

次に示すポインタ版の「Boyer-Moore 法による文字列探索」プログラムを入力し、自分のパソコンでコンパイル、実行できることを確認してください。なお、プログラムの日本語部分は、英語、ローマ字に変更してかまいません。さらに、以下の問いに答えなさい。

```
#include <stdio.h>
#include <string.h>
#include <limits.h>
/*--- Boyer-Moore 法による文字列探索 ---*/
char *bm_match(char *pat , char *txt){
    char    *pt;                /* txt をなぞるカーソル */
    char    *pp;                /* pat をなぞるカーソル */
    int     txt_len = strlen(txt); /* txt の文字数 */
    int     pat_len = strlen(pat); /* pat の文字数 */
    int     skip[CHAR_MAX + 1]; /* スキップテーブル */
    int     i;

    for (i = 0; i <= CHAR_MAX; i++) /* スキップテーブルの作成 */
        skip[i] = pat_len;
    for (pp = pat; *pp != '\0'; pp++)
        skip[*pp] = strlen(pat) - 1;
    skip[*pp - 1] = pat_len; /* パターンの最後文字の移動距離はパターンの文字数 */

    pt = txt + pat_len - 1; /* pat の末尾と比較する txt の文字を決定 */
    while ( pt < txt + txt_len) { /* txt の比較する文字の位置が txt の末尾を越えるまで */
        pp = pat + pat_len - 1; /* pat の最後の文字に着目 */
        while (*pt == *pp) { ①
            if (pp == pat) return (pt); /* 一致した文字がパターンの最初の文字になれば終了 */
            pp--;
            pt--;
        }
        pt += (skip[*pt] > strlen(pat)) ? skip[*pt] : strlen(pat);
    }
    return (NULL);
}

int main(void){
    char    *s;
    char    s1[80]; /* テキスト */
    char    s2[80]; /* パターン */

    printf("テキスト:");
    scanf("%s", s1);
    printf("パターン:");
    scanf("%s", s2);

    s = bm_match(s2, s1); /* 文字列 s1 から文字列 s2 を Boyer-Moore 法で探索 */
    ②
    if (s == NULL)
        puts("テキスト中にパターンは存在しません。");
    else
        printf("%d 文字目に見つかりました。¥n", s - s1 + 1);
    return (0);
}
```

1) キーボードからテキストとして「CGAAGACGACAGG」を、パターンとして「AGG」を入力したとき、次の間に答えなさい。

(ア) スキップテーブルを表す配列の `skip['G']` と `skip['A']` の値はいくらですか。

(イ) 下線②の関数 `bm_match(s2, s1)` の戻り値のポインタが指している文字は、テキスト「AACAGACAGG」のどの文字ですか。

2) キーボードからテキストとして「ACCCCACCAAAACCC」を、パターンとして「AAC」を入力したとき、下線部①の `while` の条件式 (`*pt == *pp`) は何度評価されますか。

3) 関数 `bm_match` の Boyer-Moore 法のアルゴリズムを変更して、テキストから指定した文字列の並びと逆順のパターンを探索する関数 `char *bm_reverse_macth(char *pat, char *txt)` を作成してください。但し、関数 `bm_reverse_macth` の引数も戻り値も、関数 `bm_match` と同じとしますが、`bm_reverse_macth` では、`pat` が指す文字列には、探索したいパターンの文字の並びと逆順で文字が並んでいるものとします。さらに、`main` から呼び出される `bm_match` を `bm_reverse_macth` に変更してください。