

## 2015 Summer Entrance Examination

Department of Creative Informatics  
Graduate School of Information Science and Technology  
The University of Tokyo

# Programming

### INSTRUCTIONS

1. Do not open this problem brochure until the signal to begin is given.
2. Write your examinee ID number below on this cover page.
3. An answer sheet and a draft sheet accompany this brochure. Write down your examinee ID number on these sheets.
4. You should have received a USB flash drive. It contains a text file `program.txt`.  
*Before the examination starts, copy the file to your PC and browse to it. Verify that you can see the text file and then remove your hands from your PC. If you cannot read the file properly, consult the test supervisor.*
5. You may choose any programming language to answer.
6. You may consult only one printed manual of a programming language during the examination.  
*You can use or copy any libraries or program fragments stored in your PC, but you may not connect to the Internet.*
7. By the end of the examination, make a directory/folder on your PC, whose name is the same as your examinee ID number, and put your program files and related files into the directory/folder. Copy the directory/folder onto the USB flash drive that you received.
8. At the end of the examination, the USB flash drive, the answer sheet and the draft sheet will be collected.
9. After these are collected, stay at your seat, until all the examinee program results have been checked briefly by the test supervisor.
10. After the brief check, try to save your program execution environment on the PC so that you can run your program as soon as possible during the oral examination in the afternoon.
11. *Leave your PC and this brochure together in the room for the oral examination and stay out of the room until you are called.*

Examinee ID \_\_\_\_\_

This page is blank.

This page is blank.

# Programming

A large program often contains duplicate code or *code clone*. We will write a simple program for detecting duplicate code in a given file.

(1) Write a program that reads `program.txt` in the given USB flash drive and counts the number of semi-colons ; in that file. The file `program.txt` is a text file.

(2) Write a program that reads `program.txt` and prints all the lines containing `main` with their line numbers.

(3) Write a program that reads `program.txt` and prints duplicate lines successively repeated in the file. The duplicate lines must be printed only once. For example, if the file is:

```
a = 1
a = a + 1
a = a + 1
b = a
a = 1
```

then, the program prints:

```
a = a + 1
```

It does not print `a = 1` because the two lines are not adjacent.

(4) Write a program that reads `program.txt` and prints duplicate lines in the file. It prints a duplicate line even if the second copy of the duplicate lines is not adjacent to the first one. In the case of the example above, the program prints both:

```
a = 1
a = a + 1
```

Every duplicate line must be printed only once in the order of the first occurrence. Furthermore, the program must print the number of the lines printed as duplicates. For the example above, the program prints 2. Note that a duplicate line may occur more than twice in `program.txt`.

(5) Write a program that reads `program.txt` and prints all pairs of similar two lines in the file. The program must also print the number of the found pairs. The two lines are not necessarily adjacent. Ignore lines shorter than 20 characters.

Two lines are similar if they are not identical and the number of positions at which the corresponding characters are different is less than 5. Here, the corresponding characters at position  $i$  are the  $i$ -th characters in the two lines. If the lengths of the two lines are different, append white spaces to the shorter line so that their lengths are equal. For example (for simplicity, we illustrate with lines shorter than 20 characters),

```
a = 1
a = a + 1
```

The number of positions at which the corresponding characters are different is 3. The characters at the 5th, 7th, and 9th positions are different. The first four characters and two spaces around + are identical.

(6) Modify the program written for (5) to change the definition of similarity. In the new definition, two lines are similar if the minimum number of steps for changing one line into the other is less than 4. Two identical lines are not recognized as similar lines. Each step is either substitution of a character for another, deleting one character, or inserting any single character.

(7) Write a program that reads `program.txt` and prints groups of successive lines if they are more than three lines and they are duplicated in the file. Lines are recognized as duplicate ones only if they are identical. Each group of successive duplicate lines is printed only once.

This page is blank.

This page is blank.

