大视野在线测评

F.A.Qs Home Discuss ProblemSet Status Ranklist Contest ModifyUser Logout 捐

free_bzoj 赠

本

站

Notice: 1:由于本OJ建立在Linux平台下,而许多题的数据在Windows下制作,请注意输入、输出语句及数据类型及范围,避免无谓的RE出现。 2:本站即将推出针对初学者的试题系统(与目前OJ是分开的,互不影响),内容覆盖从语法入门到NOI的所有知识点,敬请关注。

1238: [SCOI2008]警告warning

Time Limit: 20 Sec Memory Limit: 162 MB

Submit: 26 Solved: 19 [Submit][Status][Discuss]

Description

有一个简单的脚本语言,只有赋值语句、条件语句和返回语句三种。变量名 必须是单个大写字母,且变量都是32位带符号整数。

该语言的每条语句必须单独占一行。程序中不含空行,且每行的行首行末均 无空格。每行的不同 token 之间用单个空格隔开。该语言的 BNF 如下:

:: <head> | <assignment> | <if> | ELSE | END IF | <return>

(head) :: PARAM (paramlist) | PARAM

<assignment> :: <variable> = <rvalue>

:: IF <variable> <relation> <value> THEN

<return> :: RETURN <value>

⟨paramlist⟩ :: ⟨variable⟩ | ⟨variable⟩ ⟨paramlist⟩ <rvalue> :: <value> | <value> <operator> <value>
<value> :: <variable> | <integer>

<operator> :: + | - | * | / <relation> :: < | = | > <variable> :: A | B | ... | Z

(integer) :: 不含前导 0 的 32 位带符号整数

程序的第一行是一条<head>语句,定义了函数的参数,而最后一行一定是 <return>语句。<head>语句不能在除了第一行之外的其他任何地方出现,但 <return>语句可以在程序中多次出现。行号从1开始编号。

每条IF语句一定有一个配套的 END IF语句,还有一个可选的 ELSE语句(注 意没有 ELSE IF 语句)。IF 语句可以嵌套,它总是比较一个变量和一个整数或者 另一个变量。

你应该分析一个给定的程序,并输出两类警告信息(格式见样例输出):

- 第一类警告:无法到达的代码行。不管各条 IF 语句的布尔表达式是真还是 假(假设每条 IF 语句的布尔表达式都是既可能为真也可能为假,不受其他 IF 语句结果影响)。
- 第二类警告:可能未初始化的变量。该语句用到了某个变量的值,但这个变 量既不在第一行指定的参数列表里,也没有在此之前由赋值语句赋值过。如 果这条语句无法到达,那么不应该给出这类警告。

注意,语句 ELSE 和 END IF 不是可执行语句,因此不应该收到任何警告信息。

Input

最多包含50行,即你要处理的程序。保证该程序合法

Output

警告按照行号从小到大排序。如果同一行内有多个可能未初始化的变量,按 照字 母顺序从小到大排列。如果没有任何警告信息,你的输出应该为空。

Sample Input

输入	输出
ARAM A B	Line 5: variable C might not have been
IF A > 5 THEN	initialized
C = B * A	Line 6: variable X might not have been
END IF	initialized
D = B - C	Line 6: variable Y might not have been
Z = Y + X	initialized
E = T	Line 7: variable T might not have been
F = E + E	initialized
V = G + G	Line 9: variable G might not have been
RETURN F	initialized
PARAM G	Line 3: unreachable code
RETURN G	Line 4: unreachable code
B = K	
RETURN C	
PARAM T C	Line 6: variable F might not have been
B = T	initialized
A = 4	Line 11: variable V might not have
IF A < 4 THEN	been initialized
IF B > 3 THEN	Line 14: variable I might not have
Q = 100 + F	been initialized
ELSE	Line 15: unreachable code
IF $C = -111111111111$ THEN	
Q = T - A	
IF Q = 0 THÉN	
V = V - 1	
END IF	
ELSE	
RETURN I	
E = A	
END IF	
END IF	
ELSE	
Q = 1	
END IF	
RETURN Q	

Sample Output

HINT

Source

[Submit][Status][Discuss]

HOME Back