
| Godavari  Godavari | Two strings are equal |
| Narmada  narmada | Two strings are not equal |



Q5) Fill in the missing code in the below sample code to search the occurrence of a given character in a given string.  
  
Read a string and a character from the standard input device and write a loop to check each character of the string with a given character.  
  
If the given character is equal to a character in the string then increment the count with in the loop.  
  
Finally, display the count variable which has the total number of occurrences of the given character.

For example:

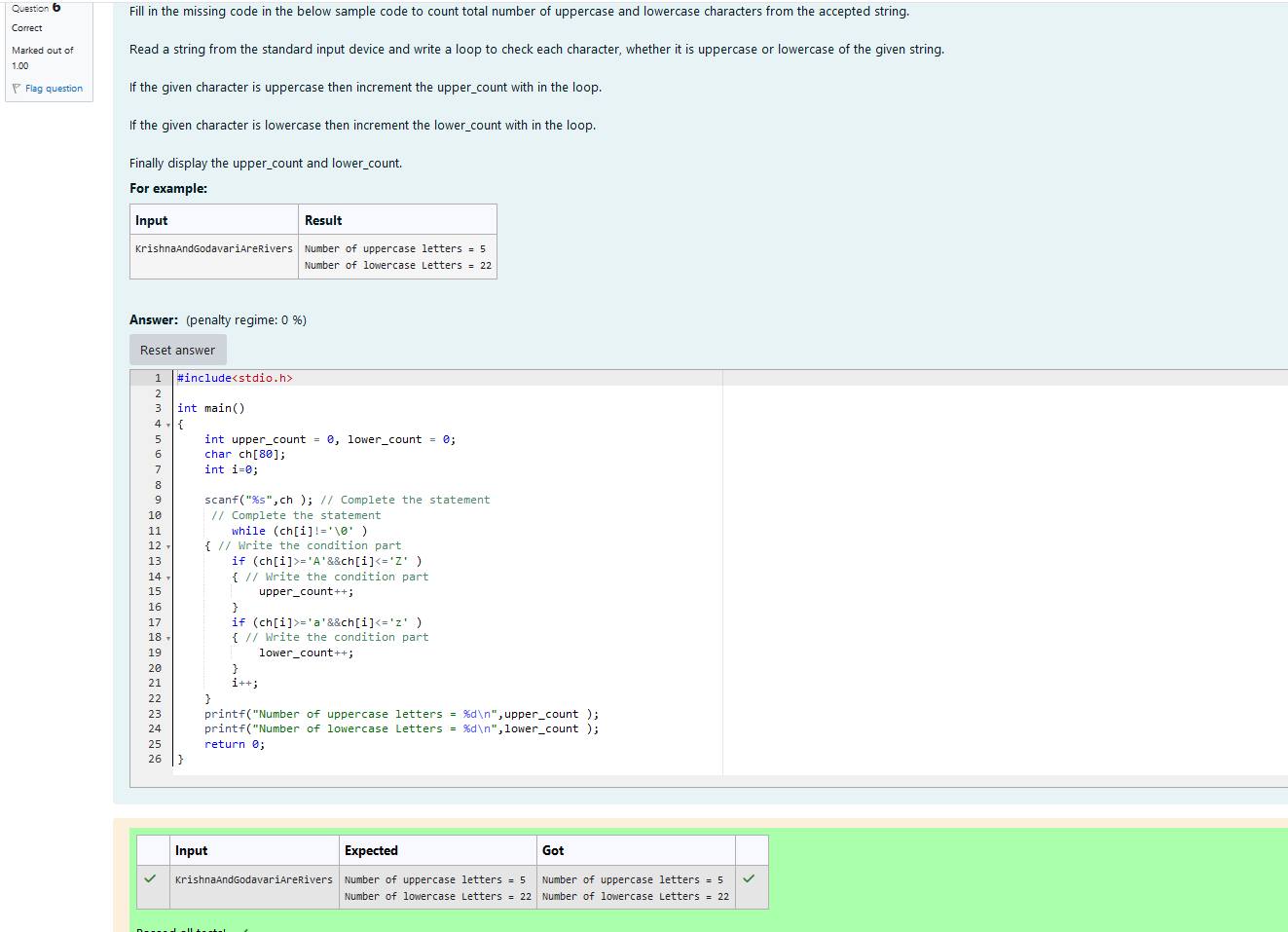
| Input | Result |
| --- | --- |
| CurrencyDemonitisation  n | Occurence of character 'n' in the given string CurrencyDemonitisation = 3 |



Q6) Fill in the missing code in the below sample code to count total number of uppercase and lowercase characters from the accepted string.  
  
