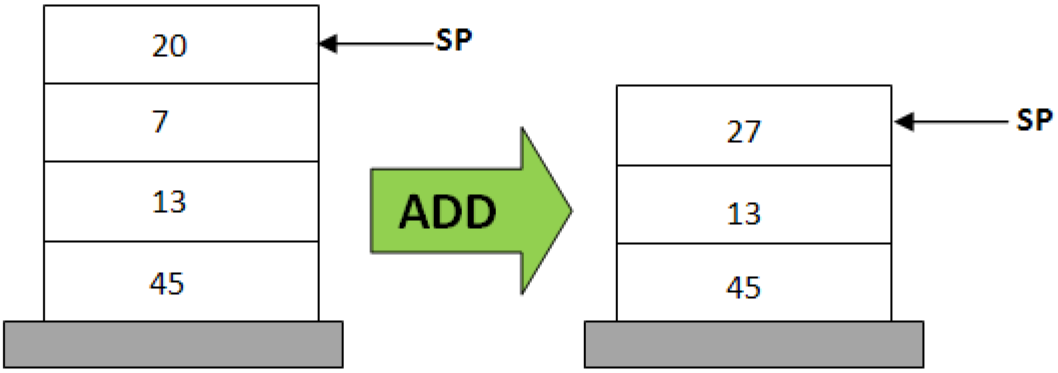
Stack-Base Machine

Description

Stack-base machine is a kind of CPU design architecture which sometimes be compared with register-based machine. Unlike register-based machine which used 3 address code taking 1 opcode and 2 corresponding operands e.g. add %eax, 3, stack based machine was designed with opcode with one or zero operand and one long stack which served as registers to take temporary value in calculation.
It works as follow:



As the above figure, we may first perform 4 instructions: load 45, load 13, load 7, load 20, then the stack will layout like the stack on left side (note that the "load" instruction is the only opcode that take 1 operand). At this time, if we continue to perform add instruction which have 0 operand, it will pop two number (20, 7) from top of stack then perform add with those number. Finally push the result (27) back to the stack which will have final layout like the stack on the right side.

In this problem, you will be given a program for stack-based machine. You need to write a yacc program to check if the format of the program and evaluate its result. The instruction set of our machine are: "add", "sub", "mul", "mod" which will pop 2 operands from stack then perform addition, subtraction, multiplication and modulation and push back the result, "inc", "dec" which will pop 1 operand from stack then perform increased by 1, decreased by 1, then push back the result, "load <number>" which will push a constant <number> on the top of the stack.

Sample Input

```
load 1
load 2
sub
load 5
mod
```

Sample Output

```
0
```

Explanation

5 % (2-1) = 0

Sample Input

```
load 3
load 4
add
inc
inc
```

Sample Output

```
Invalid format
```

Explanation

If after finish the program, there are more than 1 numbers in the stack it will be consider invalid.

Sample Input

```
load 1
load 1
add
sub
```

Sample Output

```
Invalid format
```

Explanation

No enough operands in stack to perform sub.