13_邏輯電路(Logic_Circuit)

(7分 /9分 / 9分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

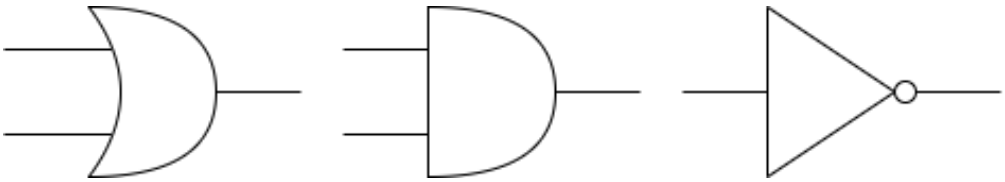
數位系統與實驗是許多大學資工系都會開設的課程，其中布林代數及數位電路的繪製是最基本的內容之一。在這道題目中，我們將完成一支程式，將輸入的邏輯運算式轉換成邏輯電路圖。還不會布林代數嗎？不用擔心，接下來我們進行簡單的教學。

在本題中，我們統一將邏輯運算式的輸出以大寫字母 `F` 作為變數名稱，並以剩下的 25 個大寫字母作為其他輸入的變數名稱。就如同程式語言中的邏輯運算，布林代數中最主要的三個運算子為 OR、AND、NOT，接下來我們將依序介紹它們。

在布林代數中，我們以類似數學中加法的表現方式表示 OR 運算。舉例來說，若將變數 `A` 和 `B` 進行 OR 運算，我們會寫作 `A+B`。而 AND 運算我們則以類似數學中乘法的表現方式表示，但我們會省略掉乘號，直接將兩個變數並列。舉例來說，若將變數 `A` 和 `B` 進行 AND 運算，我們會寫作 `AB`。最後則是 NOT 運算，我們在要取反的變數或運算式後加上一個單引號表示。舉例來說，若將變數 `A` 進行 NOT 運算，我們會寫作 `A'`。

而這三個運算子的運算優先順序也與大多數程式語言相同，首先優先計算括號內容，接下來先進行 NOT 運算，再進行 AND 運算，最後進行 OR 運算。

接下來將介紹如何將一個邏輯運算式轉換為邏輯電路圖。首先，我們先介紹三種運算子對應的邏輯閘。下圖由左至右分別為 OR、AND、NOT 的邏輯閘圖示，左側的線段表示邏輯閘的輸入，右側的線段表示邏輯閘的輸出。其中 OR 和 AND 的輸入可能有兩個或以上，而 NOT 只會有一個輸入；所有邏輯閘都只有一個輸出。



為了精確地定義輸入格式，我們保證在本題中，所有的輸入皆遵守以下巴科斯範式（Backus Normal Form）的 `<input>` 非終端符（Non-Terminal Symbol）：

```
<input> ::= “F=” <disj>
<disj>  ::= <conj> | <disj> “+” <conj>
<conj>  ::= <atom> | <conj> <atom>
<atom>  ::= <var> | <var> “'” | “(” <disj> “)” | “(” <disj> “)'”
<var>   ::= “A” | “B” | “C” | “D” | “E” | “G” | “H” | “I” | ... | “Z”
```

巴科斯範式的說明如下：

- `<input>`：表示本題的輸入字串。其中 `F` 為邏輯運算式的輸出，而 `<disj>` 則是主要的邏輯運算式內容。
- `<disj>`：由一連串的 `<conj>` 組成，`<conj>` 之間透過 `+` 字元隔開。表示邏輯運算式之間進行 OR 運算。
- `<conj>`：由一連串的 `<atom>` 組成，`<atom>` 之間沒有任何字元分隔。表示邏輯運算式之間進行 AND 運算。

- `<atom>`：可以是一個 `<var>` 或一個加了左右括號的 `<disj>`，並且可能加上了單引號表示對其進行 NOT 運算。
- `<var>`：表示邏輯運算式的輸入的變數名稱。注意 `F` 不在此列表內。

