Programming Assignment #1

Binary Decision Diagram (BDD)

Lab 1 Introduction

1. To exercise the concept of binary decision diagram.
2. To understand the ordering effects of BDD.
3. Problem Description

Please construct BDDs with given variable orderings, and find the minimum number of nodes required from the given variable orderings.

Input

```
Boolean equation.
Variable ordering 1.
Variable ordering 2.
…
Variable ordering n.
```

The first line specifies the Boolean equation, while the following lines give the various variable orderings. Each equation ends up with a period and every variable is represented by exactly one character (i.e., 26 variables at most). The Boolean equation is given in sum-of-product (SOP) form: lowercase character represents a plain variable, whereas its uppercase counterpart is for its complement (~e represents as E).

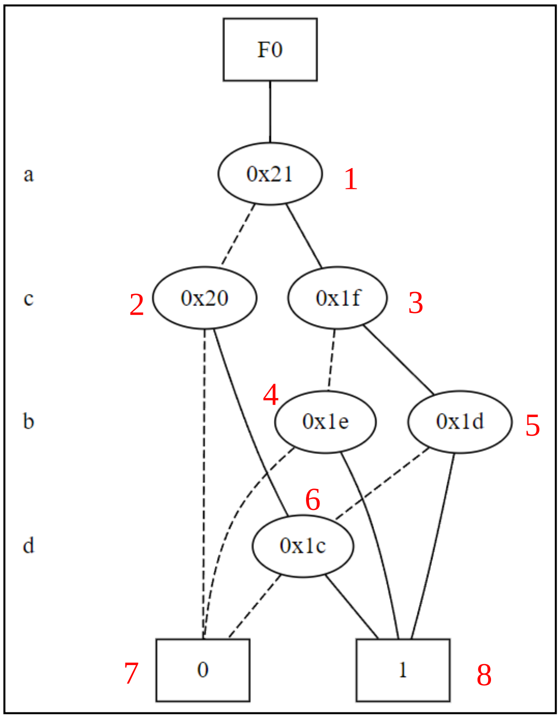
Input example

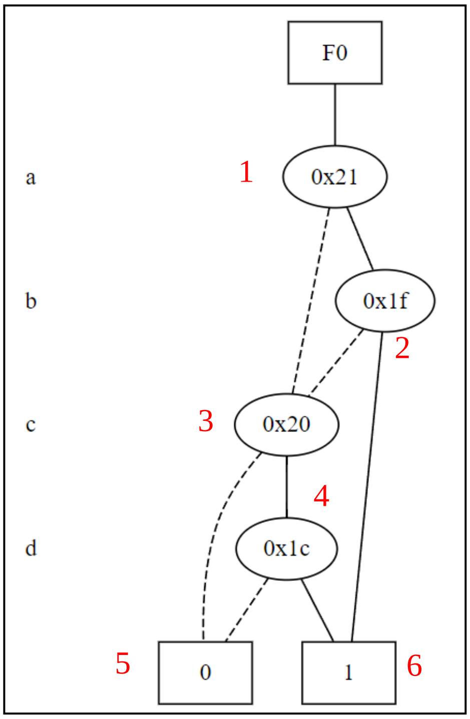
```
ab+cd.
acbd.       // First is variable ‘a’, then ‘c’ …
abcd.       // First is variable ‘a’, then ‘b’ …
```

Output

Output the minimum number of nodes required to represent the given BDD from the given variable orderings.

```
6           // Minimum number of nodes required is 6, as the following figure
```





Compile & Execute

Compile command :	**$ make**
Execute command :	**$ ./Lab1 [input file] [output file]**
e.g.	**$ ./Lab1 case2.txt out2.txt**

Note that input and output file should be the arguments of program. Your executable binary file after "make" should be named as "Lab1" (Hint: add "-o Lab1" in your compilation command). Please make sure your code can be compiled and executed. If it cannot be executed, you will get zero point!

Program Submission

1. Please use the C/C++ language, and write your own code.
2. The materials of CUDD package is provided, which is optionally used in your program.
3. Please upload the following materials in a “zip” file to New E3 by the deadline. Name the zip file as: Student_ID.zip. (e.g. 0610128.zip)
    1. Source code
        - (.c, .cpp, .h).
    2. Makefile.
    3. A readme file
        - (Describe your compile and execution information and whether to use the CUDD package).
    4. cudd-3.0.0/
        - (If you use it, please upload.
        - Make sure your cudd path is working by "make")
5. Don’t print any words on the terminal.

Grading

■ **Case1**	**20%**
■ **Case2**	**20%**
■ **Case3**	**20%**
■ **Case4 (hidden)**	**20%**
■ **Case5 (hidden)**	**10%**
■ **Case6 (hidden)**	**10%**
■ ***Unused the CUDD package***	***20%bonus***

* Time limit is 300s. Otherwise, the case is regarded as failed.

Notices

- Please make sure your code is available on our Linux server. If it cannot be executed, you will get zero point.
- Accept four days late submission, 10% deduction per day.
- Plagiarism is strictly forbidden. 0 grade guarantee