考試規定

- 使用 Dev-C++ 考試，我們會以 Dev-C++ 作為批改標準。
- 請將每題撰寫不同檔案中，並以「學號_題號.c」的方式命名。
- 題目共有五題，單一題目的分數僅有全拿與零分兩種。
- 交卷時將五題一起上傳到網大的作業繳交區，無須壓縮。

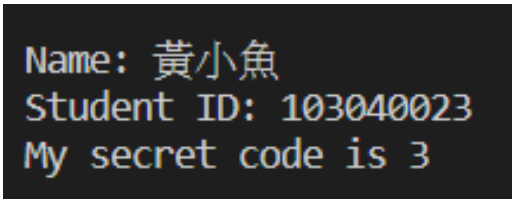
1. Self Introduction (20%)

Design a program to print your name and student ID (Exclude the first letter).

Then print your secret code.

secret code : Sum up all digits in your student ID, then take the remainder of 10.

The output should look like below:



2. Single Number (20%)

Input an array `nums`, every element appears *twice* except for one. The maximum size of array is 100. Let user continuously input until user input -1.

Find that single one.

Example 1:

```
Input: 2 2 1 -1

Output: 1
```

Example 2:

```
Input: 4 1 2 1 2 -1

Output: 4
```

Example 3:

```
Input: 1 -1

Output: 1
```

3. Palindrome Number (20%)

Input an integer, output `true` if it is palindrome integer.

An integer is a **palindrome** when it reads the same backward as forward.

For example, `121` is a palindrome while `123` is not.

Example 1:

```
Input: 121

Output: true

Explanation: 121 reads as 121 from left to right and from right to left.
```

Example 2:

```
Input: -121

Output: false

Explanation: From left to right, it reads -121. From right to left, it
becomes 121-. Therefore it is not a palindrome.
```

Example 3:

```
Input: 10

Output: false

Explanation: Reads 01 from right to left. Therefore it is not a palindrome.
```

4. Majority Element (20%)

Input an array `nums`. The maximum size of array is 100. Let user continuously input until user input -1.

Output the majority element.

The majority element is the element that appears more than `⌊n / 2⌋` times. You may assume that the majority element always exists in the array.

Example 1:

```
Input: 3 2 3 -1

Output: 3
```

Example 2:

```
Input: 2 2 1 1 1 2 2 -1

Output: 2
```

5. Move Zeroes (20%)

Input an array nums. The maximum size of array is 100. Let user continuously input until user input -1. Move all `0`'s to the end of it while maintaining the relative order of the non-zero elements.

Note that you must do this in-place without making a copy of the array.

Example 1:

```
Input: 0 1 0 3 12 -1

Output: [1,3,12,0,0]
```

Example 2:

```
Input: 0 -1

Output: [0]
```