Read a string from the standard input device and write a loop to check each character, whether it is uppercase or lowercase of the given string.  
  
If the given character is uppercase then increment the upper\_count with in the loop.  
  
If the given character is lowercase then increment the lower\_count with in the loop.  
  
Finally display the upper\_count and lower\_count.

For example:

| Input | Result |
| --- | --- |
| KrishnaAndGodavariAreRivers | Number of uppercase letters = 5  Number of lowercase Letters = 22 |



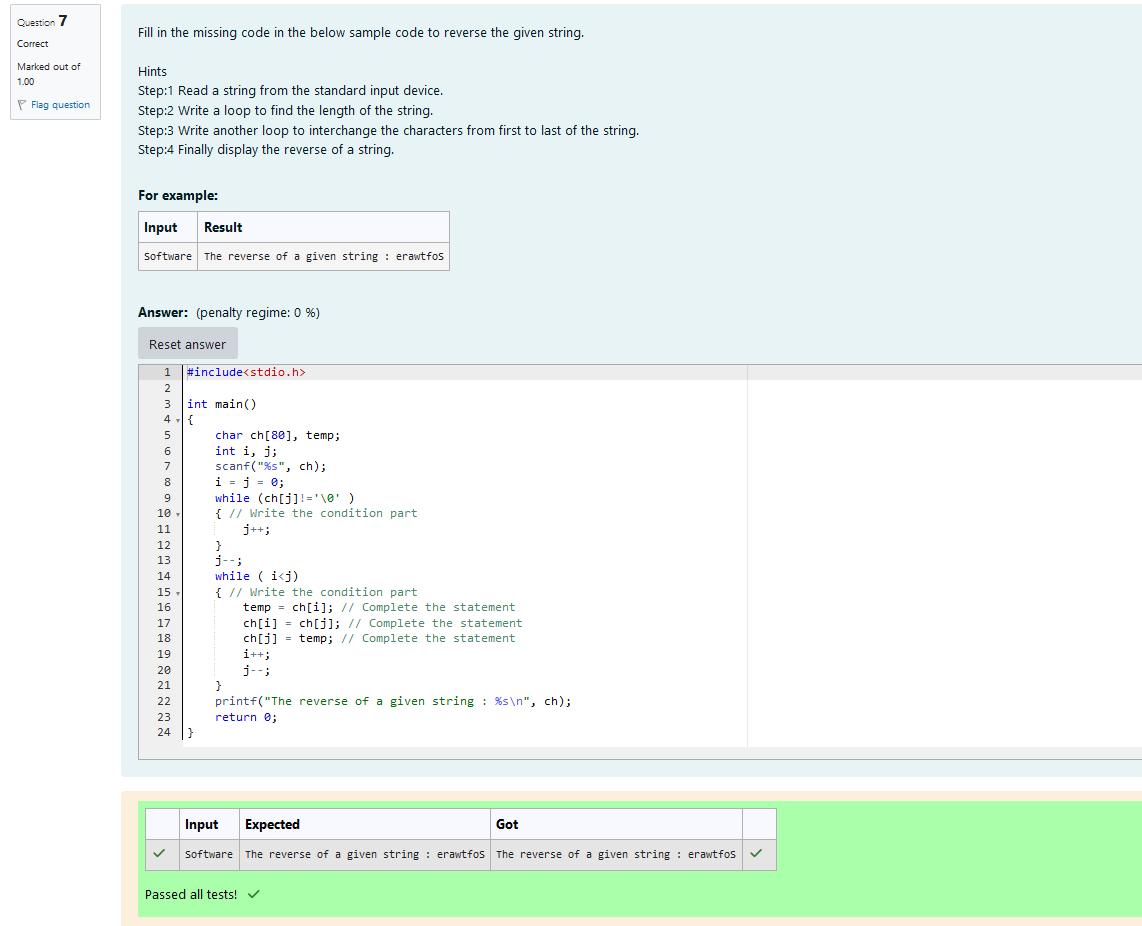
Q7) Fill in the missing code in the below sample code to reverse the given string.

