

# Import message on plagiarism

---

The single most important point for you to realize before the beginning of your studies at ShanghaiTech is the meaning of "plagiarism":

Plagiarism is the practice of taking someone else's work or ideas and passing them off as one's own. It is the misrepresentation of the work of another as your own. It is academic theft; a serious infraction of a University honor code, and the latter is your responsibility to uphold. Instances of plagiarism or any other cheating will be reported to the university leadership, and will have serious consequences. Avoiding any form of plagiarism is in your own interest. If you plagiarize and it is unveiled at a later stage only, it will not only reflect badly on the university, but also on your image/career opportunities.

Plagiarism is academic misconduct, and we take it very serious at ShanghaiTech. In the past we have had lots of problems related to plagiarism especially with newly arriving students, so it is important to get this right upfront:

## **You may ...**

- ... discuss with your peers about course material.
- ... discuss generally about the programming language, some features, or abstract lines of code. As long as it is not directly related to any homework, but formulated in a general, abstract way, such discussion is acceptable.
- ... share test cases with each other.
- ... help each other with setting up the development environment, etc.

## **You may not ...**

- ... read, possess, copy or submit the solution code of anyone else (including people outside this course or university)!
- ... receive direct help from someone else (i.e. a direct communication of some lines of code, no matter if it is visual, verbal, or written)!
- ... give direct help to someone else. Helping one of your peers by letting him read your code or communicating even just part of the solution in written or in verbal form will have equal consequences.
- ... gain access to another one's account, no matter if with or without permission.
- ... give your account access to another student. It is your responsibility to keep your account safe, always log out, and choose a safe password. Do not just share access to your computer with other students without prior lock-out and disabling of automatic login functionality. Do not just leave your computer on without a lock even if it is just for the sake of a 5-minute break.
- ... work in teams. You may meet to discuss generally about the material, but any work on the homework is to be done individually and in privacy. Remember, you may not allow anyone to even just read your source code.

With the Internet, "paste", and "share" are easy operations. Don't think that it is easy to hide and that we will not find you. We have just as easy to use, fully automatic and intelligent tools that will identify any potential cases of plagiarism. And do not think that being the original author will make any difference. Sharing an original solution with others is just as unethical as using someone else's work.

# CS100 Homework 1 (Spring 2023)

---

Deadline: 2023-02-26 (Sun) 23:59:59

Late submission will open for 24 hours after the deadline, with 50% point deduction.

If you get full marks in this assignment by no more than 20 submission attempts, you can earn special OJ displays and a "1-case protection" that can be used in further assignments to cancel one testcase failure. See Piazza or OJ dashboard for more information.

## Problem 1: Find Bugs!

---

### Description

Your friend has just started learning programming, and has spent hours on a textbook practice problem. He/She asks you for help by treating you to a KFC Crazy Thursday meal. You agree, and he/she sends you a `.txt` file:

```
#include <sudo.h>

void main ()
{
    // Calculate the average score of all students in a class.

    int num;
    int sum;

    printf('How many students are there?\n')
    scanf('d%', num)
    printf("What are their scores?\n")

    // We programmers count from zero!
    for (int i = 0, i <= num, i++);
    {
        int sum, score;
        scanf("d%", score);
        sum += score;
    }

    double average = sum / num;
    if (average = 60)
        printf('Good!\n');
    if (average > 60)
        printf('Excellent!\n');
    else
        printf("Bad!\n");

    printf("Average score is %.2f.\n", average);

    return 0;
}
```

For the sake of Crazy Thursday, you now have the responsibility to debug your friend's code (Don't do so in ShanghaiTech!). You may first fix compile errors so that the code runs, and then figure out why it (possibly) does not produce correct results.

## Input format

The first line contains a number  $n$ , representing the number of students in a class.

The following  $n$  lines each contain  $s_i$ , score of a student.

- For 100% cases,  $0 < n \leq 50$  and  $0 \leq s_i \leq 100$ .

## Output format

Four sentences, specified as in `printf` statements in the code.

In the last sentence, format the average score to **2 decimal digits**.

