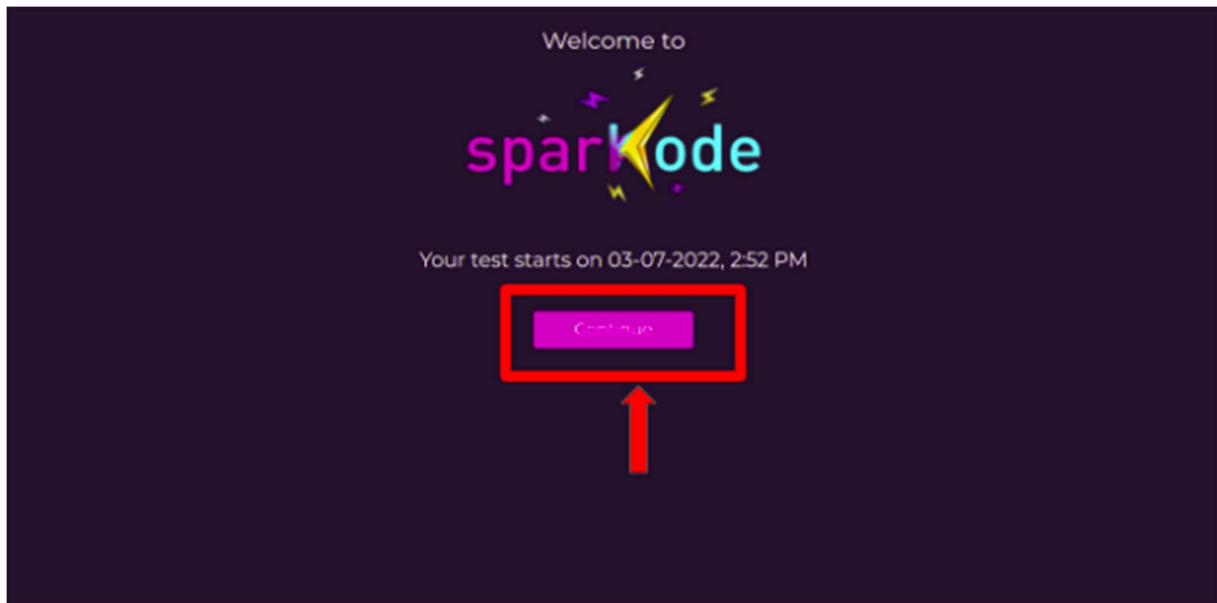


Instructions for Candidates

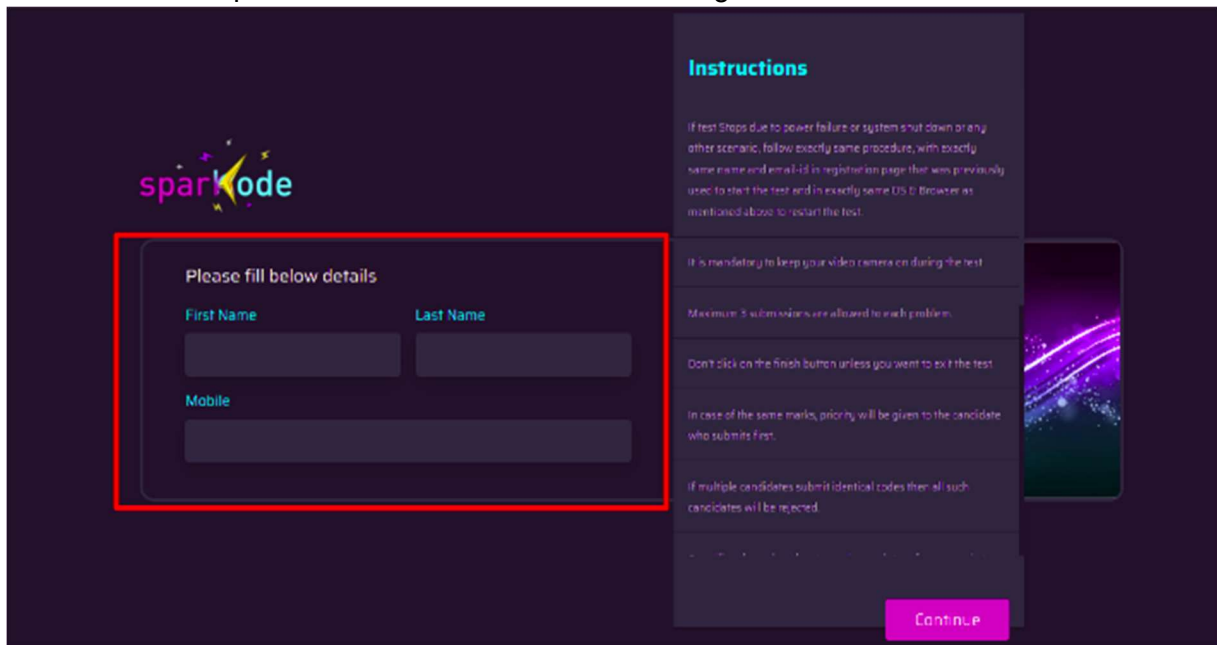
Step 1:

After clicking the invite link that was sent to the user via email, the screen shown below will appear with the date and time and a button to continue.

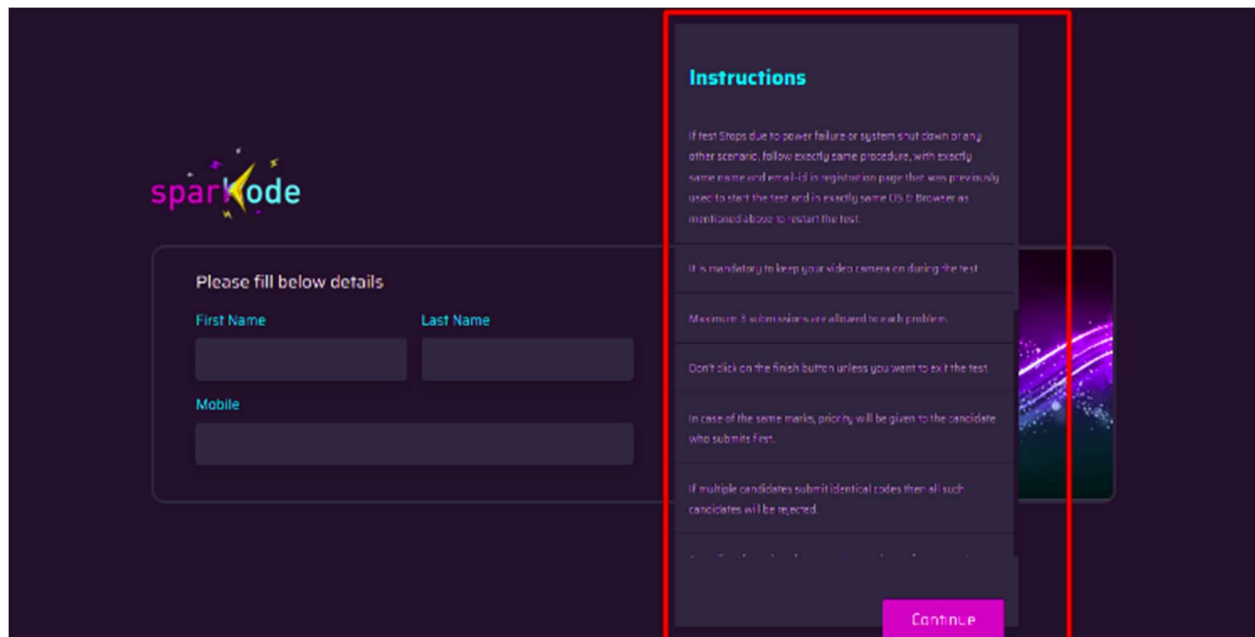


Step 2:

The test screen opens Click on the Continue button to go to the next screen.



Please fill in the details like first and last name and mobile number. The candidate should carefully read the scrollable instructions. Click Continue.



The image shows a registration form on the left and a scrollable instructions panel on the right. The form has fields for First Name, Last Name, and Mobile, with a 'Continue' button at the bottom. The instructions panel lists rules for the test, such as following the same procedure as registration, keeping the video camera on, and not clicking the finish button until ready to exit. A red box highlights the instructions panel.