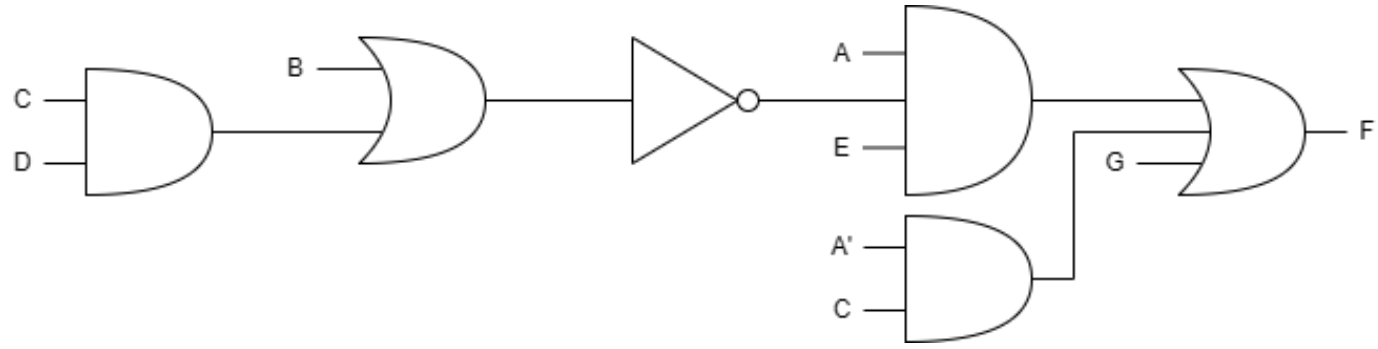
而以下是每個非終端符繪製成邏輯電路圖的方式：

- `<input>` 中的 `<disj>` 為主要的電路，其輸出為變數 `F`。
- 對於每個 `<disj>`，我們將其分解成一連串的 `<conj>`。
  - 如果 `<disj>` 只由一個 `<conj>` 組成，則將其視為一個 `<conj>`。
  - 否則，將這些 `<conj>` 的輸出，由上到下依序接到一個 OR 邏輯閘作為其輸入。
- 對於每個 `<conj>`，我們將其分解成一連串的 `<atom>`。
  - 如果 `<conj>` 只由一個 `<atom>` 組成，則將其視為一個 `<atom>`。
  - 否則，將這些 `<atom>` 的輸出，由上到下依序接到一個 AND 邏輯閘作為其輸入。
- `<atom>` 則依照不同情況進行繪製。
  - `<var>`：為一變數，寫下該變數的名稱。
  - `<var> “'”`：為一變數取反，寫下該變數的名稱，並加上單引號。
  - `“(” <disj> “)”`：為一個新的子運算式，遞迴繪製此運算式的內容。
  - `“(” <disj> “)'”`：為一個新的子運算式並取反，遞迴繪製此運算式的內容，並將其輸出接上一個 NOT 邏輯閘。

舉例來說，輸入 `F=A(B+CD)'E+A'C+G`，則繪製邏輯電路圖的過程如下：首先得到輸入的主要電路為 `A(B+CD)'E+A'C+G`。將其分解成三個 `<conj>`：`A(B+CD)'E`、`A'C`、`G`。

- 將 `A'(B+CD)'E` 分解成三個 `<atom>`：`A'`、`(B+CD)'`、`E`。
  - `A'` 為一變數取反，寫下變數名稱，並加上單引號。
  - `(B+CD)'` 為運算式取反，遞迴繪製運算式 `B+CD`，並將其輸出接上一個 NOT 邏輯閘。
  - `E` 為一變數，寫下變數名稱。
- 將 `A'C` 分解成兩個 `<atom>`：`A'`、`C`。
  - `A'` 為一變數取反，寫下變數名稱，並加上單引號。
  - `C` 為一變數，寫下變數名稱。
- `G` 只由一個 `<atom>` 組成，將其視為一個 `<atom>`。此 `<atom>` 為一變數，寫下變數名稱。

繪製出的電路電路如下圖：



學會了如何繪製之後，最後我們要來將邏輯電路圖利用 ASCII Art 的方式輸出到螢幕上！接下來我們會先說明如何繪製三種不同的邏輯閘，最後再說明將其接成電路圖時，邏輯閘之間的排列方式。

