

次の問 8 は必須問題です。必ず解答してください。

問 8 次のプログラムの説明及びプログラムを読んで、設問 1～3 に答えよ。

〔プログラムの説明〕

関数 BMMatch は、Boyer-Moore-Horspool 法（以下、BM 法という）を用いて、文字列検索を行うプログラムである。BM 法は、検索文字列の末尾の文字から先頭に向かって、検索対象の文字列（以下、対象文字列）と 1 文字ずつ順に比較していくことで照合を行う。比較した文字が一致せず、照合が失敗した際には、検索文字列中の文字の情報を利用して、次に照合を開始する対象文字列の位置を決定する。このようにして明らかに不一致となる照合を省き、高速に検索できる特徴がある。

(1) 対象文字列を Text[], 検索文字列を Pat[] とする。ここで、配列の添字は 1 から始まり、文字列 Text[] の i 番目の文字は Text[i] と表記される。Pat[] についても同様に i 番目の文字は Pat[i] と表記される。また、対象文字列と検索文字列は、英大文字から構成される。

例えば、対象文字列 Text[] が “ACBBMACABABC”，検索文字列 Pat[] が “ACAB” の場合の例を図 1 に示す。

	1	2	3	4	5	6	7	8	9	10	11	12
Text[]	A	C	B	B	M	A	C	A	B	A	B	C

	1	2	3	4
Pat[]	A	C	A	B

図 1 対象文字列と検索文字列の格納例

(2) 関数 BMMatch では、照合が失敗すると、次に照合を開始する位置まで検索文字列を移動するが、その移動量を格納した要素数 26 の配列 Skip[] をあらかじめ作成しておく。Skip[1] に文字 “A” に対応する移動量を、Skip[2] に文字 “B” に対応する移動量を格納する。このように、Skip[1] ～ Skip[26] に文字 “A” ～ “Z”

(4) 図1の例で照合する場合の手順は、次の①～⑨となり、その流れを図3に示す。

この例では、PatLen = 4 なので、検索文字列の末尾の文字は Pat[4] である。

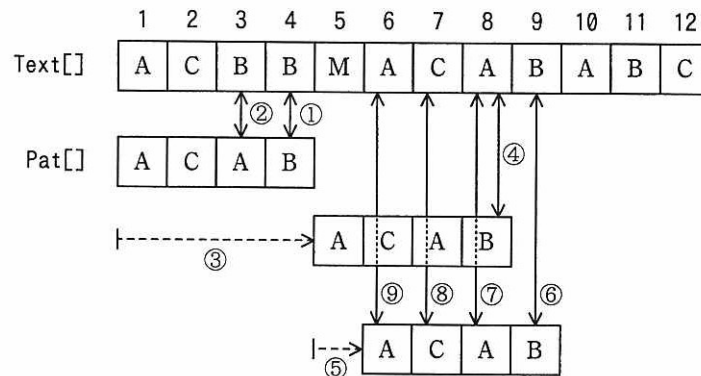


図3 図1の場合の照合手順

- ① Text[4]と Pat[4]を比較する。Text[4]と Pat[4]は同じ文字“B”である。
- ② Text[3]と Pat[3]を比較する。Text[3]の“B”と Pat[3]の“A”は異なる文字である。
- ③ ①で検索文字列の末尾の文字 Pat[4]と比較した Text[4]を基準に、Text[4]の文字“B”に対応する移動量である Skip[2]の値4だけ Pat[]を右側に移動し、Text[8]と Pat[4]の比較に移る。
- ④ Text[8]と Pat[4]を比較する。Text[8]の“A”と Pat[4]の“B”は異なる文字である。
- ⑤ ④で検索文字列の末尾の文字 Pat[4]と比較した Text[8]を基準に、Text[8]の文字“A”に対応する移動量である Skip[1]の値1だけ Pat[]を右側に移動し、Text[9]と Pat[4]の比較に移る。
- ⑥ Text[9]と Pat[4]を比較する。Text[9]と Pat[4]は同じ文字“B”である。
- ⑦ Text[8]と Pat[3]を比較する。Text[8]と Pat[3]は同じ文字“A”である。
- ⑧ Text[7]と Pat[2]を比較する。Text[7]と Pat[2]は同じ文字“C”である。
- ⑨ Text[6]と Pat[1]を比較する。Text[6]と Pat[1]は同じ文字“A”である。

⑥～⑨の比較で、対象文字列 Text[] の連続した一部分が検索文字列 Pat[] に完全に一致したので、検索は終了する。

[関数 BMMatch の引数と返却値]