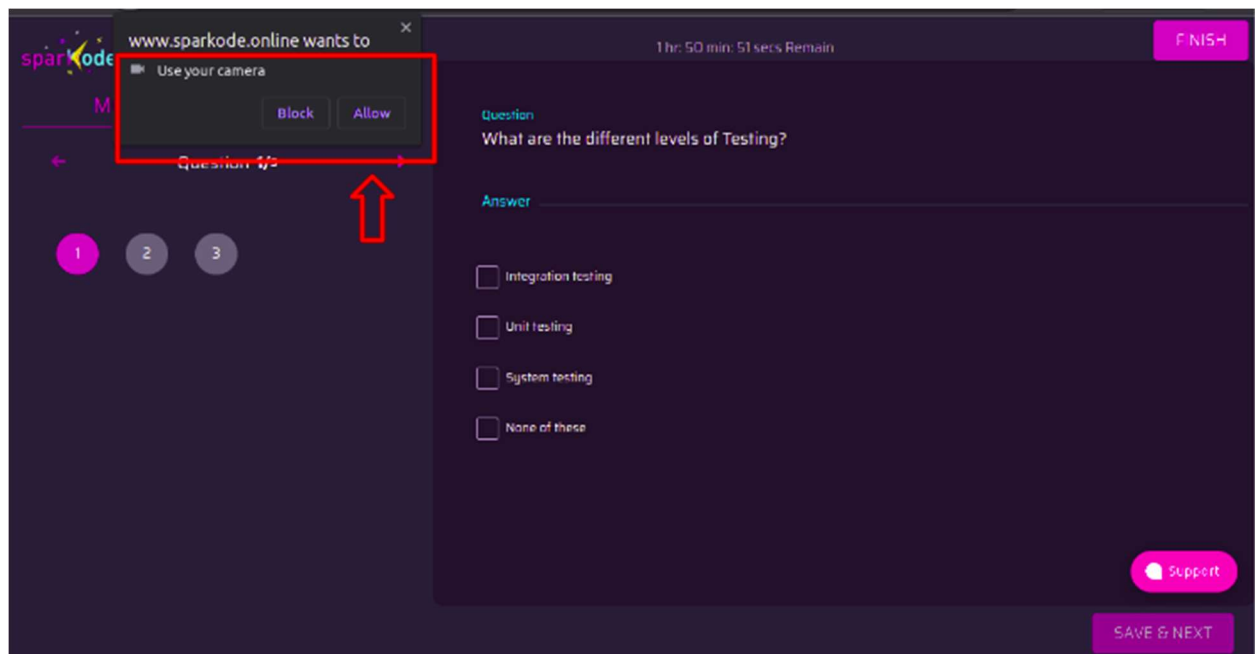
Instructions

- If test Stops due to power failure or system shut down or any other scenario, follow exactly same procedure, with exactly same name and email-id as registration page that was previously used to start the test and in exactly same OS & Browser as mentioned above to restart the test.
- It is mandatory to keep your video camera on during the test.
- Maximum 8 submissions are allowed to each problem.
- Don't click on the finish button unless you want to exit the test.
- In case of the same marks, priority will be given to the candidate who submits first.
- If multiple candidates submit identical codes then all such candidates will be rejected.

Continue

Step 3:

To grant access to the camera—which is necessary while giving the test—click Allow.



The image shows the test interface with a camera permission dialog box. The dialog box asks 'www.sparkcode.online wants to Use your camera' and has 'Block' and 'Allow' buttons. A red box highlights the dialog box, and a red arrow points to the 'Allow' button. The test interface shows a question 'What are the different levels of Testing?' with four options: Integration testing, Unit testing, System testing, and None of these. A 'FINISH' button is in the top right, and a 'Support' button is in the bottom right. A 'SAVE & NEXT' button is at the bottom. A timer shows '1 hr: 50 min: 51 secs Remain'.

www.sparkcode.online wants to Use your camera

Block Allow

Question: What are the different levels of Testing?

Answer:

- ☐ Integration testing
- ☐ Unit testing
- ☐ System testing
- ☐ None of these

FINISH

Support

SAVE & NEXT

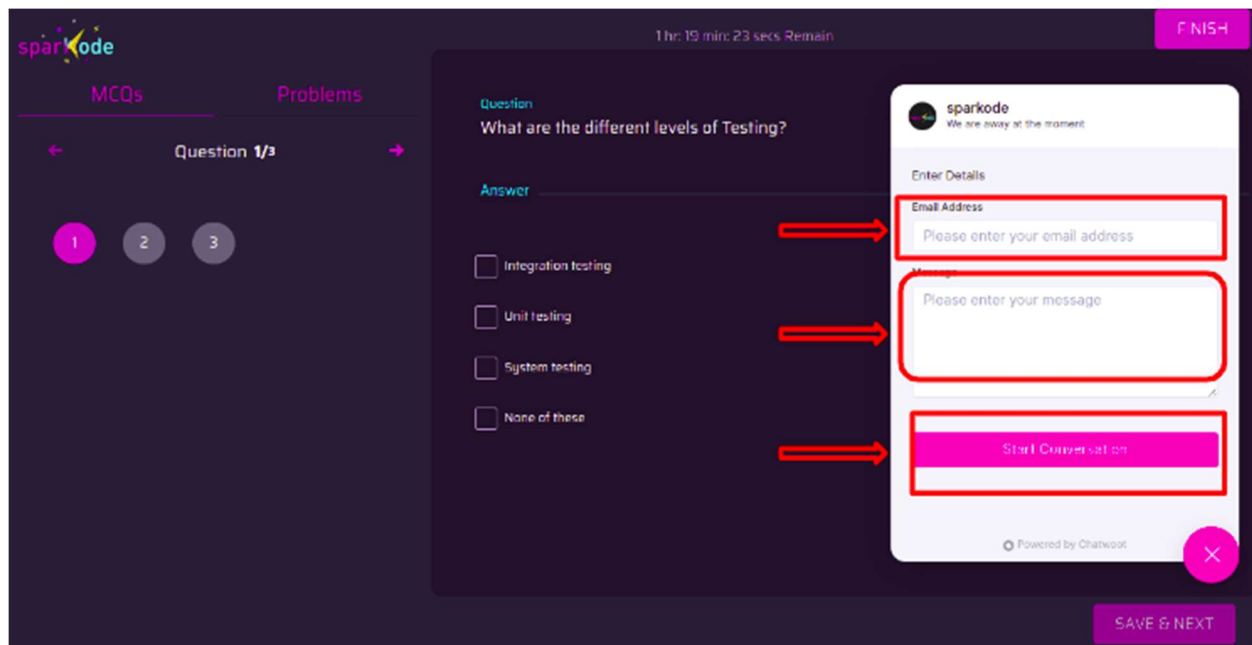
The highlighted region displays the remaining test time.

The screenshot shows the SparkCode coding environment. On the left, the problem description for 'Problem 1/3' is visible, titled '[DO NOT USE]- TEST- TO LOWER CASE'. It asks to return a string with uppercase letters swapped for lowercase. Examples and constraints are provided. The main area is a code editor for Ruby (2.7.0) with a few lines of code. A red box highlights the remaining time '1 hr: 19 min: 35 secs Remain' in the top right. At the bottom right, there is a 'Support' button. The bottom bar shows '3 Submissions Left', 'Add Custom Input', and buttons for 'SAVE', 'RUN', and 'SUBMIT'.

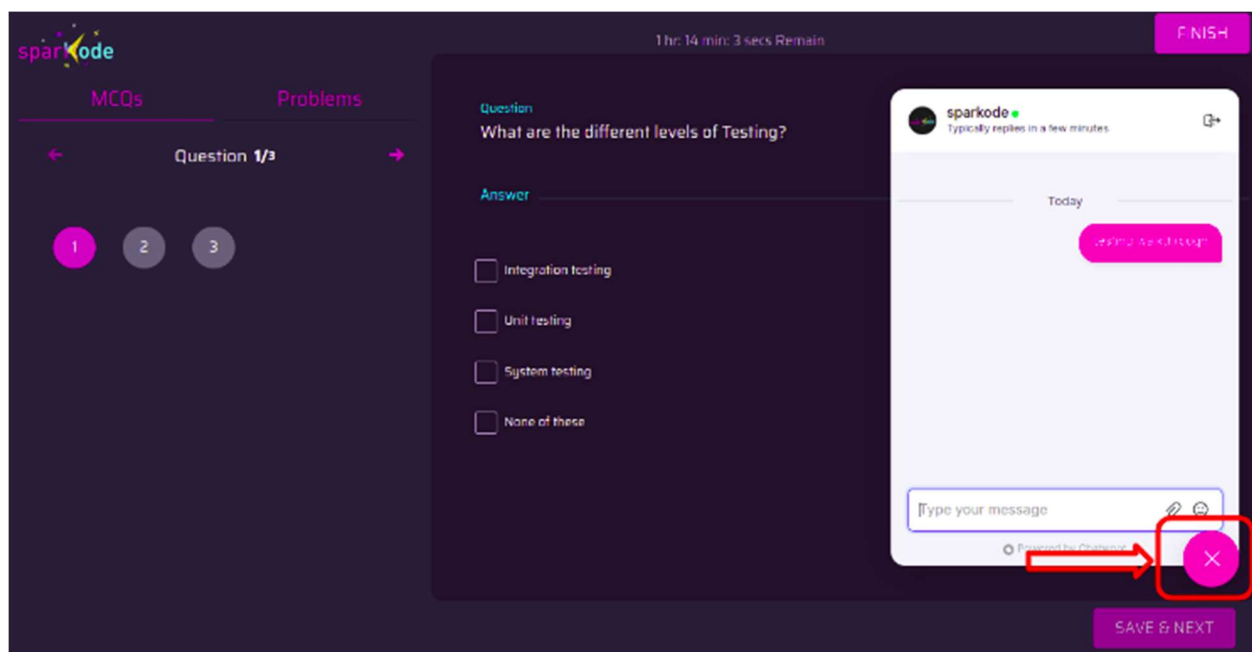
Step 4:

Click on the Start Conversation button Enter the Email, Message and click on the Start Conversation button.

