

ACM ICPC 2015 国内予選 [問題 B 短句](#)

【考察】

- 単語の長さだけ考慮すれば良いので、単語長を配列に格納しておく。
- 先頭の単語から始まる短句を構成出来るかをチェック。構成できる場合は先頭の単語位置を出力して終了。構成できない場合は、次の単語から始まる短句を構成できるかをチェック。このような操作を短句が構成できるまで繰り返す。（「必ず短句を含んでいる」ので、この手順は必ず短句が構成できることが判明して終了する）
- 複数の短句を構成可能な場合、最初のを出力する。
- 短句を構成できるかどうかのチェック：
 - 最初の区切り（長さ 5）に対し、先頭から順に単語長を引いていく。0 になれば、次の区切りをチェック。負になれば、短句を構成できない。
 - 次の区切り（長さ 7）に対しても、同様にチェック。
 - 同様に、長さ 5, 長さ 7, 長さ 7 の各区切りを順にチェックしていけば良い。

【プログラム例 pb.c】

```
// ACM-ICPC 2015 国内予選 Problem B 短句
// http://icpc.iisf.or.jp/past-icpc/domestic2015/contest/all_ja.html#section_B
// Filename:      pb.c
// Compile:       gcc pb.c
// Execution:     ./a.out < B0 > B0.result
// Check:        diff B0.ans B0.result
// Algorithm:     先頭単語を後ろにずらしながら短句を構成できるかをチェック
//               短句の構成可能性は、各区切り毎に単語長を引いていくことで
//               チェック。負になれば構成不可。0 になれば次の区切りをチェック
```

```
#include <stdio.h>
#include <string.h>
#define MAXN 40          // 単語数の最大値
#define LEN 10          // 単語長の最大値
int n;                  // 単語数
int len[MAXN];         // 単語の長さを覚える配列
char word[LEN+1];      // 単語を読み込む配列

// pos 番目以降の単語列で短句が構成出来るかどうかをチェック
int is_ok(int pos)
{
    int rest;
    rest = 5;           // 最初の区切りの長さは5
    while(rest > 0){
        rest -= len[pos++]; // 必要な文字数から単語の文字数を引く
        if(rest < 0) return 0; // 負になったらダメ
    }
    rest = 7;           // 2番目の区切りの長さは7
    while(rest > 0){
        rest -= len[pos++]; // 必要な文字数から単語の文字数を引く
        if(rest < 0) return 0; // 負になったらダメ
    }
    rest = 5;           // 3番目の区切りの長さは5
    while(rest > 0){
        rest -= len[pos++]; // 必要な文字数から単語の文字数を引く
        if(rest < 0) return 0; // 負になったらダメ
    }
    rest = 7;           // 4番目の区切りの長さは7
    while(rest > 0){
        rest -= len[pos++]; // 必要な文字数から単語の文字数を引く
        if(rest < 0) return 0; // 負になったらダメ
    }
    rest = 7;           // 5番目の区切りの長さは7
    while(rest > 0){
        rest -= len[pos++]; // 必要な文字数から単語の文字数を引く
        if(rest < 0) return 0; // 負になったらダメ
    }
}
```

```

    }
    return 1;        // 短句になっている
}

int main(void)
{
    for(;;){
        int i;
        scanf("%d¥n", &n);           // 単語数 n を入力
        if(n==0) break;             // n==0 なら終了

        for(i=0; i<n; i++){         // 単語を n 個読み込んでその長さを格納
            scanf("%s¥n", word);    // 単語を読み込む
            len[i] = strlen(word);  // その長さを格納
        }

        for(i=0; i<n; i++){         // 先頭の単語から順に
            if(is_ok(i)){           // その単語から始めて短句を構成できるかをチェック
                printf("%d¥n", i+1); // 構成出来る場合はその単語位置をプリント
                break;
            }
        }
    }
}

```