首先是 OR 邏輯閘。對於一個有 n 個輸入的 OR 邏輯閘，其會佔用一個 $2n+2$ 行高，7 個字元寬的矩形範圍。邏輯閘第 1 行的第 $2,3,4$ 個字元為底線（`_`）；第 2 行的第 $1,5$ 個字元為反斜線（`\`）；第 3 行的第 1 個字元為減號（`-`），第 $2,6$ 個字元為反斜線（`\`）；第 4 到第 $2n$ 行的第 $2,7$ 個字元為豎線（`|`），並且對於這之間的所有奇數行，第 1 個字元為減號（`-`）；第 $2n+1$ 行的第 1 個字元為減號（`-`），第 $2,6$ 個字元為斜線（`/`）；第 $2n+2$ 行的第 $1,5$ 個字元為斜線（`/`），第 $2,3,4$ 個字元為底線（`_`）。

舉例來說，下方由左到右分別是有 $2,3,4$ 個輸入的 OR 邏輯閘。邏輯閘的輸入位於第 3 行開始的每個奇數行的左側，而輸出則位於第 $n+2$ 行的右側。請注意，左側的減號也視為邏輯閘的一部分。

```
 ___         ___         ___
\   \       \   \       \   \
-\   \      -\   \      -\   \
 |    |      |    |      |    |
-/   /      -|    |     -|    |
/___/        |    |      |    |
            -/   /      -|    |
            /___/        |    |
                        -/   /
                        /___/
```

接下來是 AND 邏輯閘。對於一個有 n 個輸入的 AND 邏輯閘，其會佔用一個 $2n+2$ 行高，7 個字元寬的矩形範圍。邏輯閘第 1 行的第 $2,3,4$ 個字元為底線（`_`）；第 2 行的第 1 個字元為豎線（`|`），第 5 個字元為反斜線（`\`）；第 3 行的第 1 個字元為豎線（`|`），第 6 個字元為反斜線（`\`）；第 4 到第 $2n$ 行的第 $1,7$ 個字元為豎線（`|`）；第 $2n+1$ 行的第 1 個字元為豎線（`|`），第 6 個字元為斜線（`/`）；第 $2n+2$ 行的第 1 個字元為豎線（`|`），第 $2,3,4$ 個字元為底線（`_`），第 5 個字元為斜線（`/`）。

舉例來說，下方由左到右分別是有 $2,3,4$ 個輸入的 AND 邏輯閘。邏輯閘的輸入位於第 3 行開始的每個奇數行的左側，而輸出則位於第 $n+2$ 行的右側。

```
 ___         ___         ___
|   \       |   \       |   \
|    \      |    \      |    \
|     |     |     |     |     |
|    /      |     |     |     |
|___/       |     |     |     |
           |    /      |     |
           |___/       |     |
                       |    /
                       |___/
```

接下來是 NOT 邏輯閘。一個 NOT 邏輯閘會佔用一個 6 行高，7 個字元寬的矩形範圍。邏輯閘第 1 行的第 $2,3$ 個字元為底線（`_`）；第 2 行的第 1 個字元為豎線（`|`），第 4 個字元為反斜線（`\`）；第 3 行的第 1 個字元為豎線（`|`），第 5 個字元為反斜線（`\`）；第 4 行的第 1 個字元為豎線（`|`），第 6 個字元為右括號（`)`），第 7 個字元為小寫字母 `o`；第 5 行的第 1 個字元為豎線（`|`），第 5 個字元為斜線（`/`）；第 6 行的第 1 個字元為豎線（`|`），第 $2,3$ 個字元為底線（`_`），第 4 個字元為斜線（`/`）。

下方為一個 NOT 邏輯閘。邏輯閘的輸入位於第 4 行的左側，而輸出則位於第 4 行的右側。

```
 __
|  \
|   \
|    )o
|   /
|__/
```