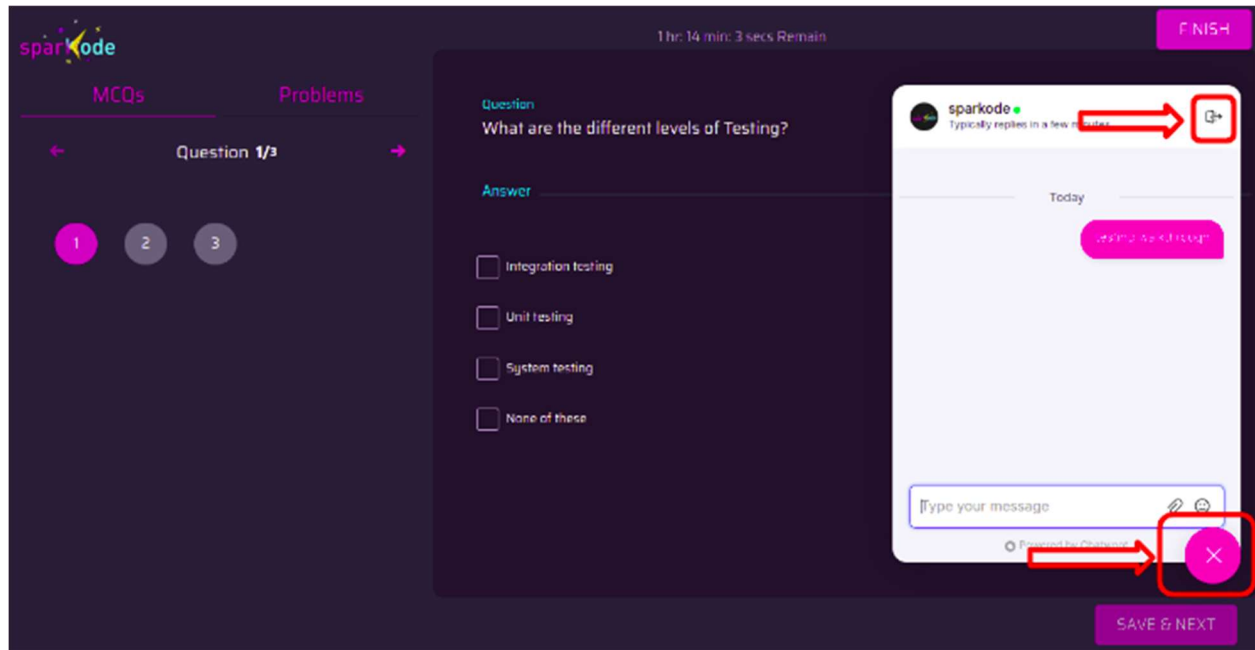
The screenshot shows the SparkCode interface with a multiple-choice question: 'What are the different levels of Testing?'. The options are 'Integration testing', 'Unit testing', 'System testing', and 'None of these'. A red box highlights a chat window on the right side of the screen. The chat window has a 'Welcome' message and a list of instructions. At the bottom of the chat window, there is a 'Start Conversation' button, which is highlighted by a red box and an arrow. The bottom bar shows 'SAVE & NEXT'.



Click on the 'X' icon to close the support chat window.

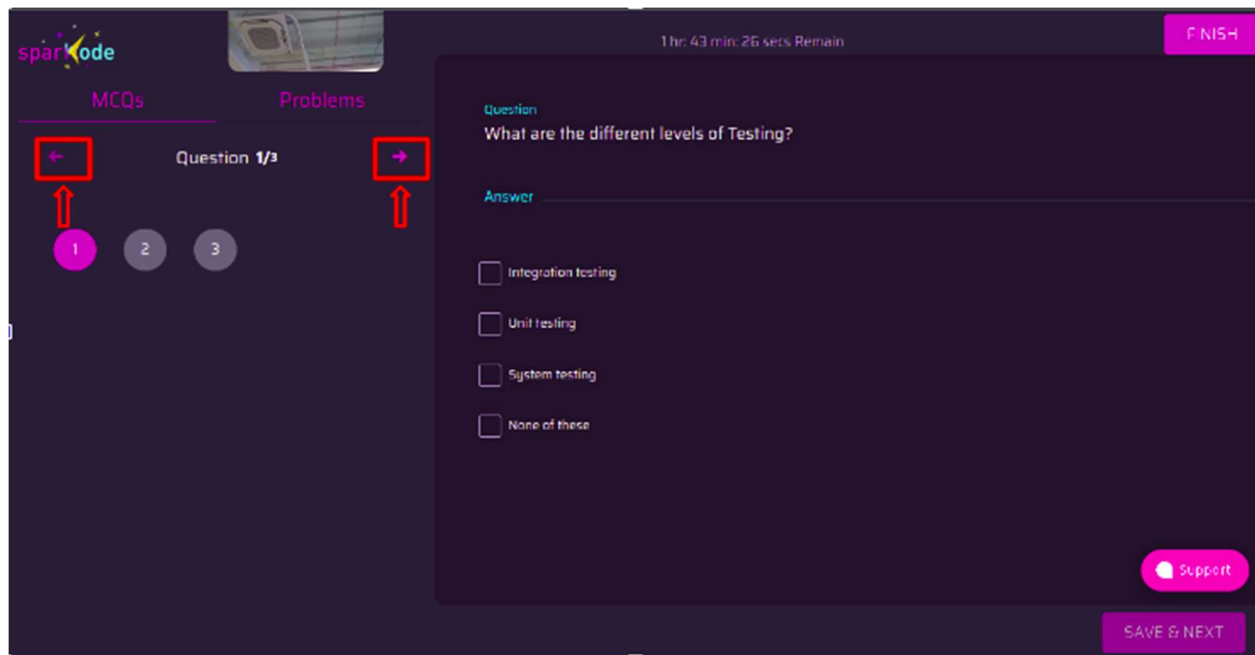


Click on the top right corner of the chat window to end the conversation.



Step 5:

Click on the arrows shown to navigate between the Questions.



In the part where you provide your answers, check the appropriate boxes. To initiate a chat with the support staff in the event of any problems, click the Support button.

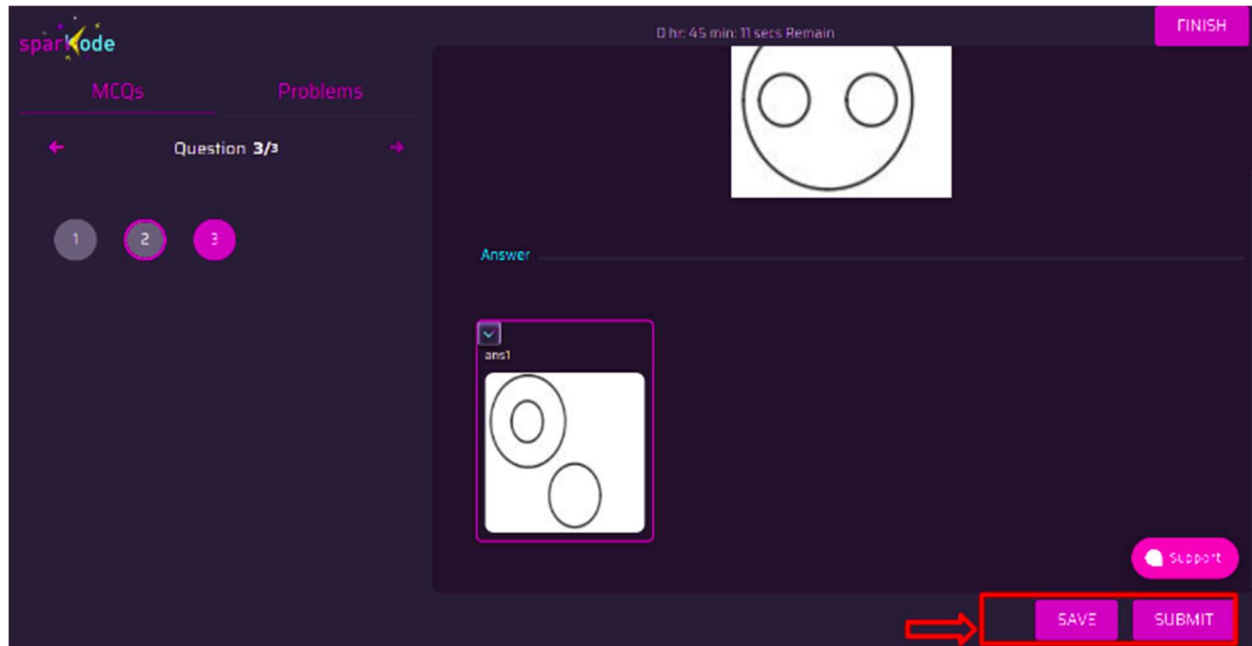
The screenshot shows the SparkCode interface for a quiz. On the left, there's a sidebar with 'MCQs' and 'Problems' tabs. Below them, 'Question 1/3' is displayed with three numbered buttons (1, 2, 3). The main area shows the question: 'What are the different levels of Testing?'. Under 'Answer', there are four checkboxes: 'Integration testing' (checked), 'Unit testing' (checked), 'System testing' (unchecked), and 'None of these' (unchecked). A red box highlights these checkboxes. At the bottom right, there's a 'Support' button with a speech bubble icon, also highlighted with a red box and an arrow pointing to it. Other buttons include 'FINISH' at the top right, 'SAVE & NEXT' at the bottom right, and a timer showing '1 hr: 35 min: 54 secs Remain'.

