

# Logic System Assignment 2

A simple CAD tool based on Quine-McCluskey method

Due date:2021/05/16 11:59:59pm

## 1. Description

- This assignment is to let students know how to automate the process taught on textbook by writing a program of Quine-McCluskey.
- This program should be able to handle 8~10 variables.
- The prime implicant chart should be output to the designated file.
- This program should also output the Minimum SOP (Sum of Product).

## 2. Requirement

### 1. Read the input file

Your program should read the input file named “**input.txt**” containing *minterm* and *don't care term* information:

v [Variable number]: Variable number, range (8~10)

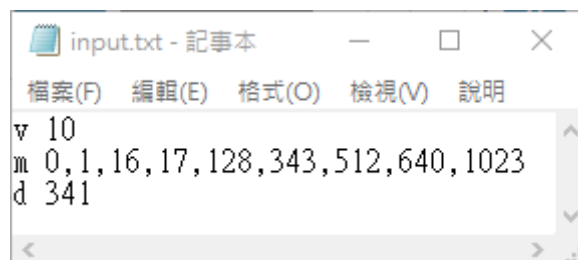
m [index,index ...]: Minterm value index, range( $0 \sim 2^V - 1$ )

d [index,index ...]: Don't care index, range( $0 \sim 2^V - 1$ )

- Variable number (v), minterm value indexes (m), don't-care indexes (d) would be in three different lines.
- Between *v*, *m*, *d* and following value(s) would be separated by a space.
- Each index would be separated by a single comma with no space.

Eg.  $F(A, B, C, D, E, F, G, H, I, J) =$

$$\Sigma m(0,1,16,17,128,343,512,640,1023) + \Sigma d(341)$$



## 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. Taking above input for example,

- Shows process of grouping with v for used, d for don't care and x for repeated
- Shows result and minimum SOP.

```

=====
Group 1
=====
v 000000000: 0
-----
v 000000001: 1
v 000001000: 16
v 001000000: 128
v 100000000: 512
-----
v 000001001: 17
v 101000000: 640
-----
d 01010101: 341
-----
v 010101011: 343
-----
111111111: 1023
=====

Group 2
=====
v 00000000-: 0,1
v 00000-0000: 0,16
v 00-0000000: 0,128
v -00000000: 0,512
-----
v 00000-0001: 1,17
v 000001000-: 16,17
v -010000000: 128,640
v 10-0000000: 512,640
-----
01010101-1: 341,343
=====

Group 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
=====

Result
=====

```

	0	1	16	17	128	343	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'j						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'bc'de'fg'hj+abcdefghij

```

### 3. Specification

- This assignment can be finished in **C, C++ or Java**.
- Requirement Summary:
  - Read an input file named “**input.txt**”
  - Write an output file named “**output.txt**”
  - Both input and output files should be put under the same directory as the program is.
- What you need to include in the zip file you submit:
  1. A source code named “**QM.\***” ( “\*” can change according to the language you use)
  2. A “**README.txt**” to indicate the **environment (programming language, compiler version, OS distribution version, etc.)** and **bonus**.
  3. A document report with some explanations, e.g., execution flow or tricks in your program, named “**report.pdf**.”
- How to upload this assignment
  1. All files should be put in a directory named “**StudentID\_HW2**” and **compress** it into **.tar or .zip**, e.g., “E24101234\_HW2.zip.”
  2. Hand in the zip file on moodle.

### 4. Grading

- Program (total 80%)
  - Basic function (70%)

Function correctness would be tested by test cases in addition to the ones provided by TA.
  - Comment (10%)
- Document Report (20%)
  - Code explanation

Flow chart of the program and usage of functions.
  - Test results
- **Late submission is allowed until 5/23 23:59, and will deduct score by 40%.**

## 5. Hint

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

