次の間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1,2に答えよ。

### 〔プログラムの説明〕

英字  $A \sim Z$  から構成される文字列を圧縮する副プログラム Compress, 及び圧縮された文字列を復元する副プログラム Decompress である。

#### [圧縮処理の説明]

副プログラム Compress では、配列 Plaindata に格納されている圧縮前の文字列を受け取り、圧縮後の文字列を配列 Compresseddata に格納する。図1に示す文字列 "ABCDEFABCDABCDEF"を例として、圧縮処理の内容を説明する。

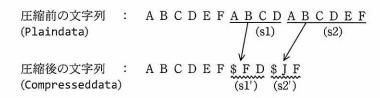


図1 文字列の圧縮例

- (1) 図1の圧縮前の文字列 "ABCDEFABCD …"のように、"ABCD"という同じ文字の並びが複数回出現する場合、二つ目以降の "ABCD"を圧縮列に置き換えることによって、文字列の長さを短くする。図1では、Plaindataに格納されている圧縮前の文字並び(s1)、(s2)を圧縮列(s1')、(s2')に置き換えてCompresseddataに格納している。
- (2) 圧縮列は、制御記号、距離、文字数の三つの部分から成る。
  - ① 制御記号は、"\$"であり、圧縮列の先頭を表す。
  - ② 距離と文字数は,圧縮前の文字列において,"この場所の文字並びは,何文字前(距離)の文字から始まる何文字の長さ(文字数)の文字並びと同じである" という内容を意味する。距離と文字数の最大値は 26 とし, 1, 2, …, 26 を A, B, …, Z で表す。図 1 の例では,圧縮前の文字並び(s1)は,(s1)の先頭文字の

6文字前から始まる長さ 4の文字並び "ABCD" と一致するので,圧縮列 (s1') は "\$FD" となる。また,圧縮前の文字並び (s2) は,(s2) の先頭文字の 10 文字前から始まる長さ 6 の文字並び "ABCDEF" と一致するので,圧縮列 (s2') は "\$JF" となる。

- (3) 圧縮処理は、配列 Plaindata に格納されている圧縮前の文字列中で、圧縮する 文字並びを検索する検索処理と、検出した文字並びを圧縮列に置き換えて配列 Compresseddata に格納する置換処理から成る。
- (4) 圧縮する文字並びの検索処理の内容を次に示す。
  - ① 圧縮処理の対象となる文字並びを圧縮文字並び、圧縮文字並びの先頭位置を圧縮文字位置という。圧縮列の文字数は3なので、一致する文字数が4以上の圧縮文字並びの場合に圧縮列に置き換える。したがって、最初の圧縮文字位置は、図2に示すように圧縮前の文字列の先頭から5文字目となる。この文字位置から圧縮文字位置を文字列の後方に向かって移動させながら検索処理②、③を行う。

#### 図 2 最初の圧縮文字位置

② 圧縮文字並びの比較対象とする文字並びを比較文字並び、比較文字並びの先頭 位置を比較文字位置という。最初の比較文字位置は、図3に示すように圧縮文字 位置の4文字前とする。



図3 最初の比較文字位置

- ③ 比較文字位置を文字列の先頭に向かって,圧縮文字位置から最大 26 文字前まで移動させながら,次の(a),(b)を行う。
  - (a) 圧縮文字位置と比較文字位置から文字列の後方に向かって, 圧縮文字並び と比較文字並びが何文字一致するかを調べる。

図 4 は,圧縮文字位置が 7 文字目のときに,比較文字位置を文字列の先頭に向かって  $p_1$ ,  $p_2$  ,  $p_3$  と移動しながら,一致する文字を検索している様子を示している。比較文字位置  $p_1$  と  $p_2$  では,1 文字目が一致していない。比較文字位置  $p_3$  では 1 文字目が一致し,一致する文字数は 4 となる。

