ICPC2011 国内予選問題 B　世界の天秤

アルゴリズム

丸括弧と角括弧がバランスしていることのチェックは、スタックを用いると簡単に行える。文字列を左側から順に 1 文字ずつ取り出し、その文字を c とする。

- c が左丸括弧または左角括弧なら c をスタックに入れる。
- c が右丸括弧の時、スタックのトップが左丸括弧ならそれをスタックから取り出し、左丸括弧でなければ「バランスしていない」。
- c が右角括弧の時、スタックのトップが左角括弧ならそれをスタックから取り出し、左角括弧でなければ「バランスしていない」。
- c がピリオドの時、スタックが空なら「バランスしている」が、空でなければ「バランスしていない」。
- c が丸括弧や角括弧、ピリオド以外の時は、単に無視すれば良い。

※ スタックは文字型の配列で実現する。「一行の長さは 100 文字以下」なので、配列の大きさは 100 で良い。

※ 文字列を 1 行分読み込むには fgets 関数を用いる。fgets で読み込むと行末に改行文字と NULL 文字が付加されることに注意。

```
// fgets の使い方

#define MAX_LSIZE 100          // 一行の文字列の長さの最大値
char buff[MAX_LSIZE+2];        // 読み込んだ文字列を格納する配列
// ↑改行文字も読み込まれ、最後に NULL 文字が付加されるため
// 配列の大きさは MAX_LSIZE+2 としている

……
// 標準入力より 1 行文字列を読み込み配列 buff に格納する
// fgets の第 2 引数は配列 buff の大きさ
fgets(buff, MAX_LSIZE+2, stdin);
```

［プログラム例］

```
// ICPC 2011 国内予選問題 B
// http://icpc2011.ait.kyushu-u.ac.jp/icpc2011/contest/all_ja.html#section_B
//          Filename = pb.c
//          Compile:                cc pb.c
//          Execution:  ./a.out < B0 > B0.result
//          Check:                  diff B0.ans B0.result
//
// スタックを用いて、括弧の対応関係をチェックする
//
#include <stdio.h>
#define MAX_LSIZE 100                       // 一行の最大の長さ
char buff[MAX_LSIZE+2];                     // 一行の文字列を格納する配列
                                            // 行末の改行文字、NULL 文字も格納される
char stack[MAX_LSIZE];                      // スタック
int sp;                                     // スタックポインタ

int main()
{
   char c;              // 処理する文字を入れる変数
   int match;           // 括弧のバランス状況を表す
                        // 1 -> バランス,   0 -> 非バランス, -1 -> まだ不明
   int i;
   while(1){
      fgets(buff, MAX_LSIZE+2, stdin);          // 標準入力より配列 buff に1行読み込む
      if(buff[0] == '.') break;         // 先頭の文字が '.' なら終了
      match = -1;                       // バランス状況はまだ不明
      sp = 0;                           // stack pointer sp を初期化
      for(i=0; ; i++){      // 文字列の先頭から順に1文字ずつ以下の処理を行う
         c = buff[i];                   // c = 処理対象の文字
         switch(c){
         case '(':
         case '[':
              stack[sp++] = c; break;           // stack に 左括弧 を入れる
         case ')':          // stack が空か、
                            // stack から取り出した文字が '(' でなければバランスしていない
              if(sp==0) match=0;
              else if(stack[--sp] != '(') match = 0;
              break;
         case ']':          // stack が空か、
                            // stack から取り出した文字が '[' でなければバランスしていない
              if(sp==0) match=0;
              else if(stack[--sp] != '[') match = 0;
              break;
         case '.':          // stack が空ならバランス、そうでなければ非バランス
              if(sp==0) match=1;
              else match=0;
         }
         if(match != -1) break;         // 結果が判明したので終了
      }
```

```
        if(match) printf("yes¥n");      // 結果を出力
        else printf("no¥n");
    }
}
```