Step 6:

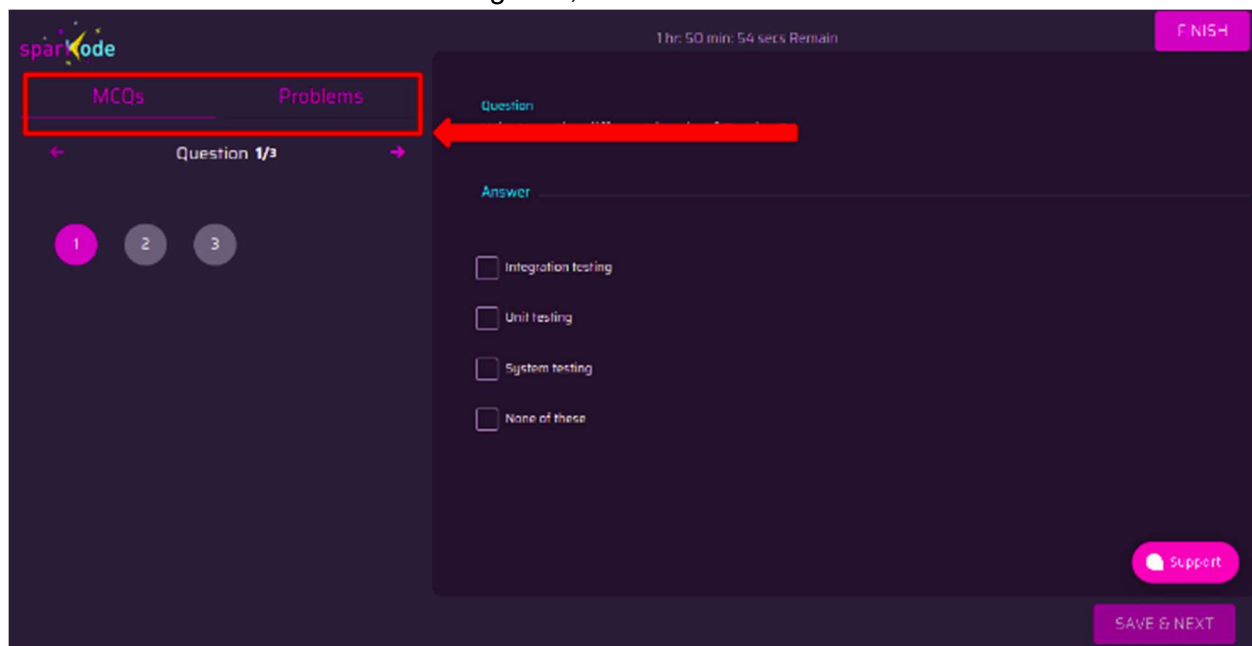
To save the selected response and move on to the following question, click the Save & Next option.

This screenshot is similar to the previous one, showing the same question and answer options. However, the 'Support' button is no longer highlighted. Instead, the 'SAVE & NEXT' button at the bottom right is highlighted with a red box and an arrow pointing to it. The timer now shows '1 hr: 1 min: 36 secs Remain'.

Click on the Save and then Click on the Submit button to submit MCQs.

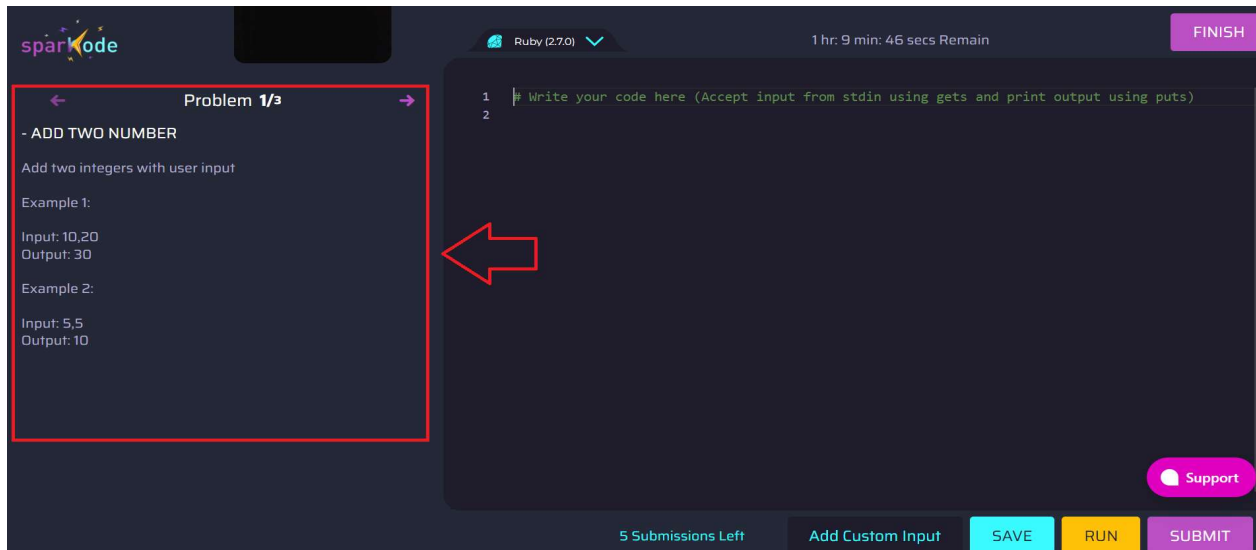


To switch between the Question categories, candidates can click on MCQs and Problems.

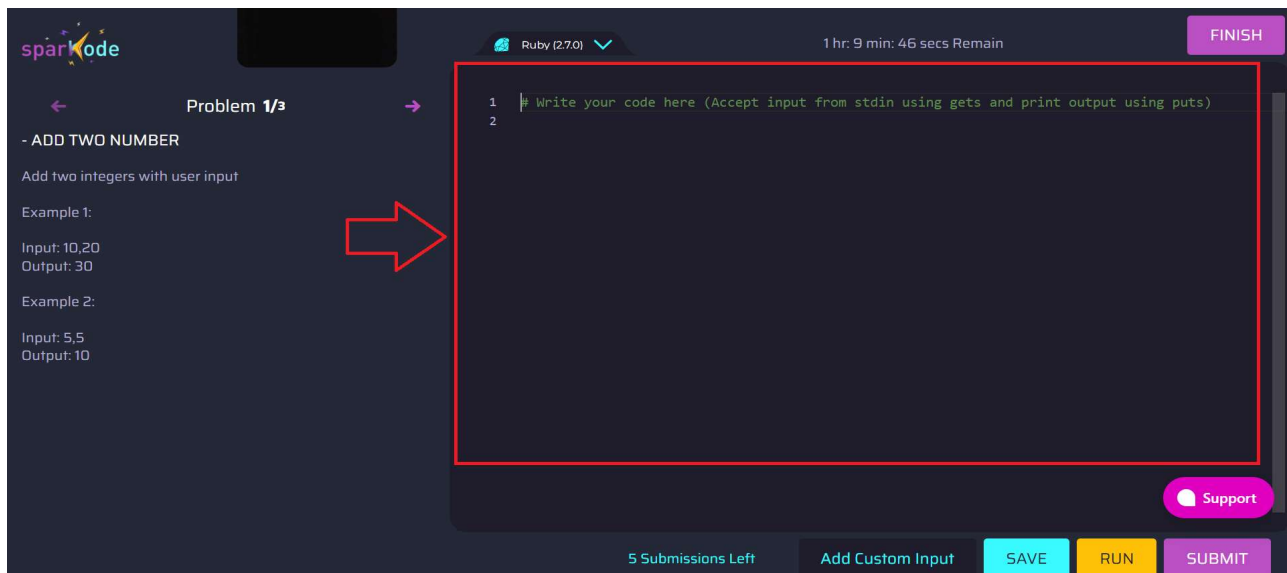


Step 7:

The problem statement is displayed in the portion that is highlighted.

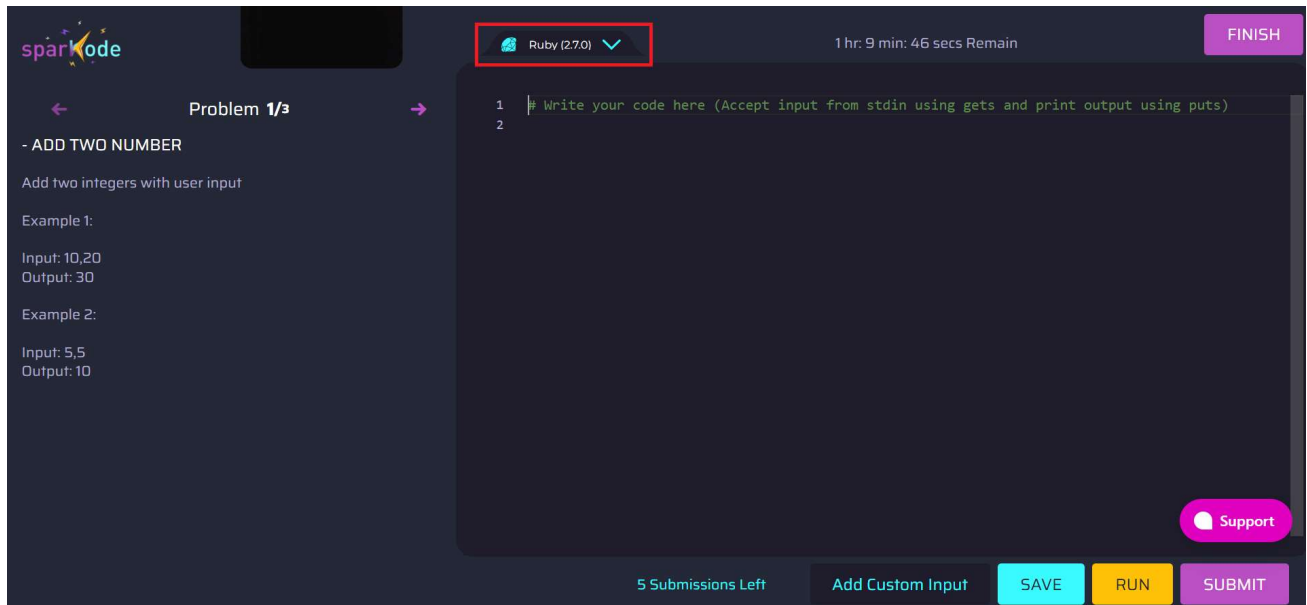


The Editor in which the candidate can write his code is displayed in the highlighted session.

