5. Pretty Print - ACSL Assembly Language

PROBLEM: Given a valid ACSL Assembly Language program but with the spacing and line breaks not aligned, format the program using the following rules:

- Each command is on a separate line.
- Each line has up to 3 left-justified columns: The **label**, **opcode**, and **loc** fields. Some lines will not have a **label.** Some lines will not have a **loc**.
- The width of the **label** and **opcode** columns is the width of the maximum **label** or **opcode**, respectively.
- A single space separates the maximum **label** or **opcode** from the next column. Multiple spaces will separate the columns which are not the maximum length; do not use tabs.
- The **label** and **loc** fields are case-sensitive; do not change them from what is input. The **opcode** must be in all capital letters.
- The line ends at the end of the **loc** field, or the end of the **opcode**, for those commands that do not allow **loc** fields.

Here is an example:

Unformatted	Formatted

`A  DC  8`
`B DC  -2  C DC 3`
`LOAD B mult  C ADD A`
`DIV =2`
`sub A`
`end`
	`A DC  8`
	`B DC  -2`
	`C DC  3`
	`LOAD B`
	`MULT C`
	`ADD  A`
	`DIV  =2`
	`SUB  A`
	`END`

INPUT: The input for this program will be a set of valid ACSL Assembly Language programs. Each will be preceded by a line with one integer, *K*, and each program will end with a single blank line, or the end of the input file. The input program may contain both spaces and tabs.

OUTPUT: Print the *K*th line of the formatted program replacing spaces with a number representing the number of spaces in that part of the line.

SAMPLE INPUT:

(*2 programs only; the Test Data will have 10 input programs*)

```
5
A  DC  8
B DC  -2  C DC 3
LOAD B mult  C ADD A
DIV B
sub A
End
4
Beta19 READ Alpha
PRINT Alpha
```

SAMPLE OUTPUT:

(*Output must match exactly*)

1. 2MULT1C

2. 7BU4Beta19

BG Done BU Beta19 Done END

TEST DATA

TEST INPUT:

```
11
               Read    Num
A  DC   1   Start    LOAD        Num
SUB    =1      BE     Rslt
Store   Num     Load     A    ADD         =2
STORE A     BU  Start
Rslt END

3
Read input lOAD input  Top           Sub =1
BE   done    SToRE   Answer  bu  Top    done
end

10
LOAD value                     SUB num    Bl low    equal
LoaD   value    add     num       bG     high   low   Load
value   SUB    value          BE equal   high  Store   ans   Print
    ans   end    value dc
45        num DC    47

14
LOAD Aval Sub bValue Store Ans bu lbl3 l2 LOAD Aval Add bValue Store
Ans bu printIt lbl3 load bValue Mult Aval store ans BU l2 done end
printIt print ans BU done Aval dc 5 bValue dc 3

4
LOAD   aX
DIV Byz      store aX     Top  LOAD aX   div
Byz   STORE aX          LOAD
number       Sub    =2   STORE number      BG     Top        PRINT
number
END     Ax    dc
31      Byz  DC  2   number
DC  10
```

5. Pretty Print - ACSL Assembly Language

```
15
factor      DC       2    num     DC
288     count
dc     0           Top    LOAD num
DIV  factor     store    num
LOAD count     SUB =1   Store   count   BE
     Part     LOAD      factor          ADD      =1       STORE
factor     BU Top     Part   LOAD count    print   count          END

10
start     dc     10    part    DC    2        stop
DC         5    Start
LOAD   start     SUB     part      BG Part
  LOAD      part       MULT          stop
    BL       Stop    Part         LOAD     start
add      stop     BE  Start    Stop
LOAD start   DIV part   store     stop     PRINT    stop    END

7
     minimum    dc 10000   top      read number
load number be stop sub
    minimum     bl  change  stop
print    minimum      end    change
    store       minimum      bu    top

6
CX    dc    2   Num    Dc
   288    TOP   Load    Num
div  CX  Mult  CX      Store
   X   Load   Num
SUB   X     BE   PART   LOAD CX    Add
=1       Store    CX   BU
TOP   PART   Print     CX
Load   Num
sub        CX   be     TOP    Load    Num   DIV   CX   Store    Num
bu    TOP    Stop
          End
```

5. Pretty Print - ACSL Assembly Language

```
22
CNT dc 10    MAXI dc
0   load     =0   store     TOTAL   Start    read     VAL  load
TOTAL   add    VAL   store     TOTAL   load    CNT    sub     =1
store     CNT     load     VAL    sub    MAXI   bu
Start     Next    load   TOTAL     div   CNT   store      AVG
print      AVG     bg       Final      Stop
end      Final     print    MAXI    bu        Stop
```

TEST OUTPUT: (*Output must match exactly. Same capitalization; no embedded spaces; etc.):*

1. Rslt2END
2. Top2SUB3=1
3. high2STORE1ans
4. printIt1PRINT1ans
5. Top4LOAD2aX
6. Part3LOAD2count
7. Part2LOAD2start
8. stop4PRINT1minimum
9. 5STORE1X
10. 6BU4Stop