2003-2004

American Computer Science League

Contest #4

Senior Division
ASSEMBLY

PROBLEM: ACSL Assembly Language programming is one of the ACSL categories and will be tested at the All-Star Contest. You should have a program to test your answers to your practice questions. So we are going to let you write an ACSL Assembly Language compiler. For this version of the compiler only the commands listed in the chart below will be used.

Execution starts at the first line of the program and continues sequentially, except for "branch" instructions, until the "end" instruction is encountered. The result of each operation is stored in a special word of memory, called the "accumulator" (ACC). Each line of an assembly language program has the following fields (lower-case italics indicates optional components):

label **OPCODE LOC**

The *label* is a character string beginning in the first column. Valid **OPCODE's** are listed in the chart below. The LOC field is either a reference to a label or "immediate data". For example, "LOAD A" would put the contents referenced by the label "A" into the ACC; "LOAD =123" would store the value 123 in the ACC. Only those instructions that do not modify the LOC field can use the "immediate data" format. In the following chart, they are indicated by an asterisk in the first column.

OPCODE	ACTION
*LOAD	Contents of LOC are placed in the ACC. LOC is unchanged.
STORE	Contents of ACC are placed in the LOC. ACC is unchanged.
*ADD	Contents of LOC are added to the contents of the ACC. The sum is stored in the ACC. LOC is unchanged. Addition is modulo 1,000,000.
*SUB	Contents of LOC are subtracted from the contents of the ACC. The difference is stored in the ACC. LOC is unchanged. Subtraction is modulo 1,000,000.
*MULT	The contents of LOC are multiplied by the contents of the ACC. The product is stored in the ACC. LOC is unchanged. Multiplication is modulo 1,000,000.
*DIV	Contents of LOC are divided into the contents of the ACC. The signed integer part of the quotient is stored in the ACC. LOC is unchanged.
BE	Branch to instruction labeled with LOC if ACC=0.
BG	Branch to instruction labeled with LOC if ACC>0.
BL	Branch to instruction labeled with LOC if ACC<0.
BU	Branch unconditionally to instruction labeled with LOC.
END	Program terminates. LOC field is ignored.
READ	Read a signed integer (modulo 1,000,000) into LOC.
PRINT	Print the contents of LOC.
DC	The value of the memory word defined by the LABEL field is defined to contain the specified constant. The LABEL field is *mandatory* for this opcode. The ACC is not modified.

INPUT: The input for this program will be a valid ACSL Assembly Language program. Each line will contain one command. Each line must be entered as a string. All *labels* will be single characters. There will be 5 PRINT statements in the program. Data for READ statements will follow the END statement. The data does not have to be read as a string.The program will have a maximum of 50 lines. Each program line string can be entered with or without spaces. Insure that your advisor knows how to enter the data for your program.

OUTPUT: For each PRINT statement in the program, print the current value of the listed LOC.

SAMPLE INPUT

```
A       DC      1
        READ    B
        READ    C
        PRINT   C
        LOAD    = 1
        MULT    C
        BE      D
        STORE   A
        PRINT   A
        DIV     B
        BL      D
        STORE   C
        PRINT   C
        SUB     A
        BG      D
        STORE   B
        PRINT   B
        ADD     A
        STORE   A
        PRINT   A
D       END
2,3
```

SAMPLE OUTPUT

1. 3
2. 3
3. 1
4. -2
5. 1