Hints

Step:1 Read a string from the standard input device.  
Step:2 Write a loop to find the length of the string.  
Step:3 Write another loop to interchange the characters from first to last of the string.  
Step:4 Finally display the reverse of a string.

For example:

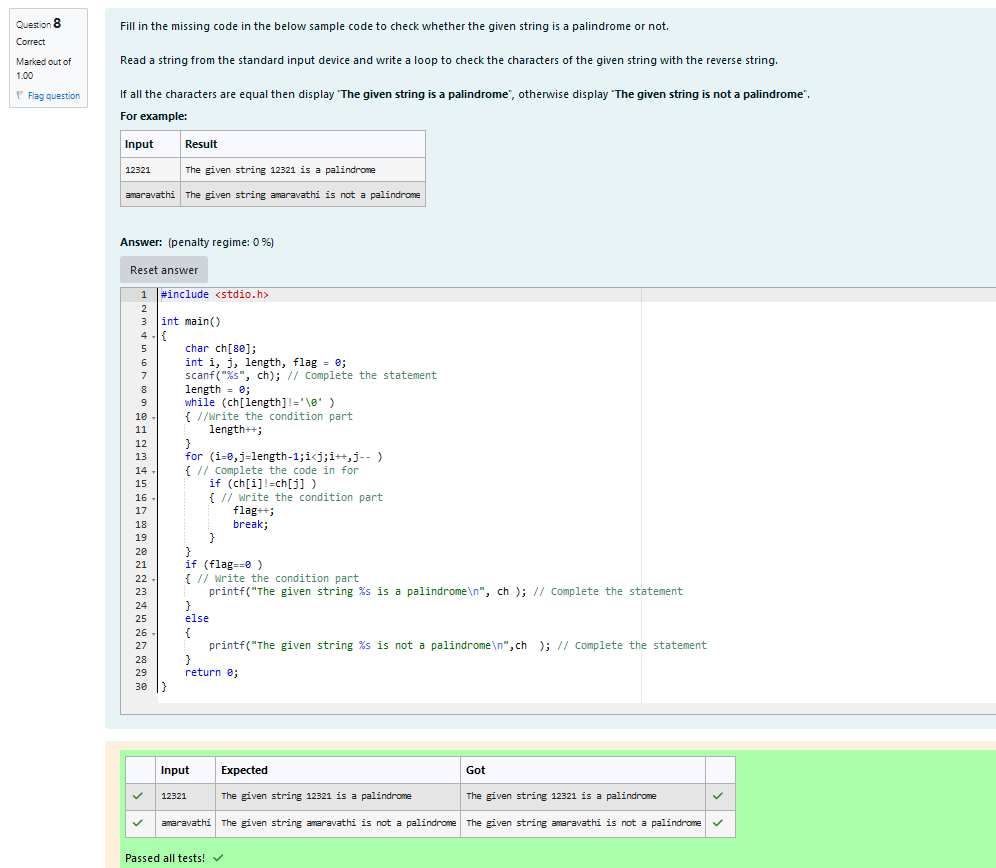
| Input | Result |
| --- | --- |
| Software | The reverse of a given string : erawtfoS |



Q8) Fill in the missing code in the below sample code to check whether the given string is a palindrome or not.  
  
Read a string from the standard input device and write a loop to check the characters of the given string with the reverse string.  
  
If all the characters are equal then display "**The given string is a palindrome**", otherwise display "**The given string is not a palindrome**".

For example:

| Input | Result |
| --- | --- |
| 12321 | The given string 12321 is a palindrome |
| amaravathi | The given string amaravathi is not a palindrome |