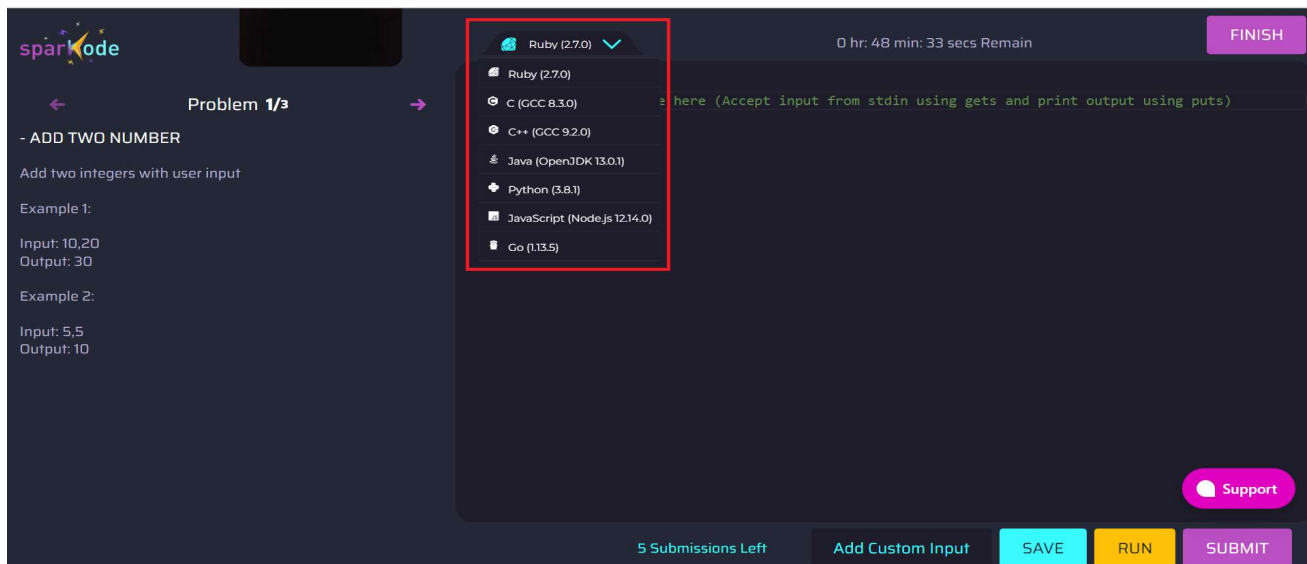


Step 8:

To change the writing language, click on the language template.



From the languages list drop-down, choose the necessary language template.



Click on the RUN button after entering the code to check the Output.

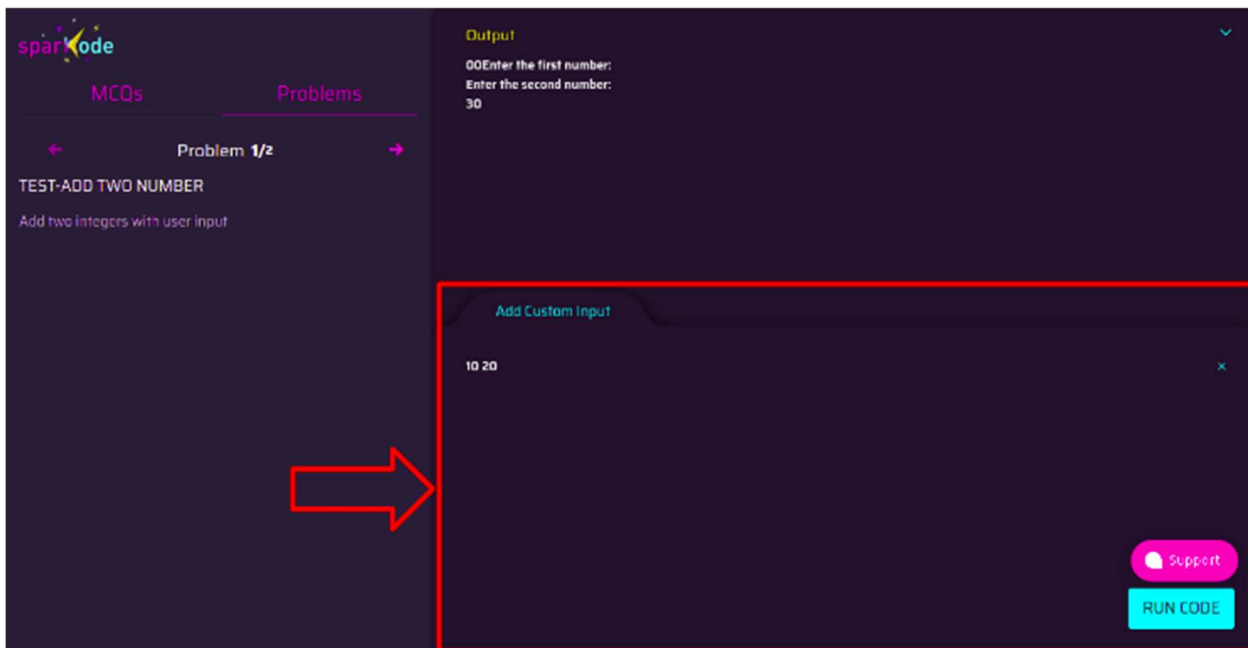
The screenshot shows the Sparkode IDE interface. On the left, the problem title is "Problem 1/3 - ADD TWO NUMBER" with a description "Add two integers with user input" and two examples. The main editor displays C++ code for adding two numbers. The bottom bar includes a "Support" button, a "5 Submissions Left" indicator, and buttons for "Add Custom Input", "SAVE", "RUN", and "SUBMIT". A red arrow points to the "RUN" button.