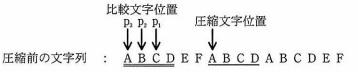


図 4 文字列の比較

(b) 圧縮する文字並びは、それよりも前にある比較文字並びの中で、一致する文字数が4以上で最も多い比較文字並びとする。圧縮する文字並びの対応例を図5に示す。図5では、圧縮文字位置から始まる圧縮文字並び"ABCD…"と、それよりも前方に出現する比較文字並びの中で、一致する文字数が4以上の比較文字並びは、4文字前の(s1)"ABCD"と10文字前の(s0)"ABCDE"である。したがって、一致する文字数が多い(s0)に対応させて、(s2)"ABCDEF"を圧縮列に置き換えることになる。

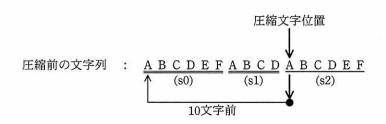


図5 圧縮する文字並びの対応例

- (5) (4)の検索処理の結果に対する置換処理の内容を次に示す。
  - ① 一致する文字数が3以下の場合には、圧縮列に置き換えないで、圧縮文字位置の文字を配列Compresseddataに格納する。
  - ② 一致する文字数が4以上の場合には、その文字数が最も多い比較文字並びに対応する圧縮列を配列Compresseddataに格納する。

## [副プログラム Compress の引数の仕様]

Compress の引数の仕様は、次のとおりである。各配列の添字は、0から始まる。

引数名	データ型	入力/出力	意味
Plaindata[]	文字型	入力	圧縮前の文字列が格納されている1次元配列
Plength	整数型	人力	圧縮前の文字列の長さ (1以上)
Compresseddata[]	文字型	出力	圧縮後の文字列が格納される1次元配列
Clength	整数型	出力	圧縮後の文字列の長さ

#### 〔復元処理の説明〕

副プログラム Decompress では、配列 Compresseddata に格納されている圧縮された文字列を受け取り、復元後の文字列を配列 Plaindata に格納する。復元処理の内容を次に示す。

- (1) 配列Compresseddataの先頭から圧縮された文字列を順に調べる。
- (2) 文字が制御記号でなければ、その文字をそのまま配列Plaindataに格納する。
- (3) 文字が制御記号ならば、圧縮列の距離、文字数から、圧縮前の文字並びを復元 して配列Plaindataに格納する。

#### 〔副プログラム Decompress の引数の仕様〕

Decompress の引数の仕様は、次のとおりである。各配列の添字は、0 から始まる。

引数名	データ型	入力/出力	意味
Compresseddata[]	文字型	入力	復元前の文字列が格納されている1次元配列
Clength	整数型	入力	復元前の文字列の長さ (1以上)
Plaindata[]	文字型	出力	復元後の文字列が格納される1次元配列
Plength	整数型	出力	復元後の文字列の長さ

副プログラム Compress と Decompress で使用している関数 IntToAlphabet と AlphabetToIntの仕様は、次のとおりである。

## 〔関数 IntToAlphabet の仕様〕

整数  $1\sim 26$  を順に英字  $A\sim Z$  に変換する。IntToAlphabet の引数と返却値の仕様は、次のとおりである。

引数/返却值	データ型	意味
引数	整数型	整数1~26の値
返却値	文字型	引数に対応した英字

## 〔関数 AlphabetToInt の仕様〕

英字  $A\sim Z$  を順に整数  $1\sim 26$  に変換する。AlphabetToInt の引数と返却値の仕様は、次のとおりである。

引数/返却値	データ型	意味	
引数	文字型	英字A~Zの文字	
返却值	整数型	引数に対応した整数の値	

### 〔プログラム1〕

○副プログラム: Compress(文字型: Plaindata[], 整数型: Plength,

文字型: Compresseddata[], 整数型: Clength)