接下來我們要說明如何輸出一個變數。每個變數或變數取反的輸出會佔用一個 1 行高，4 個字元寬的矩形範圍。第 1 個字元為變數名稱，第 2 個字元若為變數取反則為單引號（`'`），否則留空，第 $3, 4$ 個字元為減號（`-`）。

舉例來說，下方分別是變數 `A` 及其取反，其輸出位於右側。

```
A --    A'--
```

最後，我們要來說明如何將邏輯閘組裝成完整的邏輯電路。我們會用遞迴的方式進行繪製邏輯電路：先繪製邏輯閘，再由上往下依序繪製每一個輸入接到的邏輯電路。為了方便進行電路繪製時是否有重疊區域的判斷，我們定義一個邏輯電路佔據的矩形區域為，最小可以覆蓋住所有邏輯閘及變數和變數取反的矩形區域。

對於一個有 n 個輸入的邏輯閘由上往下的第 i 個輸入，若該輸入為一變數或變數取反，則直接繪製變數或變數取反的矩形於邏輯閘左側，與輸入位置相同的高度。若該輸入為另一個電路，則其矩形應於邏輯閘的左側，兩者左右之間留下 $2n + 3$ 個空格作為稍後進行連接使用。接下來決定電路的上下位置，首先將電路上下移動，使其輸出位置的高度與邏輯閘第 i 個輸入的高度相同，再調整高度，若其與其他已繪製的電路所佔的矩形有重疊部分，則向下移動直到沒有重疊為止。最後進行邏輯閘間的連接，如果電路的輸出與邏輯閘的輸入高度相同，則在兩者左右之間的空格補上 $2n + 3$ 個減號（`-`）；如果兩者高度不同，則先在電路輸出的右側空格由左至右依序補上 $2i$ 個減號（`-`）及 1 個加號（`+`），並在邏輯閘輸入的左側由右至左依序補上 $2(n - i) + 2$ 個減號（`-`）及 1 個加號（`+`），此時兩個加號應位於同一欄，最後在兩加號上下之間的空格補滿豎線（`|`）。

繪製完完整的邏輯電路後，在電路的輸出位置依序加上 2 個減號（`-`）及 1 個空格，最後寫上變數 `F`，表示整個電路的輸出。

舉例來說，繪製 `F=A'B+(CD)'` 的邏輯電路圖時，我們先繪製一個 OR 邏輯閘，接下來依序繪製 `A'B` 和 `(CD)'`。

繪製 `A'B` 時，先繪製一個 AND 邏輯閘，並且由於 `A'` 及 `B` 分別為變數取反及變數，我們直接將其接上 AND 邏輯閘，完成 `A'B` 的繪製。`A'B` 電路圖繪製如下：

```
     ___
    |   \
A'--|    \
    |     |
B --|    /
    |___/
```

而 `(CD)'` 我們會先繪製一個 NOT 邏輯閘，再繪製 `CD` 部分。由於 NOT 邏輯閘只有 1 個輸入，其與 `CD` 部分之間間隔為 5 格，並且由於電路輸出及邏輯閘輸入高度相同，故用減號補上之間的空格。`(CD)'` 部分的電路與電路圖繪製如下：

```
     ___          __
    |   \        |  \
C --|    \       |   \
    |     |------|    )o
D --|    /       |   /
    |___/        |__/
```

最後，依序將兩者接上 OR 邏輯閘。`A'B` 與 OR 邏輯閘左右之間應間隔 7 格，由於電路輸出與邏輯閘輸入之間高度相同，故用減號補上之間的空格。而 `(CD)'` 與 OR 邏輯閘左右之間應間隔 7 格，並且為了避免與上方的 `A'B` 重疊，`(CD)'` 會向下移動 3 行，並依上述格式連接至 OR 邏輯閘。最後再補上電路輸出 `F` 即大功告成。

```
                ___
               |   \           ___
           A'--|    \         \   \
               |     |--------\   \
           B --|    /          |    |-- F
               |___/       +---/   /
     ___         __        |  /___/
    |   \       |  \       |
C --|    \      |   \      |
    |     |-----|    )o----+
D --|    /      |   /
    |___/       |__/
```