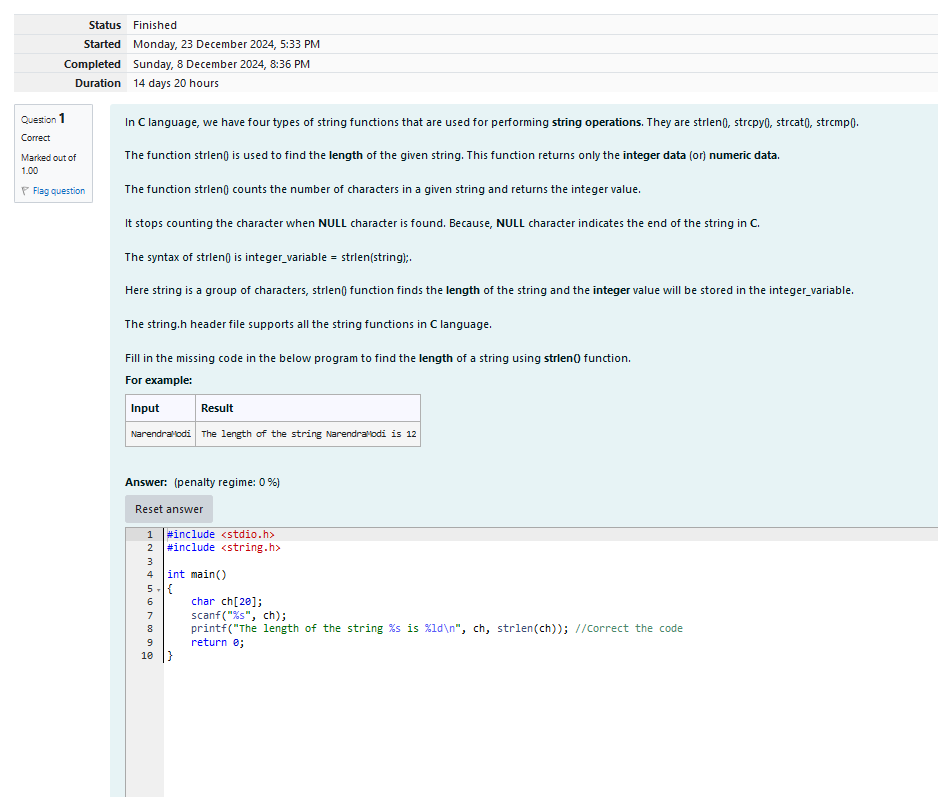


## **String manipulation functions**

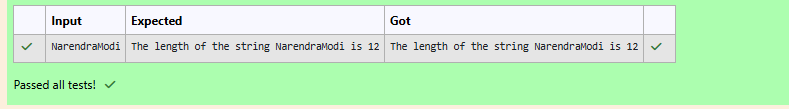
Q1) In **C** language, we have four types of string functions that are used for performing **string operations**. They are strlen(), strcpy(), strcat(), strcmp().  
  
The function strlen() is used to find the **length** of the given string. This function returns only the **integer data** (or) **numeric data**.  
  
The function strlen() counts the number of characters in a given string and returns the integer value.  
  
It stops counting the character when **NULL** character is found. Because, **NULL** character indicates the end of the string in **C**.  
  
The syntax of strlen() is integer\_variable = strlen(string);.  
  
Here string is a group of characters, strlen() function finds the **length** of the string and the **integer** value will be stored in the integer\_variable.  
  
The string.h header file supports all the string functions in **C** language.  
  
Fill in the missing code in the below program to find the **length** of a string using **strlen()** function.

For example:

| Input | Result |
| --- | --- |
| NarendraModi | The length of the string NarendraModi is 12 |



Ouptut:



Q2) The function strcpy() is used to **copy** one string into another string including the NULL character (terminator char '\0').  
  
The syntax of strcpy( ) is strcpy(string1, string2);.  
  
Where string1, string2 are two strings and the string2 is copied into string1. In this case the copied string is available in string1 and both strings contains the same data.  
  
If the length of string1 is less than the length of string2 then entire string2 value will not be copied into string1.  
  
For example, consider the length of string1 is **20** and the length of string2 is **30**. Then, only the first **20** characters from string2 will be copied into string1, the remaining **10** characters will not be copied and will be **truncated**.  
  
Understand and retype the below code which demonstrates the usage of **strcpy()** function.

#include <stdio.h>

#include <string.h>

int main()

{

    char str1[20], str2[20];

    scanf("%s", str2);

    strcpy(str1, str2);

