

Logic System Assignment 2

A simple CAD tool based on Quine-McCluskey method

Due date: 2020/06/10

1. Description

In this homework you will write a program to implement a 8~10-variable by Quine-McCluskey method.

Finally, your program should show **the prime implicant chart** and the **final result in minimum SOP** (Sum of Product).

2. Requirement

[1] Read the input file

Your program will read the input file for the *minterm* information and *don't care term* information:

$$\text{Eg. } F(X_0, X_1, \dots, X_n) = \Sigma m(0, 1, 3, 8, 14, 53, 103, 201, 499, \dots) \\ + \Sigma d(4, 6, 10, 77, 99, 226, \dots)$$

The format of the information are followed by [File Specification](#).

[2] Divide the minterms into groups and compare adjacent groups

You can use arrays to store the different groups of minterms and allocate new arrays to store the result of previous comparison.

[3] Write the output file

You should show the process of comparison and the prime implicant chart in the output file.

3. Input/Output Specification

[1] Programming language

You can finish your program in C, C++, Java.

Your program should be compiled as an executable file. (Need to tell TA what OS you use.)

Your program should *read input file*.

After program execution, *output file should be created* to dump the information for the simplification results.

[2] File Specification

The following is the input/output format example:

(Change the format of input/output is not allowed)

● **Input filename: input.txt**

Three lines in the input file

1. *<number>*

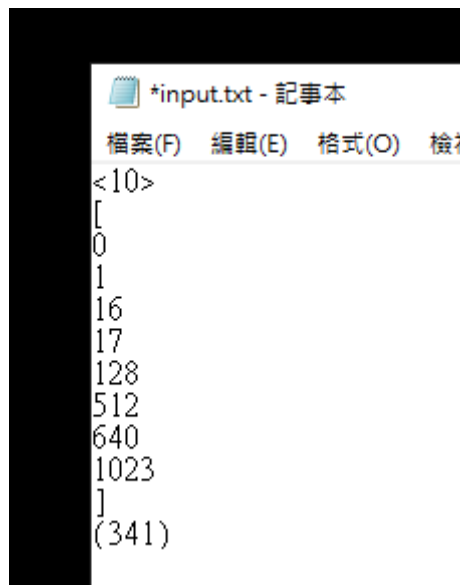
Variable number, Range: 8~10

2. *[index, index...]*

Minterm value index, Range: $0 \sim (2^N - 1)$, N = Variable number,
(The existed index stands for value 1)

3. *(index, index...)*

Don't care index, Range: $0 \sim (2^N - 1)$, N = Variable number,
(The existed index stands for value X)



- **Output filename: output.txt**

Your program should create the file like this.

Column 1

```

v 0000000000 : 0
v 0000000001 : 1
v 0000010000 : 16
v 0010000000 : 128
v 1000000000 : 512
v 0000010001 : 17
v 1010000000 : 640
d 0101010101 : 341
1111111111 : 1023

```

Column 2

```

v 000000000- : 0, 1
v 00000-0000 : 0, 16
v 00-0000000 : 0, 128
v -000000000 : 0, 512
v 00000-0001 : 1, 17
v 000001000- : 16, 17
v -010000000 : 128, 640
v 10-0000000 : 512, 640

```

Column 3

```

00000-000- : 0, 1, 16, 17
x 00000-000- : 0, 1, 16, 17
-0-0000000 : 0, 128, 512, 640
x -0-0000000 : 0, 128, 512, 640

```

Process of Grouping
Maximum column number is 11

Result:
show **prime implicant chart** and **final result** in minimum SOP

Result

	0,	1,	16,	17,	128,	512,	640,	1023
a'b'c'd'e'g'h'i'	x	x	x	x				
b'd'e'f'g'h'i'j'	x				x	x	x	
a b c d e f g h i j								x

$$F(A,B,C,D,E,F,G,H,I,J) = a'b'c'd'e'g'h'i' + b'd'e'f'g'h'i'j' + a b c d e f g h i j$$

4. Hint

You can reference the flow chart below to design your program.

