

2013 Winter 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.
3. An answer sheet and a draft sheet accompany this brochure. Write down your examinee ID number on these sheets.
4. You may choose any programming language to answer.
5. 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.
6. 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 delivered USB flash drive.
7. At the end of the examination, the USB flash drive, the answer sheet and the draft sheet are collected.
8. After these are collected, stay at your seat, until all examinee program results have been checked briefly by the test supervisor.
9. 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.
10. *Leave your PC and this brochure together in the room for the oral examination and leave the room until you are called.*

Examinee ID _____

This page is blank.

This page is blank.

Let the lower-case letters a to z be boolean variables. The operator & represents logical conjunction and + represents logical disjunction. & has higher precedence than +. For a given boolean formula constructed with these operators and variables, an assignment of values to each variable is a *solution* if that formula evaluates to true under that assignment. A boolean formula is a single line of text and the text does not include spaces or tabs but only variable names and operators.

For example, if a given formula is:

```
b&a+b&c+a&b&c
```

then an assignment a=true, b=true, c=false is a solution.

(1) Write a program that reads a boolean formula and splits it by using + as a separator. For example, if the formula above is given, then the program prints:

```
b&a
b&c
a&b&c
```

(2) Write a program that reads a boolean formula and finds all solutions to the boolean formula. The program must print the found solutions or "none" if there exists no solution.

(3) Extend the program for (2) so that it will support the negation operator !. This operator is a prefix unary operator and it has the highest precedence. For example, the program must accept the following formula:

```
!a&b&!c+a&!d
```

(4) Extend the program for (3) so that it can accept a formula including parentheses (and).

(5) Write a program that reads a boolean formula in the style specified in (4) and prints the *disjunctive* normal form (DNF) equivalent to that formula. The DNF is a disjunction of clauses consisting of variables (appearing once for each), &, and/or !. The following formula is a DNF:

```
a&b&c+a&b&!c+!a&b&c
```

(6) Write a program that reads a boolean formula in the style specified in (4) and prints the *conjunctive* normal form (CNF) equivalent to that formula. The CNF is a conjunction of clauses consisting of variables (appearing once for each), +, and/or !. The following formula is a CNF:

```
(!a+b+!c)&(!a+b+c)&(a+!b+c)&(a+b+!c)&(a+b+c)
```

This page is blank.

This page is blank.