○文字型: Esym

○整数型: Pindex, Cindex

○整数型: Maxfitnum, Maxdistance, Distance, Fitnum

```
▲ Plength ≧ 1
                     /* 制御記号の設定 */

    Esym ← "$"

                      /* 圧縮前の文字列の文字位置を初期化 */

    Pindex ← 0

                      /* 圧縮後の文字列の文字位置を初期化 */

    Cindex ← 0

 ■ (Pindex < Plength) and (Pindex < 4)</p>
   • Compresseddata[Cindex] ← Plaindata[Pindex]
   • Cindex \leftarrow Cindex + 1
    • Pindex ← Pindex + 1
 ■ Pindex < Plength</p>
   • Maxfitnum \leftarrow -1

    Maxdistance ← -1

    • Distance ← 4
   \blacksquare (Distance \leq 26) and ( a
     • Fitnum ← 0
     /* 同じ文字並びの文字数を調べる */
     ■ (Fitnum < Distance) and ((Pindex + Fitnum) < Plength)</p>
      /* 文字が一致するかどうかを調べる */
       ▲ Plaindata[Pindex + Fitnum] ≠ 
         • break /* 最も内側の繰返し処理を終了する */
       • Fitnum ← Fitnum + 1
     /* 文字数が4以上,かつ,最も多いかどうかを調べる */
     \blacktriangle (Fitnum \ge 4) and (Maxfitnum < Fitnum)
       • Maxfitnum ← Fitnum
       • Maxdistance ← Distance
    Maxfitnum = -1
      • Compresseddata[Cindex] ← Plaindata[Pindex]
      • Cindex ← Cindex + 1
      • Pindex ← Pindex + 1
     • Compresseddata[Cindex] ← Esym

    Compresseddata[Cindex + 1] ← IntToAlphabet(Maxdistance)

    Compresseddata[Cindex + 2] ← IntToAlphabet(Maxfitnum)

      • Cindex ← Cindex + 3
                   d
   Clength ← Cindex
```

```
○副プログラム: Decompress(文字型: Compresseddata[], 整数型: Clength,
                           文字型:Plaindata[],整数型:Plength)
    ○文字型: Esym
    ○整数型: Pindex, Cindex
    ○整数型: Num, Fitcnt, Start
     \triangle Clength \ge 1
       • Esym ← "$"
       • Cindex ← 0

    Pindex ← 0

      ■ Cindex < Clength</p>
        ♠ Compresseddata[Cindex] ≠ Esym

    Plaindata[Pindex] ← Compresseddata[Cindex];

         i \cdot Pindex \leftarrow Pindex + 1
         • Cindex ← Cindex + 1
          • Num ← AlphabetToInt(Compresseddata[Cindex + 2])
         • Start ← AlphabetToInt(Compresseddata[Cindex + 1])
          ■ Fitcnt: 0, Fitcnt < Num, 1
          Plaindata[Pindex + Fitcnt] ← 

    Pindex ← Pindex + Num

    Cindex ← Cindex + 3

    Plength ← Pindex

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に関する解答群
 ウ Plength - Distance ≧ 0 エ Plength - Pindex ≧ 0
bに関する解答群
 Plaindata[Pindex + Distance]

√ Plaindata[Pindex + Distance + Fitnum]

 ウ Plaindata[Pindex - Distance]
 エ Plaindata[Pindex — Distance + Fitnum]
```

[プログラム2]

#### cに関する解答群

- ウ Fitnum ← Fitnum + 1
- 工 Pindex ← Pindex + 1
- 才 Plength ← Plength + 1

# dに関する解答群

- $\mathcal{T}$  Pindex  $\leftarrow$  Pindex + 1
- ✓ Pindex ← Pindex + 3
- ウ Pindex ← Pindex + Maxdistance

#### eに関する解答群

- P Compresseddata[Pindex + Start + Fitcnt]
- ✓ Compresseddata[Pindex Start + Fitcnt]
- ウ Plaindata[Pindex + Start + Fitcnt]
- エ Plaindata[Pindex Start + Fitcnt]

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。

次の文字列を圧縮した文字列を副プログラム Decompress を使って復元する 場合, プログラム  $2 \circ \alpha$  の部分は f 回実行される。

文字列 : ABCDEFGABCDEABCDFEFGABCD

### fに関する解答群

ア 3

1 4

ウ 5

工 6

オ 7

力 8