Little Quilt is a small language introduced by Ravi Sethi in his book 'Programming Languages'. Here, a restricted version of Little Quilt is presented.

The language is defined by the following BNF grammar:

```
<QUILT> ::= A | B | turn(<QUILT>) | sew(<QUILT>,<QUILT>)
```

`A` and `B` represent the two primitive quilts. Each primitive quilt corresponds to a matricial arrangement of 2 × 2 characters. `turn()` and `sew()` are operations over quilts.

The instruction `turn(x)` turns the quilt `x` 90 degrees clockwise. The following table illustrates the primitive quilts as well as examples of the effect of the `turn()` operation:

`A`	`//` `/+`
`turn(A)`	`\\` `+\`
`turn(turn(A))`	`+/` `//`
`turn(turn(turn(A)))`	`\+` `\\`
`B`	`--` `--`
`turn(B)`	`||` `||`

Accordingly, the instruction `sew(x,y)` sews quilt `x` to the left of quilt `y`. Both `x` and `y` must have the same height, otherwise an error will be generated. The following figure represents the result of `sew(A,turn(B))`:

```
//||
/+||
```

while the `sew(turn(sew(B,turn(B))),A)` generates an error message.

Your job is to build an interpreter of the Little Quilt language.

Input

The input file will be a text file containing different Little Quilt expressions, each one ended by a semicolon character (`;`). Space and new line characters must be ignored; this means that an expression may span several lines.

Output

The output file contains the quilts produced as a result of interpreting the input expressions.

Each quilt must be preceded by a line, left aligned, with the format

`Quilt` *i*`:`

where *i* is the quilt number, starting at 1. If the expression interpretation generates and error, the word

'`error`'

must be printed.

Sample Input

```
sew(turn(sew(B,turn(B))),
       turn(sew(turn(B),B)))    ;


   sew(turn(sew(B,turn(B))),A);
sew(turn(sew(A,turn(A))),
 turn(turn(

   turn(sew(A,turn(A))))))

   ;
```

Sample Output

```
Quilt 1:
||--
||--
--||
--||
Quilt 2:
error
Quilt 3:
\\//
+\/+
+/\+
//\\
```