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     // your code goes here
6     int num1, num2;
7
8     cout<<num1;
9     cout<<num2;
10    cout<<"Enter the first number: \n";
11    cin>>num1;
12    cout<<"Enter the second number: \n";
13    cin>>num2;
14    cout<<{num1 + num2};
15 }
16
```

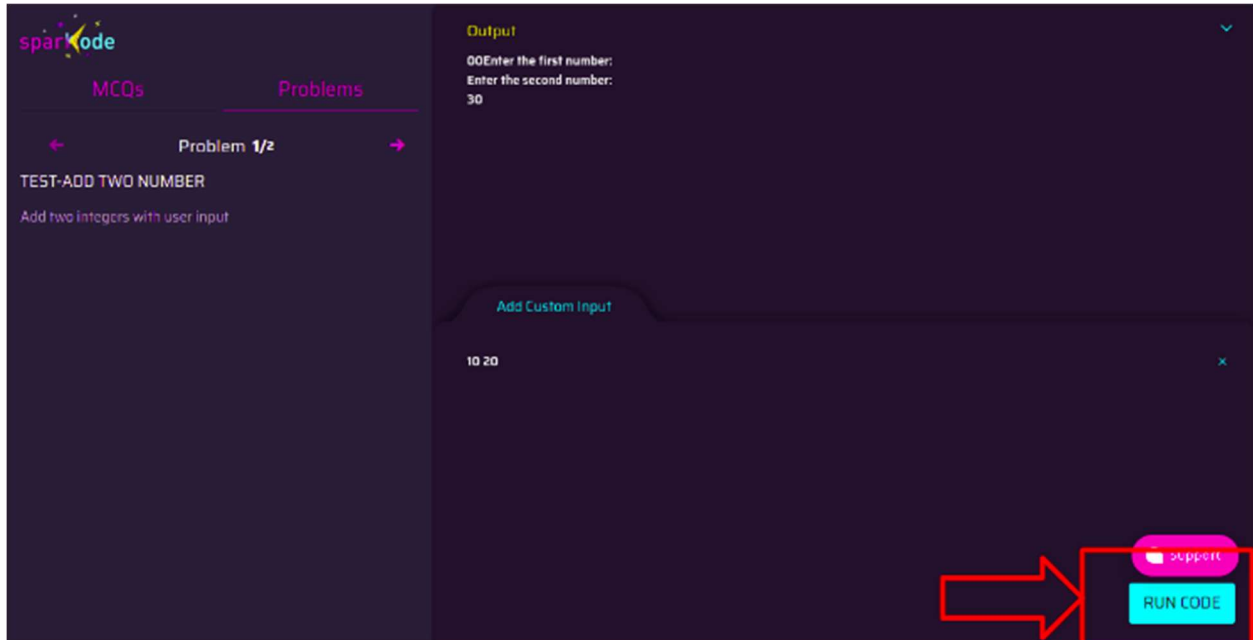
Click on the Add Custom input to enter the custom input.

This screenshot is identical to the one above, showing the same problem and code. However, a red arrow now points to the "Add Custom Input" button in the bottom bar.

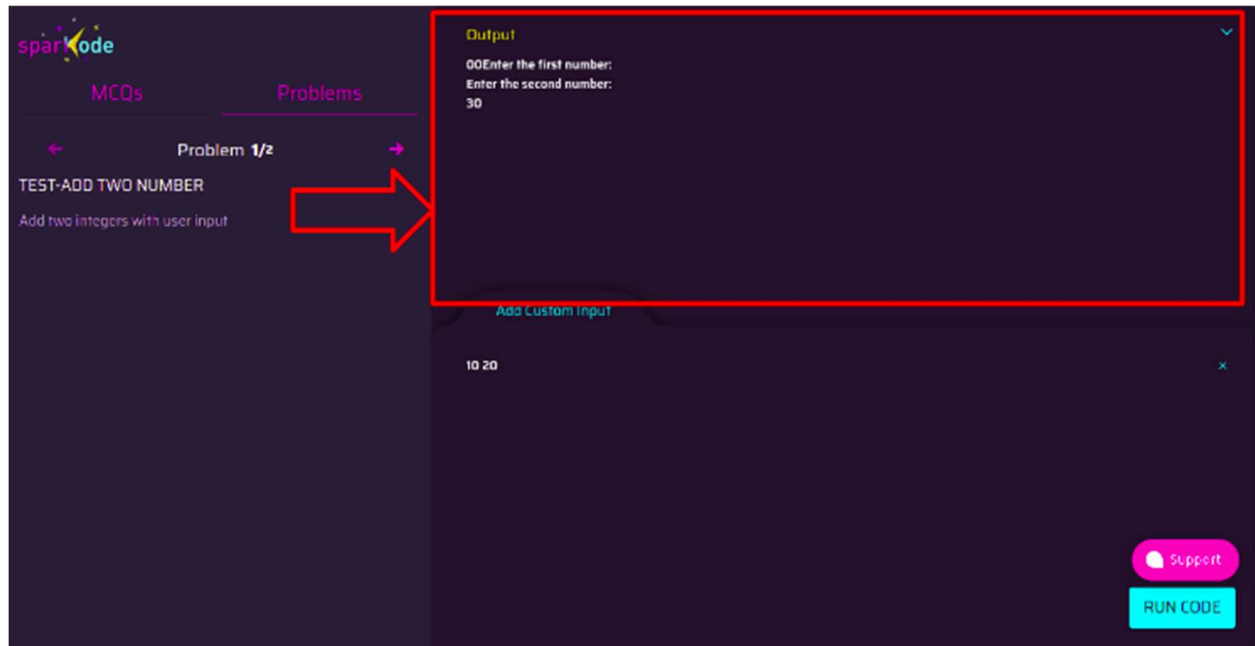
Enter Input in the highlighted field.



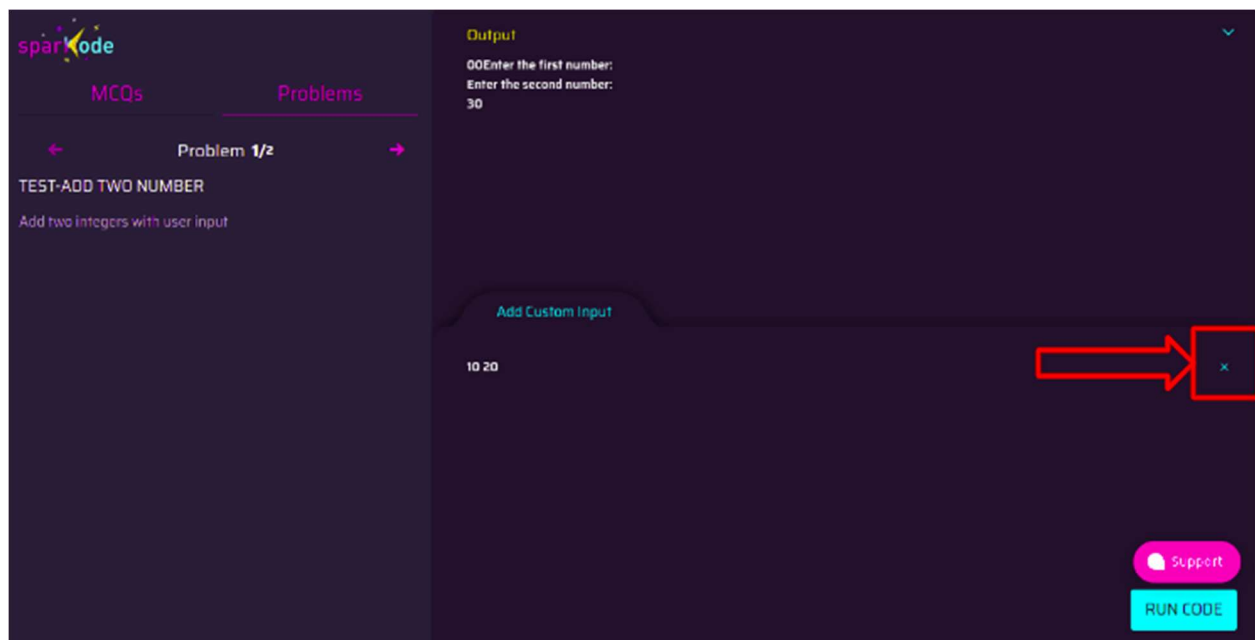
Click on the Run Code to check the Output.



Output can be shown in the highlighted section.

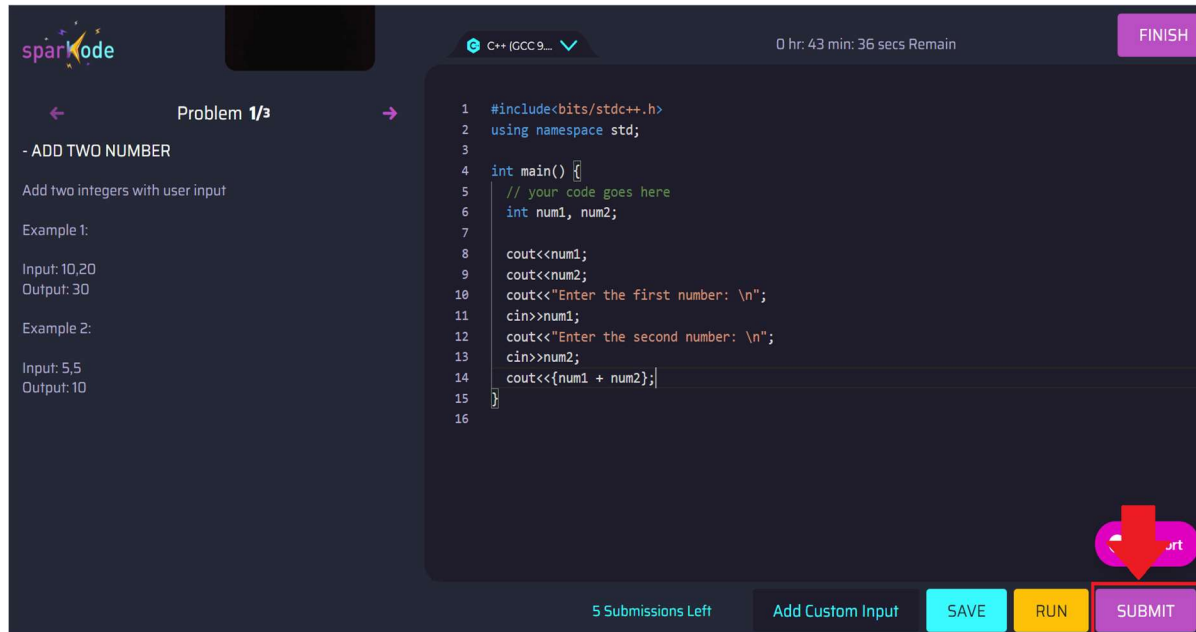


Click on the 'x' icon to close the custom input window.

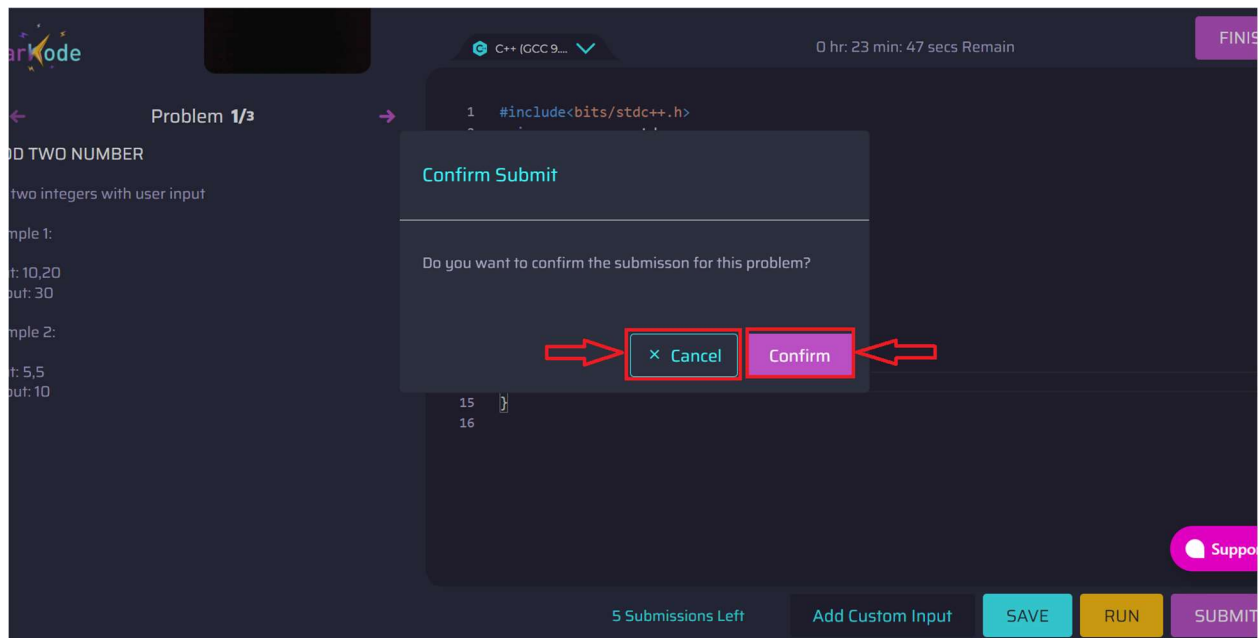


Step 9:

