CS 313000 Introduction to Computer-Aided Design of Integrated Circuits
Programming Assignment 1 (Deadline: 2018/10/24 23:59)

You are requested to implement a two-level logic optimization program. In this program, you have to reduce the literal count of given Boolean equations. Assume these equations are all written in sum of product (SOP) form. You can solve this problem using Quine-McClusky approach you have learned in class. Also, any heuristic methods in your algorithm are acceptable. Although literal count is the major consideration of this program, the run time of your program is also another important factor.

Input format:

The first two lines list the total number of variables and the number of product terms, respectively. The next n (n = total number of product terms) lines list the literals of this product term. Each of these n lines contains several characters, which represent the logic value (0, 1, or -) of the variables in the product term. The hyphen (-) means the corresponding variable is don't care. For example, a line 10--11 represents that ab'ef is in the on-set of the input Boolean space. The number of the variables shown at the first line will not exceed 26. The letters used will appear in the same order of English letter. For example, if the number of the variables is 6, the letter representing these variables would be abcdef.

Output format:

The first line lists the literal count. The second line lists the number of product terms remained in the optimized equation. The next m (m = total number of remained product terms) lines list all the remained product terms after optimization.

The following samples shows an input file and an output file with comments added.

Sample input:

```
6        // total number of variables = 6
7        // total number of product terms = 7
101110   // product terms
-10101
10-01-
--0111
10-111
100-1-
1-0111
```

Sample output:

```
11          // literal count = 11
3           // total number of product terms = 3
10--1-      // optimized product terms
-101-1
--0111
```

Grading policies:

You have to write this program in C or C++ and a MAKEFILE for compiling the program. TAs will compile your source code and run all the test cases (7 released test cases and 3 hidden cases) on **ic56**. In addition, a REPORT is also needed, in which you should introduce the algorithm you used and how to execute your program. The page limit of the report is 10. The output file of your program should be named as name.out where name is the input file name, e.g., if the input file name is case1.txt, the case1.txt.out is as the output file name. The output files will be verified by the given verify program on **ic18**.

Please package all the following items in one file named HW1_{STUDENT_ID}.zip and submit it to iLMS.

1. Source codes in C/C++.
2. A makefile.
3. A report. (in a PDF file)

Your program should be scalable for large cases. This assignment will be ranked and scored according to the quality of the experimental results and the runtime. Specifically, the percentage is as follows:

Accuracy	60%
Runtime	20%
Report	20%

For each case, the Accuracy and Runtime will be ranked and graded as follows:

Result	Accuracy	Runtime
Ranking top 35%	6	2
Ranking between 35%-70%	5	1.5
Ranking lower 70%	3	1
Functionally not equivalent	1	1
Same results as input	0	0
Segmentation fault or TLE	0	0

Note that if the runtime of your program for a case is longer than 2 hours, it will be treated as Failed for TLE (time limit exceeded).