## Example

Input:

```
2
70
81
```

Output:

```
How many students are there?
what are their scores?
Excellent!
Average score is 75.50.
```

## Problem 2: Equation of A Line

### Description

Input two points' coordinates  $(x, y)$ , where  $x$  and  $y$  are two integers. You need to calculate the equation of the line going through these two points in the form of  $y = kx + b$ , or  $x = c$  (when the slope  $k$  doesn't exist).

### Input format

Two lines, each containing a coordinate in the form  $(x, y)$ , where  $x$  and  $y$  are integers.

- For 100% cases,  $|x| \leq 1000$ , and  $|y| \leq 1000$ .

### Output format

A line in the form of  $y = kx$  `[+/-]`  $b$ , or  $x = c$ .

- $k$ ,  $b$  and  $c$  should all be of type `double`. When printed, format them to **2 decimal digits**.
- If  $b < 0$ , the minus sign `-` should take place of the `+`, turning the result into  $y = kx - b$ .

- You do not need to simplify the cases  $b = 0$ ,  $k = 0$ , and  $k = \pm 1$ . Just leave the number there, like `y = 1.00x + 0.00`.

Pay attention to spaces before and after '=' and '+ '/' -'. It's safest to directly copy-paste the above equations into your code.

## Example

Input:

```
(2, -1)
(4, -3)
```

Output:

```
y = -1.00x + 1.00
```

Input:

```
(-2, 3)
(4, -6)
```

Output:

```
y = -1.50x + 0.00
```

Input:

```
(-10, -3)
(2, -3)
```

Output:

```
y = 0.00x - 3.00
```

Input:

```
(-2, 0)
(-2, 1)
```

Output:

```
x = -2.00
```

## Problem 3: Spell It Out

---

## Description

"How I wish that numbers could be gone!", cursed you after a terrible math exam. That exact night, you had a bad dream where all numbers have disappeared! No one understands 1, 2, 3, and you had to spell them out in English like one, two, three!

Luckily you had a computer to do all this, and a piece of guide to follow:

- Use a hyphen ( '-' ) to connect a word ending in -ty to another word. (22 -> twenty-two, 48000 -> forty-eight thousand)
- There is **NO** "-s" (plural) for words "hundred" and "thousand". Also, 100 and 1000 are spelled "one hundred" and "one thousand".
- To use or to not use an "and" after "hundred" is both OK. However, inserting an "and" makes the spelling easier to read, so let's put an "and" after "hundred" or "thousand" when they are followed by a part less than 100. (101 -> one hundred **and** one, 2023 -> two thousand **and** twenty-three, 114500 -> one hundred **and** fourteen thousand five hundred)
- **NO** comma ( ',' ) after the word "thousand", although it sounds more natural to stop there when reading numbers out.
- Pay attention to spellings, especially for "fourteen", "forty", "nineteen", and "ninety"! You don't want to spend hours debugging to find out nothing but a typo!

## Input description

The input is a number  $x$ .

- For 20% cases,  $0 \leq x \leq 999$ .
- For 20% cases,  $x \% 1000 == 0$ .
- For 100% cases,  $0 \leq x \leq 999999$ .

## Output description

You should output the English spelling of number  $x$  in one line. It's OK whether you end the line with a '\n' or not.

## Hints

- This problem does NOT involve strings (something we haven't learned)! Try to separate the output into different parts, and print each one with a `printf`! Keep in mind that `printf` does not necessarily end with a '\n'.
- Some work might need to be performed multiple times (like translating 1-9 into "one" through "nine"). Why not write them into functions to make things easy and avoid bugs?

## Sample input/outputs

Input	Output
0	zero
44	forty-four
1590	one thousand five hundred and ninety
114500	one hundred and fourteen thousand five hundred

# Appendix: Numbers in English

0	1	2	3	4	5	6	7	8	9
zero	one	two	three	four	five	six	seven	eight	nine

10	11	12	13	14	15	16	17	18	19
ten	eleven	twelve	thirteen	fourteen	fifteen	sixteen	seventeen	eighteen	nineteen

20	30	40	50	60	70	80	90	100	1000
twenty	thirty	forty	fifty	sixty	seventy	eighty	ninety	hundred	thousand