

問題 4 次のプログラムの説明を読み、プログラム中の に入れるべき適切な字句を解答群から選べ。

[プログラムの説明]

英字で構成される文字列の圧縮・伸張をランレングス符号化で行うプログラムである。ランレングス符号化とはデータ圧縮に用いられる符号化の一種で、制御文字(\$), 連続する同じ値と、その個数の組み合わせで次のような形式で表現する。連続する同一文字のデータの個数は最大26とし、1, 2, …, 26を A, B, …, Zで表す。なお、連続する同一文字のデータが3文字以内の場合は、圧縮は行わない。

[制御文字(\$)] [データ] [個数] [制御文字(\$)] [データ] [個数] …

例 1	\$ a F \$ b J	データ a が 6 個連続し、データ b が 10 個連続する
例 2	\$ c E d d	データ c が 5 個連続し、データ d が 2 個連続する

ここで、圧縮前の文字列は255文字以内であり、圧縮は関数comp, 伸張は関数developで行う。

関数 comp では、配列 in_data 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 out_data に格納する。なお、in_len, out_len は、それぞれ配列 in_data と配列 out_data に格納されている文字列の長さである。

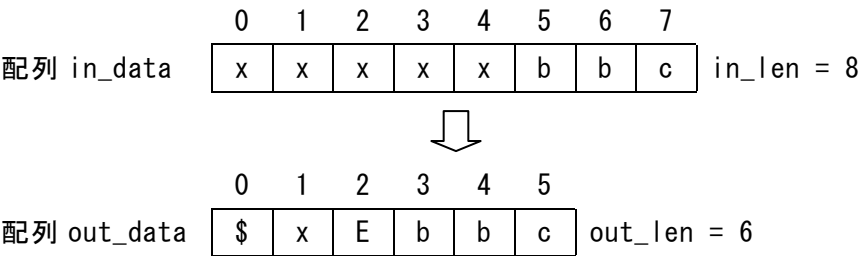


図 1 圧縮

関数 develop では、配列 out_data に格納されている圧縮されている文字列を受け取り、伸張後の文字列を配列 in_data に格納する。

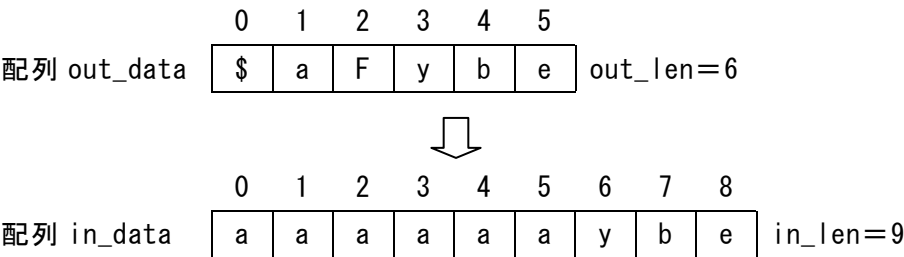


図 2 伸張

関数compと関数developで使用している関数toalphaと関数tointの仕様は次の通りである。

[関数 toalpha の仕様]

整数1, 2, ..., 26を順に英字 A, B, ..., Zに変換する。


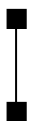
引数／返却値	データ型	意味
引数	整数型	整数1, 2, ..., 26の値
返却値	文字型	引数に対応した英字

[関数 toint の仕様]

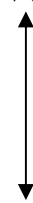
英字 A, B, ..., Zを順に整数1, 2, ..., 26に変換する。

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

[擬似言語の記述形式の説明]

記述形式	説明
○	手続き, 変数などの名前, 型などを宣言する
・変数 ← 式	変数に式の値を代入する
/* 文 */	注釈を記述する
 条件式 ・処理 1 ・処理 2	選択処理を示す。 条件式が真の時は処理 1 を実行し, 偽の時は処理 2 を実行する。
 条件式 ・処理	前判定繰り返し処理を示す。 条件式が真の間, 処理を実行する。

[演算子と優先順位]

演算の種類	演算子	優先順位
単項演算	+, −, not	高  低
乗除演算	*, /, %	
加減演算	+, −	
関係演算	>, <, ≥, ≤, =, ≠	
論理積	and	
論理和	or	

注記 整数同士の除算では, 整数の商を結果として返す。%演算子は剰余算を表す。

[プログラム]

○関数 **comp** (文字型配列 : **in_data[]**, **out_data[]**, 整数型 : **in_len**)

○整数型 : **cnt**, **i**, **j**, **k**, **m**, **out_len**;

○文字型 : **work**;

/* 文字列の圧縮 */

• **i** ← 0

• **k** ← 0

■ **k** < **in_len**

• **work** ← **in_data[k]**

• **cnt** ← 1

• **m** ← **k** + 1

■ (1)

• **cnt** ← **cnt** + 1

• **m** ← **m** + 1

■

▲ **cnt** ≥ 4

• **out_data[i]** ← '\$'

• **out_data[i+1]** ← **work**

• **out_data[i+2]** ← **toalpha(cnt)**

• **i** ← **i** + 3

—

• **j** ← 0

■ (2)

• **out_data[i]** ← **work**

• **i** ← **i** + 1

• **j** ← **j** + 1

■

▼

• (3)

■

• **out_len** ← **i**

• **return out_len**

(1) の解答群

ア. **m** < **in_len** and **work** = **in_data[m]**

イ. **m** ≤ **in_len** and **work** ≠ **in_data[m]**

ウ. **m** ≤ **in_len** or **work** = **in_data[m]**

(2) の解答群

ア. **j** < **cnt** - 1

イ. **j** < **cnt**

ウ. **j** ≥ **cnt**

(3) の解答群

ア. $k \leftarrow k + 1$ イ. $k \leftarrow m - 1$ ウ. $k \leftarrow m$

○関数 **develop** (文字型配列: **in_data[]**, **out_data[]**, 整数型: **out_len**)

○整数型: **cnt**, **i**, **j**, **k**, **m**, **in_len**;

○文字型: **work**;

/* 文字列の伸張 */

• $i \leftarrow 0$

• $k \leftarrow 0$

■ $k < \text{out_len}$

(4)

• $\text{cnt} \leftarrow \text{toint}(\text{out_data}[k + 2])$

• $j \leftarrow 0$

■ $j < \text{cnt}$

• $\text{in_data}[i] \leftarrow \text{out_data}[k + 1]$

• $i \leftarrow i + 1$

• $j \leftarrow j + 1$

■

• (5)

• $\text{in_data}[i] \leftarrow \text{out_data}[k]$

• $i \leftarrow i + 1$

• $k \leftarrow k + 1$

■

• $\text{in_len} \leftarrow i$

• **return in_len**

(4) の解答群

ア. $\text{out_data}[k] = '\$'$ イ. $\text{out_data}[k] \neq '\$'$

ウ. $\text{out_data}[k + 1] = '\$'$

(5) の解答群

ア. $k \leftarrow k + 1$ イ. $k \leftarrow k + 2$ ウ. $k \leftarrow k + 3$