関数 BMMatch の引数と返却値の仕様は、次のとおりである。

引数名／返却値	データ型	意味
Text[]	文字型	対象文字列が格納されている 1 次元配列
TextLen	整数型	対象文字列の長さ (1 以上)
Pat[]	文字型	検索文字列が格納されている 1 次元配列
PatLen	整数型	検索文字列の長さ (1 以上)
返却値	整数型	対象文字列中に検索文字列が見つかった場合は、1 以上の値を返す。 検索文字列が見つからなかった場合は、-1 を返す。

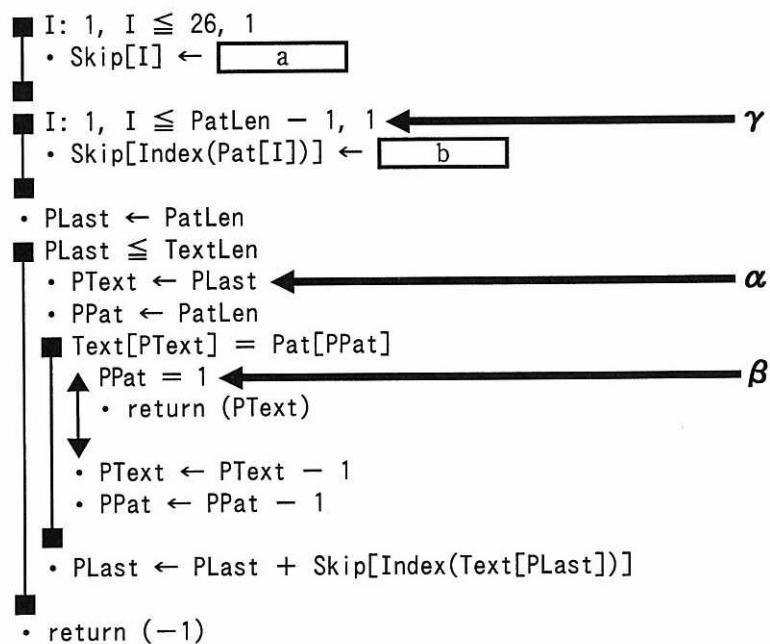
関数 BMMatch では、次の関数 Index を使用する。

[関数 Index の仕様]

引数にアルファベット順で n 番目の英大文字を与えると、整数 n ($1 \leq n \leq 26$) を返却値とする。

[プログラム]

○整数型関数: BMMatch(文字型: Text[], 整数型: TextLen,
文字型: Pat[], 整数型: PatLen)
○整数型: Skip[26], PText, PPat, PLast, I



E
後

設問1 プログラム中の に入れる正しい答えを、解答群の中から選べ。

a, b に関する解答群

- | | | |
|-------------------|-----------------------|-----------------------|
| ア 0 | イ 1 | ウ $I - \text{PatLen}$ |
| エ PatLen | オ $\text{PatLen} - 1$ | カ $\text{PatLen} - I$ |

設問2 次の記述中の に入れる正しい答えを、解答群の中から選べ。

図4のように、Text[] に “ABCXBBACABACADEC”，TextLen に 16，Pat[] に “ABAC”，PatLen に 4 を格納し、BMMatch (Text[], TextLen, Pat[], PatLen) を呼び出した。プログラムが終了するまでに α は c 回実行され、 β は d 回実行される。またこの場合、関数 BMMatch の返却値は e である。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Text[]	A	B	C	X	B	B	A	C	A	B	A	C	A	D	E	C

	1	2	3	4
Pat[]	A	B	A	C

図4 対象文字列と検索文字列

c～eに関する解答群

- | | | | | |
|-----|-----|------|------|------|
| ア 3 | イ 4 | ウ 5 | エ 6 | オ 7 |
| カ 8 | キ 9 | ク 10 | ケ 11 | コ 12 |

設問3 次の記述中の に入れる正しい答えを，解答群の中から選べ。ここで，プログラム中の a と b には正しい答えが入っているものとする。

関数 BMMatch 中の γ の処理を

$I: \text{PatLen} - 1, I \geq 1, -1$

に変更した場合，関数 BMMatch は f 。

f に関する解答群

- ア 対象文字列中に，検索文字列が含まれていないのに，1 以上の値を返す場合がある
- イ 対象文字列中に，検索文字列が含まれているのに，-1 を返す場合がある
- ウ 正しい値を返す