Click on the Submit button to submit the code and run the test cases.

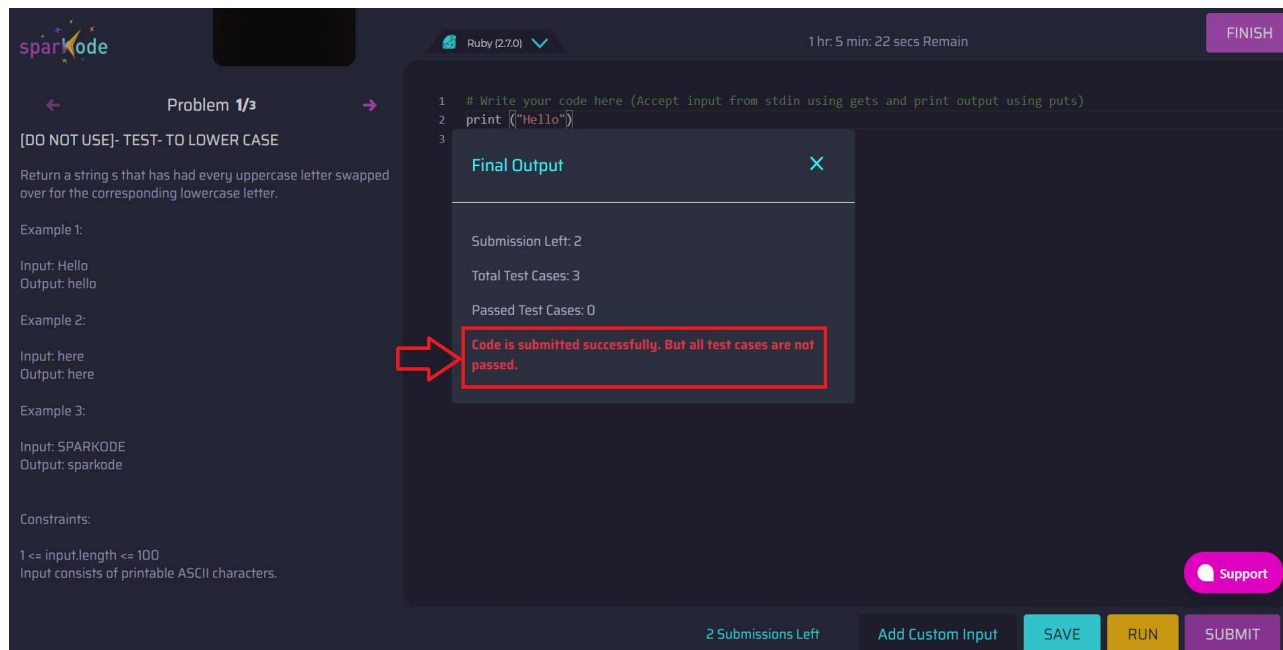


Click on the Confirm button for submission of the code or click on the Cancel button to cancel the submission.

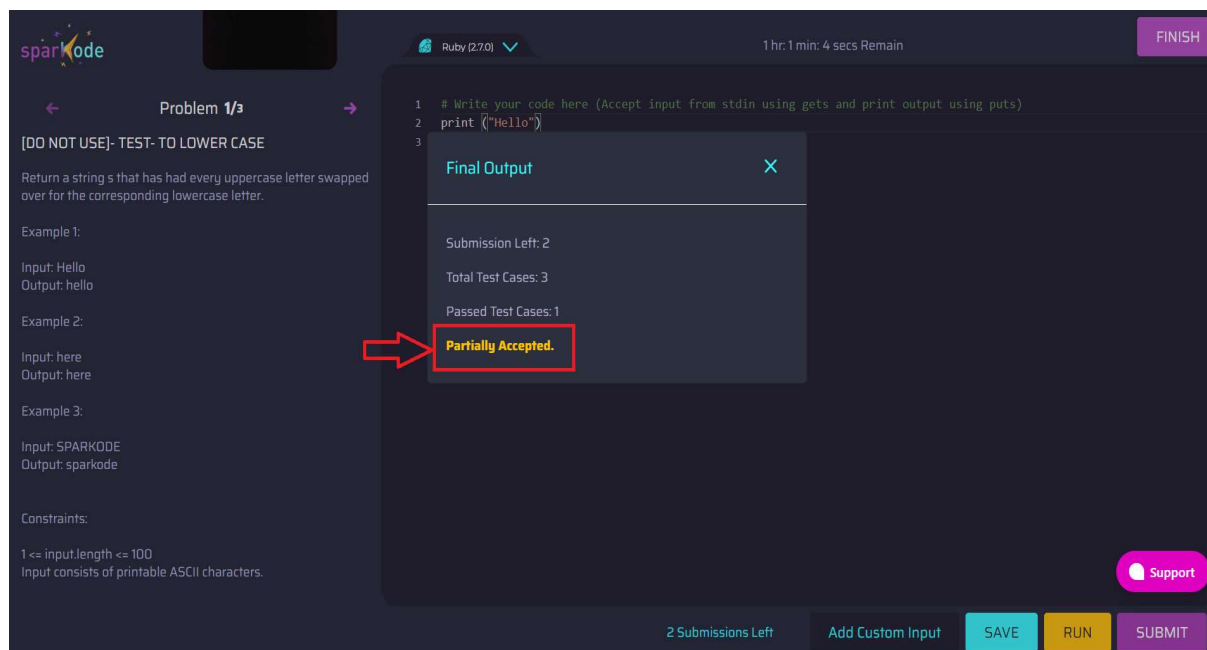


Step 10:

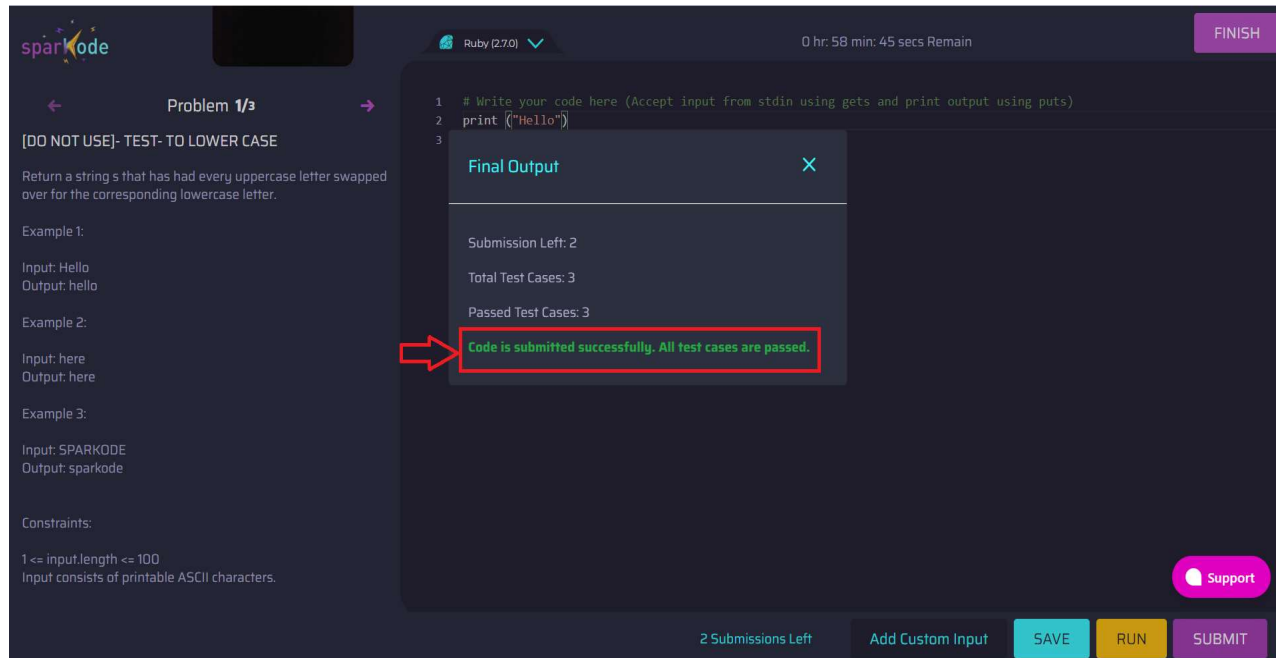
After clicking the confirm button a popup will appear showing the submission count, Total test cases number, Passed Test Cases with count and Status as Accepted or Failed. When all the test cases fail then the status will be shown as below.



When some of the test cases pass then the status will be shown as below.



When all the test cases pass then the status will be shown as below.

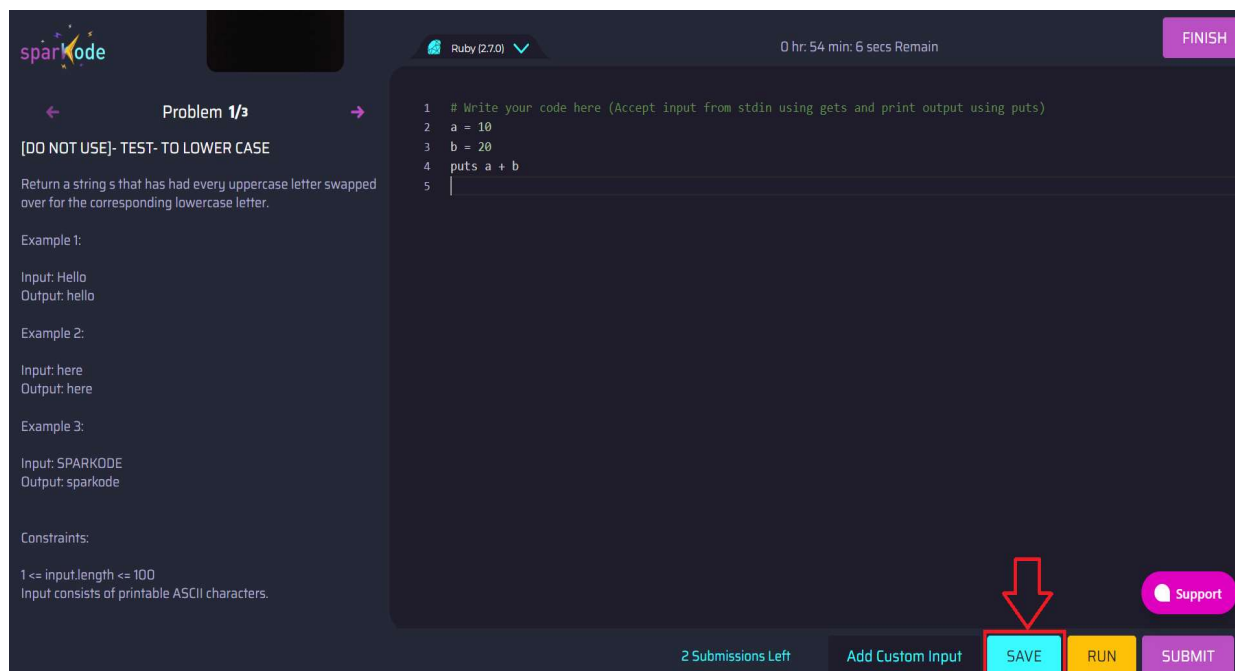


The screenshot shows the Sparkode IDE interface. On the left, the problem description for "Problem 1/3" is visible, titled "[DO NOT USE]- TEST- TO LOWER CASE". The main editor area displays Ruby code:

```
1 # Write your code here (Accept input from stdin using gets and print output using puts)
2 print ("Hello")
3
```

. A "Final Output" modal window is open in the center, displaying the following information: "Submission Left: 2", "Total Test Cases: 3", "Passed Test Cases: 3", and a green message "Code is submitted successfully. All test cases are passed." which is highlighted by a red arrow. The bottom of the interface shows a status bar with "2 Submissions Left", "Add Custom Input", and buttons for "SAVE", "RUN", and "SUBMIT". A "FINISH" button is in the top right corner.

Click on the Save Button to Save the written Code.



The screenshot shows the Sparkode IDE interface with the same problem description on the left. The main editor area contains Ruby code:

```
1 # Write your code here (Accept input from stdin using gets and print output using puts)
2 a = 10
3 b = 20
4 puts a + b
5
```

. The bottom status bar shows "2 Submissions Left", "Add Custom Input", and buttons for "SAVE", "RUN", and "SUBMIT". A red arrow points to the "SAVE" button, indicating it should be clicked to save the code. A "FINISH" button is in the top right corner.

To complete the test, click the Finish button.

sparkode

Ruby (2.7.0) 0 hr: 54 min: 6 secs Remain

Problem 1/3

[DO NOT USE]- TEST- TO LOWER CASE

Return a string s that has had every uppercase letter swapped over for the corresponding lowercase letter.

Example 1:
Input: Hello
Output: hello

Example 2:
Input: here
Output: here

Example 3:
Input: SPARKODE
Output: sparkode

Constraints:
1 <= input.length <= 100
Input consists of printable ASCII characters.

```
1 # Write your code here (Accept input from stdin using gets and print output using puts)
2 a = 10
3 b = 20
4 puts a + b
5
```

2 Submissions Left Add Custom Input SAVE RUN SUBMIT

Support

FINISH

Click on the Finish button to complete the test or click on the Cancel button to cancel the Finish action.

sparkode

Ruby (2.7.0) 0 hr: 57 min: 14 secs Remain

Problem 1/3

[DO NOT USE]- TEST- TO LOWER CASE

Return a string s that has had every uppercase letter swapped over for the corresponding lowercase letter.

Example 1:
Input: Hello
Output: hello

Example 2:
Input: here
Output: here

Example 3:
Input: SPARKODE
Output: sparkode

Constraints:
1 <= input.length <= 100
Input consists of printable ASCII characters.

```
1 # Write your code here (Accept input from stdin using gets and print output using puts)
2 print ("Hello")
3
```

Do you want to Finish the test?

Once you click on Finish you won't have access for the test.
You will logout from test.

Cancel Finish

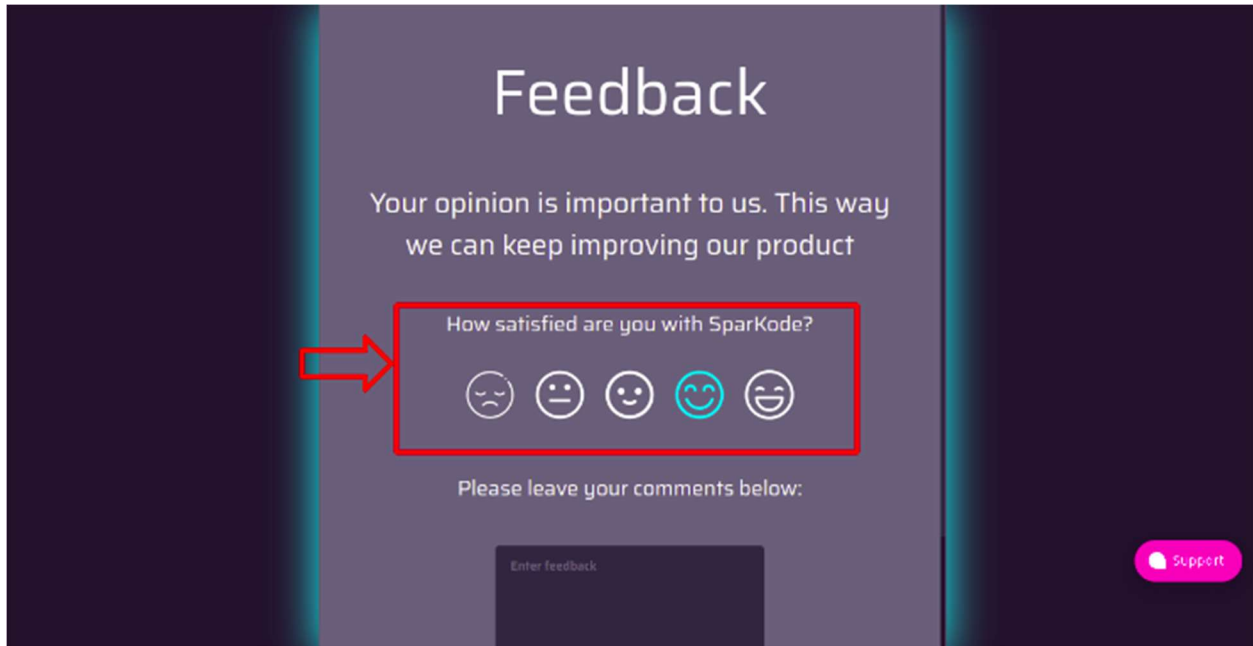
2 Submissions Left Add Custom Input SAVE RUN SUBMIT

Support

FINISH

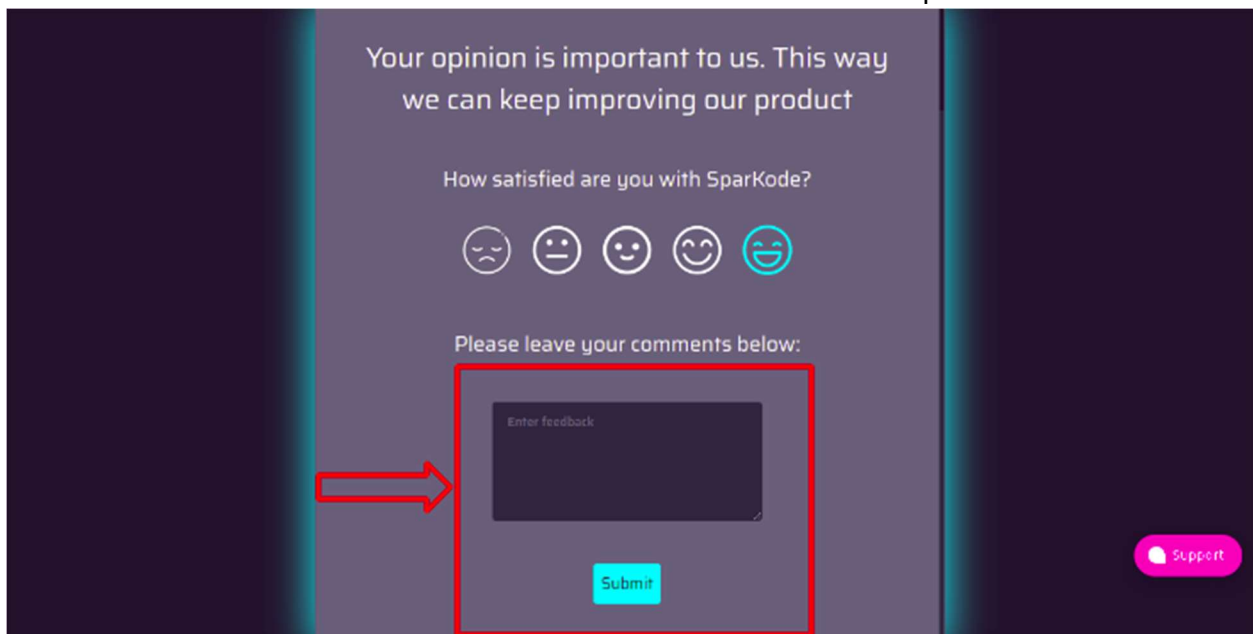
Step 11:

After Clicking on the Finish button candidates will be redirected to the Feedback Page where the Candidate should select the Feedback emoticon rating from the 1-5 score range.



The screenshot shows a feedback form titled "Feedback" with the subtitle "Your opinion is important to us. This way we can keep improving our product". Below this, the question "How satisfied are you with SparkCode?" is followed by five emoticon ratings. The third emoticon from the left (a neutral smile) is highlighted with a red box and a red arrow pointing to it. Below the emoticons, there is a text input field labeled "Enter feedback" and a "Support" button in the bottom right corner.

Enter the Feedback in the text field and Click the Submit button to complete the test.



The screenshot shows the same feedback form as above. In this step, the text input field labeled "Enter feedback" and the "Submit" button below it are highlighted with a red box and a red arrow pointing to them. The "Support" button remains in the bottom right corner.

Once the feedback is submitted then the Test completion page will be displayed.

