2018-2019

AMERICAN COMPUTER SCIENCE LEAGUE

All-Star Contest

7. ACSL Assembly Language

PROBLEM: Write an ACSL Assembly Language interpreter.

INPUT: The input for this program will be 10 valid ACSL Assembly Language programs. Each program will be preceded by a line with one or more integers separated by one or more spaces. These integers are input to the program if and when the **READ** opcode is executed. Each program will end with a blank line, or the end of the input file. Please review the ACSL Assembly Language reference guide attached. The fields on each line are separated by a combination of spaces and tabs.

OUTPUT: Print the value of the ACC when the program ends.

SAMPLE INPUT: (*2 programs; the Test Data will have 10 programs*)**:**

```
0
A      DC     8
B      DC     -2
C      DC     3
       LOAD   B
       MULT   C
       ADD    A
       DIV    B
       SUB    A
       END

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

SAMPLE OUTPUT:

1. -9
2. 1

ACSL Assembly Language Reference Guide

Each line of an ACSL Assembly Language program has 3 components: a **label** (usually optional), an **opcode** (always required), and a **loc** field (nearly always required).

- The **label**, if present, is an alphanumeric character string beginning in the first column. A label must begin with an alphabetic character(A through Z, or a through z), and labels are case-sensitive. The label field is required for the `DC` opcode; it is optional for all other opcodes.
- Valid **opcodes** are listed in the chart below; they are uppercase and case-sensitive. Opcodes are reserved words of the language and may not be used as a label.
- The **loc** field is either a reference to a label (e.g., "`ADD A`") or immediate data (e.g., "`LOAD =123`"). Only those opcodes with an asterisk in the following chart are allowed to use the immediate data format of the loc field. The loc field is required for all opcodes, except for the `END` opcode. It is prohibited on the `END` opcode.

OP CODE	DESCRIPTION
*LOAD	The contents of LOC are placed in the ACC. LOC is unchanged.
STORE	The contents of the ACC are placed in the LOC. ACC is unchanged.
*ADD	The 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	The 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	The 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.
BG	Branch to the instruction labeled with LOC if ACC>0.
BE	Branch to the instruction labeled with LOC if ACC=0.
BL	Branch to the instruction labeled with LOC if ACC<0.
BU	Branch to the instruction labeled with LOC.
READ	Read a signed integer (modulo 1,000,000) into LOC.
PRINT	Print the contents of LOC.
DC	The value of the memory slot defined by the LABEL field is defined to contain the specified constant. The LABEL field is mandatory; the ACC is not modified.
END	Program terminates. LOC field is ignored and must be blank.

TEST DATA

TEST INPUT:

```
0
TMP       DC        0
A         DC        -1
B         DC        213
          LOAD      A
          MULT      =12
          STORE     TMP
          LOAD      B
          MULT      A
          DIV       TMP
          END
```

```
-3 2 1 -4 5 -9 8 0 7
A         DC        0
B         DC        0
D         READ      N
          LOAD      N
          BE        E
          BG        C
          LOAD      B
          ADD       N
          STORE     B
          BU        D
C         LOAD      A
          ADD       N
          STORE     A
          BU        D
E         LOAD      B
          SUB       A
          END
```

7. ACSL Assembly Language

```
0
A         DC        5
B         DC        4
C         DC        12
          LOAD      A
          ADD       B
          SUB       C
          BG        ONE
          BL        TWO
ONE       DIV       B
          END
TWO       STORE     X
          LOAD      C
          MULT      X
          BU        ONE
```

```
0
NSUB      DC        3
TOTAL     DC        210
CNT       DC        0
TOP       LOAD      CNT
          ADD       =1
          STORE     CNT
          LOAD      NSUB
          MULT      =2
          STORE     NSUB
          LOAD      TOTAL
          SUB       NSUB
          STORE     TOTAL
          BG        TOP
          LOAD      CNT
          END
```

7. ACSL Assembly Language

```
5 -3 6 2 -9 7 1 4 -8
SUM       DC        0
J         DC        6
TOP       READ      VALUE
          LOAD      VALUE
          MULT      VALUE
          ADD       SUM
          STORE     SUM
          LOAD      J
          SUB       =1
          STORE     J
          BE        STOP
          BU        TOP
STOP      LOAD      SUM
          END
```

```
15 10
A         DC        0
          READ      B
          READ      C
T         DC        0
ST        LOAD      T
          ADD       =1
          STORE     T
          LOAD      B
          ADD       C
          STORE     A
          LOAD      B
          ADD       A
          STORE     B
          SUB       =1000
          BL        ST
          END
```

7. ACSL Assembly Language

```
1 -1 -12 1 2 10 3 -5 1 1 -14 49 0 8 100
S         DC        0
P         DC        0
TWO       READ      F
          READ      M
          READ      L
          LOAD      F
          BE        DOWN
          LOAD      M
          MULT      M
          STORE     S
          LOAD      L
          MULT      F
          MULT      =4
          STORE     P
          LOAD      S
          SUB       P
          BG        TWO
          BL        TWO
          LOAD      M
          MULT      =-1
          DIV       =2
          DIV       F
DOWN      END
```

7. ACSL Assembly Language

```
4213
          READ      NUM
B         DC        16
SUM       DC        0
          LOAD      NUM
TOP       DIV       B
          STORE     C
          BE        DOWN
          LOAD      C
          MULT      B
          STORE     E
          LOAD      NUM
          SUB       E
          STORE     F
          ADD       SUM
          STORE     SUM
          LOAD      C
          STORE     NUM
          BU        TOP
DOWN      LOAD      SUM
          ADD       NUM
          END
```

7. ACSL Assembly Language

```
-16 4
X         DC        4
Y         DC        6
A         READ      R
          ADD       X
          STORE     Z
          SUB       Y
          BG        A
          ADD       Y
W         DC        1
          LOAD      X
          ADD       W
          STORE     X
          LOAD      Y
          DIV       X
          STORE     Z
          LOAD      =-3
V         ADD       W
          BE        T
          BL        V
T         STORE     Z
          READ      S
          LOAD      =-10
          ADD       S
          SUB       R
          DIV       X
          MULT      Y
          END
```

7. ACSL Assembly Language

```
100 999
A         DC        0
          READ      B
          READ      E
YELLOW    LOAD      B
          DIV       =10
          STORE     C
          MULT      =10
          STORE     F
          LOAD      B
          SUB       F
          STORE     D
          LOAD      B
          DIV       =100
          SUB       D
          STORE     G
          BE        RED
BLUE      LOAD      B
          SUB       E
          BL        GREEN
          LOAD      A
          END
RED       LOAD      A
          ADD       =1
          STORE     A
          BU        BLUE
GREEN     LOAD      B
          ADD       =1
          STORE     B
          BU        YELLOW
```

7. ACSL Assembly Language

TEST OUTPUT:

1. 17

2. -32

3. -9

4. 6

5. 204

6. 590

7. 7

8. 13

9. 12

10. 90