Programming Assignment 1:

Quine-McCluskey Method (Due: 10/23)

Problem Descriptions:

Implement a two-level logic optimizer based on Quine-McCluskey method. The first step is to generate all prime implicants for the given function. The second step is to choose a minimum-cost cover based on column covering method. The target is to generate a sum-of-products expression of this function with **minimum number of literals**.

Input format:

You should allow input from a file. The following is an example.

f(A,B,C,D) = $\sum$ m(4,5,6,8,9,10,13) + $\sum$ d(0,7,5).

```
.i 4           /* the function f has 4 input variables: A,B,C,D. */
.m             /* on set */
4 5 6 8 9 10 13
.d             /* don't care set */
0 7 15
```

Output format:

(1) Generate all prime implicants to a file. The following is an example. If there are more than 20 primary implicants, report only the first 20 primary implicants with the least number of literals to reduce the file size. However, you still have to report the correct number of total primary implicants at ".pi" field.

```
.pi 7          /* there are 7 prime implicants */
0-00           /* A'C'D' */
-000           /* B'C'D' */
100-           /* AB'C' */
10-0           /* AB'D'*/
01--           /* A'B */
1-01           /* AC'D */
-1-1           /* BD */
```

(2) Generate the minimum sum-of-products expression to a file. The following is an example.

```
.mc 3          /* 3 prime implicants in minimum covering */
10-0           /* AB'D'*/
1-01           /* AC'D */
01--           /* A'B */
cost=8
```

Grading:

Your grade depends on the correctness and runtime of your program. If several students generate minimum-cost results successfully, their grades are determined based on the ascending order of run time. Please compress all the source code, the output files for the benchmarks, and a simple report explaining the implementation details and your comments, and then submit to New E3 system before the deadline.