

Problem D

Edsger Dijkstra

Time limit: 2 seconds

Memory limit: 256 megabytes

Problem Description

圖靈獎得主艾茲赫爾·韋伯·戴克斯特拉 (Edsger Wybe Dijkstra) 是荷蘭計算機科學家，他的名字的發音總是讓許多亞洲人感到困擾。在 1960 年代，他給出了以下的評論：

For a number of years I have been familiar with the observation that the quality of programmers is a decreasing function of the density of go to statements in the programs they produce. More recently I discovered why the use of the go to statement has such disastrous effects, and I became convinced that the go to statement should be abolished from all “higher level” programming languages.

你不想產出低品質的程式碼，對吧？然而你發現你的程式碼裡面一個迴圈敘述都沒有。唯一的流程控制程式碼，就只有if-goto。為了降低goto的密度，你得試著消除掉一份類 C 程式碼裡所有的if-goto 敘述，用do-while 迴圈以下面的方式取代：

- 假定if-goto 看起來像是 “if (boolean_expression) goto some_label;”。
- 在some_label 宣告處後方插入一份 “do {”。
- 用 ‘} while’ 取代 ‘if’。
- 移除 ‘goto some_label’。

舉例來說，就是將以下程式碼

```
int main() {
    int score;
    get_score:
    scanf("%d",&score);
    if (score < 0 || score > 100) goto get_score;
    if (score < 60) goto fail;
    fail:
    puts("you are failed!");
    return 0;
}
```

轉換為下面這樣：

```
int main() {
    int score;
    get_score: do {
        scanf("%d",&score);
    } while (score < 0 || score > 100) ;
    } while (score < 60) ;
    fail: do {
        puts("you are failed!");
    } while (1);
    return 0;
}
```

以上的程式碼無法通過編譯，並不是一個令人意外的結果。你的新任務是：給定一連串的敘述，請判斷是否所有的if-goto 敘述，可以用do-while 迴圈，以前述的方式取代，並且不影響到程式的輸出。為了簡單起見，你可以假定程式碼中的敘述只有下列兩種格式：

- 格式一：“line_x: puts("x");” 其中 x 是此敘述對應的行號且 “puts("x");” 會印出行號 x 。
- 格式二：“if (expr_x()) goto line_y;” 其中 x 是該敘述對應的行號，line_y 是程式中合法的標籤 (Label)。“expr_x()” 在前 x 次呼叫時，將會回傳 true，之後會回傳 false。

請留意如果程式碼無法通過編譯，將被視為與原先的程式碼輸出不同。

Input Format

輸入資料的第一行有一個整數 T ($T \leq 20$) 代表有多少組測試資料。每一組測試資料由一連串的敘述所構成，每行有一個敘述。而構成測試資料的敘述有三種：格式一、格式二以及END。END 代表該組測試資料結束，只會出現在每筆測試資料的最後一行，不用把它看作程式碼的一部分。單一筆測試資料，至多有 10000 個敘述 (不含END)。

Output Format

如果將所有的if-goto 敘述換成do-while 迴圈不會改變輸出的話，請印出 “good”，反之印出 “bad”。如果程式碼將無法編譯，請輸出 “bad”。

Sample Input

```
3
line_1: puts("1");
if (expr_2()) goto line_1;
END
line_1: puts("1");
line_2: puts("2");
line_3: puts("3");
if (expr_4()) goto line_2;
line_5: puts("5");
if (expr_6()) goto line_2;
END
line_1: puts("1");
if (expr_2()) goto line_5;
line_3: puts("3");
line_4: puts("4");
line_5: puts("5");
END
```

Sample Output

```
good
good
bad
```