接下來便輪到你寫一支程式進行轉換了，加油！

輸入格式

輸入僅包含一字串 S，表示要轉換的邏輯運算式。

輸出格式

輸出的第一行包含兩個整數 H, W，以一個空白隔開。依序表示邏輯電路圖的行數及每一行的字元數。

接下來請輸出 H 行，每行包含 W 個字元，為轉換後的邏輯電路圖。

請注意，你的輸出必須為邏輯電路圖佔據的最小矩形區域，並且所有的空白都必須輸出，包含行尾空白。此題將以嚴格比對的方式評測。

資料範圍

- $3 \leq |S| \leq 100$
- 保證 S 遵守題目敘述中的巴科斯範式

子任務

- 子任務 1 滿足 $|S| \leq 5$
- 子任務 2 滿足 S 中不會有任何單引號（`'`）
- 子任務 3 無額外限制

測試範例

輸入範例 1

```
F=A'B+(CD)'
```

輸出範例 1

```
12 41
                ___
               |   \          ___
           A'--|    \        \   \
               |     |--------\   \
           B --|    /          |    |-- F
               |___/       +---/   /
     ___        __         |  /___/
    |   \       |  \       |
C --|    \      |   \      |
    |     |-----|    )o----+
D --|    /      |   /
    |___/       |__/
```

輸入範例 2

```
F=A(B+(C+D))'E+A'C+G
```

輸出範例 2

```
14 73
                                          ___
                  ___         __           |   \
     ___         \   \       |  \       A --|    \           ___
    \   \     B ---\   \      |   \          |     |         \   \
C ---\   \         |    |-----|    )o--------|     |----------\   \
     |    |--------/   /      |   /          |     |          |    |
D ---/   /        /___/       |__/       E --|    /     +-----|    |-- F
    /___/                                    |___/      |     |    |
                                              ___       |G ---/   /
                                             |   \      |    /___/
                                          A'--|    \     |
                                             |     |----+
                                          C --|    /
                                             |___/
```

輸入範例 3

```
F=((A+B)'(CD)+E)G((HI+J)+K')LM(OP)
```

輸出範例 3

```
26 77
     ___          __
    \   \        |  \            ___
A ---\   \       |   \          |   \           ___
     |    |------|    )o--------|    \         \   \                  ___
B ---/   /       |   /          |     |--------\   \                |   \
    /___/        |__/        +--|    /          |    |----------------|    \
                  ___        |  |___/     E ---/   /                 |     |
                 |   \       |                 /___/             G --|     |
            C --|    \      |                                       |     |
                 |     |----+                               +--------|     |
            D --|    /                                      |        |     |-- F
                 |___/                                      |    L --|     |
                  ___                                       |        |     |
                 |   \           ___                        |    M --|     |
            H --|    \         \   \           ___          |        |     |
                 |     |--------\   \         \   \         |     +--|    /
            I --|    /          |    |--------\   \        |     |  |___/
                 |___/     J ---/   /          |    |------+     |
                             /___/     K'---/   /              |
                                          /___/               |
                                           ___                |
                                          |   \               |
                                     O --|    \              |
                                          |     |------------+
                                     P --|    /
                                          |___/
```

輸入範例 4

```
F=B+(B)+(B')'+((B)')'
```

輸出範例 4

```
15 45
                              ___
                             \   \
                        B ---\   \
                 __           |    |
                |  \          B ---|    |
                |   \              |    |-- F
            B'--|    )o------------|    |
                |   /              |    |
                |__/           +---/   /
```

```
     __          __            |  /___/
    |  \        |  \           |
    |   \       |   \          |
B --|    )o-----|    )o--------+
    |   /       |   /
    |__/        |__/
```

輸入範例 5

```
F=X
```

輸出範例 5

```
1 8
X ---- F
```

範例說明

輸出範例 3 中，注意 `B'` 及 `(B)'` 根據巴科斯範式的定義，兩者繪製出的電路圖是不同的。

另外，本題無須簡化電路圖或運算式，請直接將給定的邏輯運算式依照題目敘述中的規則繪製出邏輯電路圖。