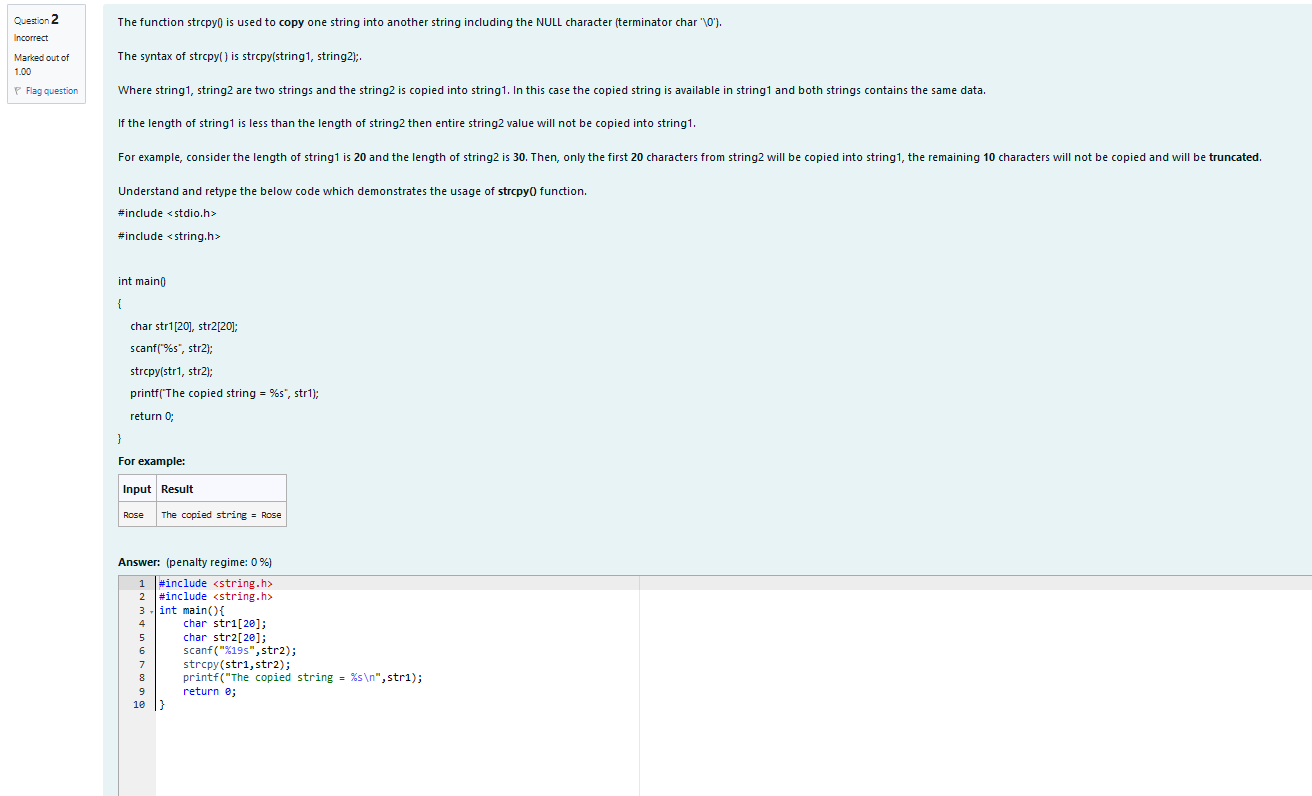
    printf("The copied string = %s", str1);

    return 0;

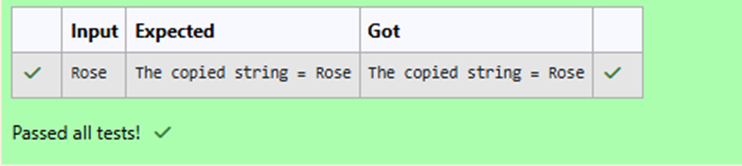
}

For example:

| Input | Result |
| --- | --- |
| Rose | The copied string = Rose |



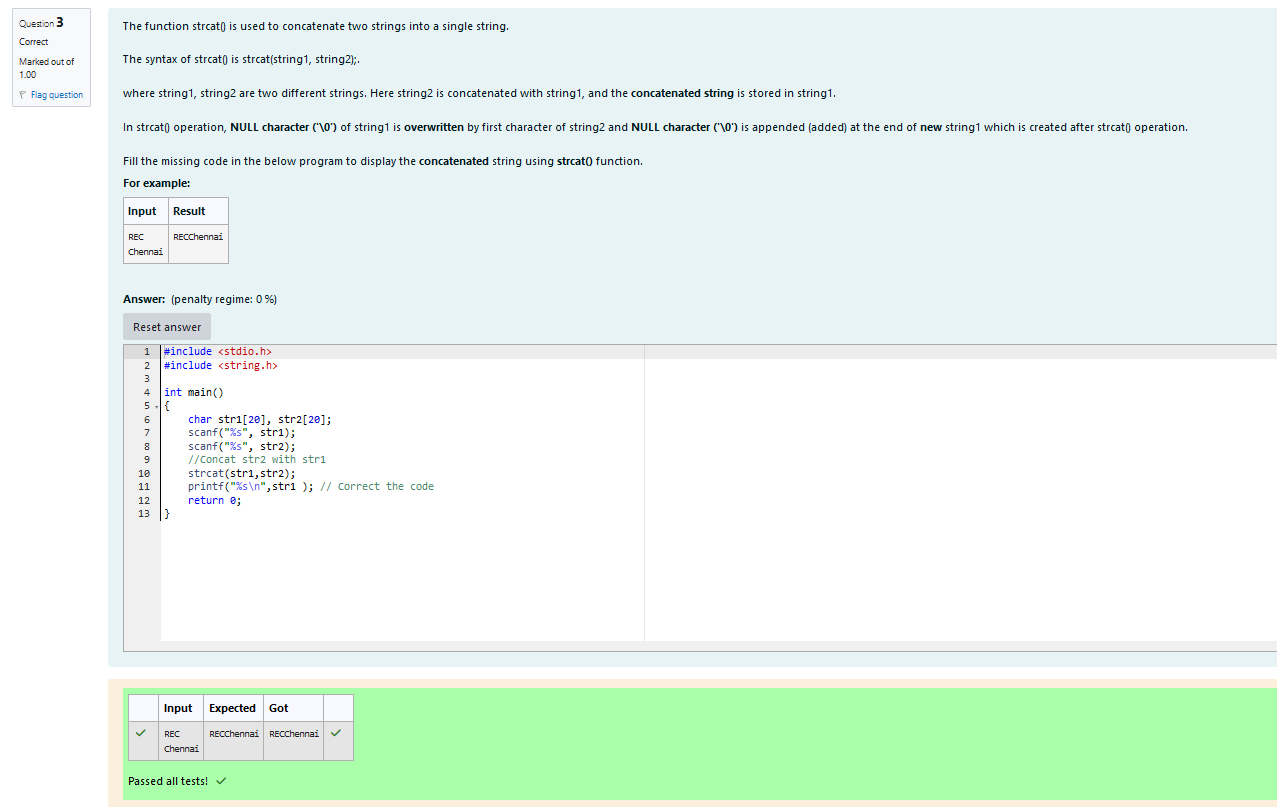
Ouptut:



Q3) The function strcat() is used to concatenate two strings into a single string.  
  
The syntax of strcat() is strcat(string1, string2);.  
  
where string1, string2 are two different strings. Here string2 is concatenated with string1, and the **concatenated string** is stored in string1.  
  
In strcat() operation, **NULL character ('\0')** of string1 is **overwritten** by first character of string2 and **NULL character ('\0')** is appended (added) at the end of **new** string1 which is created after strcat() operation.  
  
Fill the missing code in the below program to display the **concatenated** string using **strcat()** function.

For example:

| Input | Result |
| --- | --- |
| REC  Chennai | RECChennai |

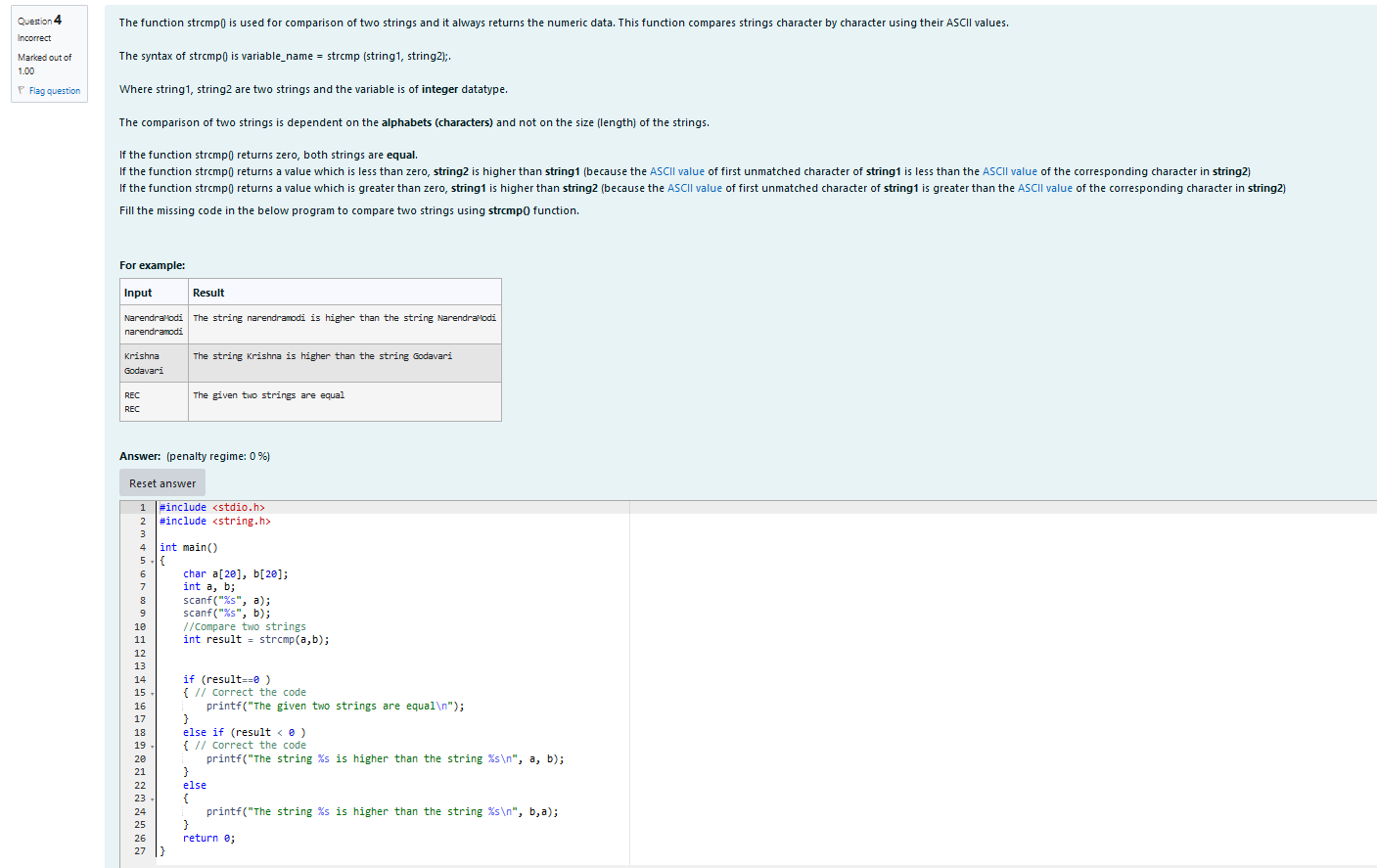


Q4) The function strcmp() is used for comparison of two strings and it always returns the numeric data. This function compares strings character by character using their ASCII values.  
  
The syntax of strcmp() is variable\_name = strcmp (string1, string2);.  
  
Where string1, string2 are two strings and the variable is of **integer** datatype.  
  
The comparison of two strings is dependent on the **alphabets (characters)** and not on the size (length) of the strings.  
  
If the function strcmp() returns zero, both strings are **equal**.  
If the function strcmp() returns a value which is less than zero, **string2** is higher than **string1** (because the [ASCII value](http://www.rajalakshmicolleges.org/moodle/mod/quiz/view.php?id=327) of first unmatched character of **string1** is less than the [ASCII value](http://www.rajalakshmicolleges.org/moodle/mod/quiz/view.php?id=327) of the corresponding character in **string2**)  
If the function strcmp() returns a value which is greater than zero, **string1** is higher than **string2** (because the [ASCII value](http://www.rajalakshmicolleges.org/moodle/mod/quiz/view.php?id=327) of first unmatched character of **string1** is greater than the [ASCII value](http://www.rajalakshmicolleges.org/moodle/mod/quiz/view.php?id=327) of the corresponding character in **string2**)

Fill the missing code in the below program to compare two strings using **strcmp()** function.

For example:

| Input | Result |
| --- | --- |
| NarendraModi  narendramodi | The string narendramodi is higher than the string NarendraModi |
| Krishna  Godavari | The string Krishna is higher than the string Godavari |
| REC  REC | The given two strings are equal |



Ouptut:

