

## Description

Simple tasks are good for younger kids, but math can be more difficult and more interesting! Quadratic equations, trigonometry, and a lot of other interesting things. Math library can help you with that.

Sometimes students want to save the results of the test. This is useful for viewing the learning dynamics on a topic or to identify difficult tasks.

At this stage, let's add integral squares. Of course, you can add more difficulty levels later.

## Objectives

1. With the first message, the program should ask for a difficulty level:  
1 - simple operations with numbers 2-9  
2 - integral squares 11-29
2. A user enters an answer.  
For the first difficulty level: the task is a simple math operation; the answer is the result of the operation.  
For the second difficulty level: the task is an integer; the answer is the square of this number.  
In case of another input: ask to re-enter. Repeat until the format is correct.
3. The application gives 5 tasks to a user.
4. The user receives one task, prints the answer.  
If the answer contains a typo, print Incorrect format. and ask to re-enter the answer. Repeat until the answer is in the correct format.  
If the answer is a number, print Right! or Wrong! Go to the next question.
5. After five answers, print Your mark is N/5. where **N** is the number of correct answers.
6. Output Would you like to save your result to the file? Enter yes or no.  
In case of yes, YES, y, Yes: the app should ask the username and write Name: n/5 in level X (<level description>). (X stands for the level number) in the results.txt file. For example — Alex: 3/5 in level 1 (simple operations with numbers 2-9).  
The results should be saved to the file immediately after the user gave the positive

answer to the question Would you like to save your result to the file?  
If the file results.txt does not exist, you should create it.

7. In case of no or any other word: exit the program.

## Example

The greater-than symbol followed by a space (> ) represents the user input. Note that it's not part of the input.

### Example 1:

```
Which level do you want? Enter a number:
1 - simple operations with numbers 2-9
2 - integral squares of 11-29
> 11
Incorrect format.
Which level do you want? Enter a number:
1 - simple operations with numbers 2-9
2 - integral squares of 11-29
> 2
11
> 121
Right!
15
> 100
Wrong!
21
> 441'
Wrong format! Try again.
21
> 441
Right!
17
> 289
Right!
13
> 169
Right!
Your mark is 4/5. Would you like to save the result? Enter yes or no.
> yes
What is your name?
> Kate
The results are saved in "results.txt".
```

## Afterword

After finishing this stage, you are totally free to improve the project in any way you like to make it a more convenient and useful tool.

You can add any features to your application. It will not be verified by tests, so there are no strict requirements.

Sample ideas:

1. Add a complex exam. Increase a difficulty level on the fly. For example, if a person passed the 1st level, start the 2nd one immediately.
2. You can add a correction level: store the tasks with wrong answers and give them next time.
3. Add more difficulty levels.
4. Track the time ([read about Timer](#)).
5. Write a more detailed report to a file with the results.
6. Show previous results inside the app (show lines from `results.txt` that contains the username)
7. Any other improvement that might be useful!