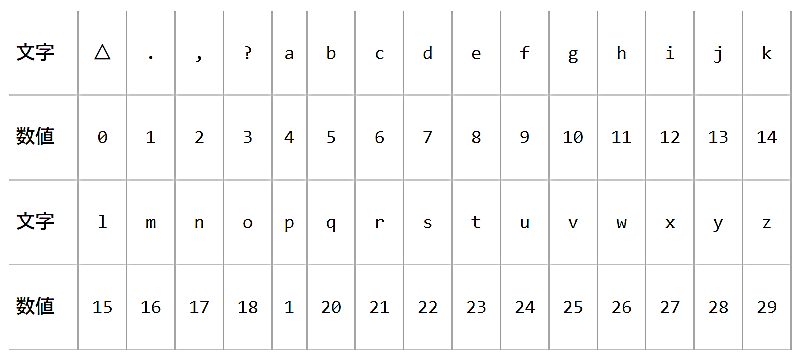
　問９：次の記述中( a 1 )・( a 2 )・( b )に入れる正しい答えを解答群の中から選べ。

　　関数 calcCheckCharacter は、引数 input[] で指定された文字列の誤りを検出するための検査文字を返す。文字列は、以下に示した N 種類 (ここでは N = 30 ）の文字から構成され、それぞれに数値が割り当てられている。空白文字は “△” と表記する。文字列の長さは、引数 len( 1 以上とする）で指定される。



**19**

　　［検査文字の生成の手順］

1. 文字列の末尾の文字を 1 番目の文字とし、文字列の先頭に向かって奇数番目の文字に割り当てた数値を 2 倍して N で割り、商と余りの和を求め、全て足し合わせる。
2. 偶数番目の文字に割り当てた数値は、そのまま全て足し合わせる。
3. ① と ②の結果を足し合わせる。
4. N から、 ③ で求めた総和を N で割った余りを引く。さらにその結果を、 N で割り、余りを求める。求めた数値に対応する文字を検査文字とする。

［検査文字の生成例］

文字列 “ipa△△” に対し、生成される検査文字は “f” である。

　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。ここで、 a1 と a2 に入れる答えは、 a に関する解答群の中から組合せとして正しいものを選ぶものとする。関数 caclCheckCharacter の中で使われている関数 getValue は、引数に与えられた文字に割り当てられた数値を返し、関数 getChar は、引数に与えられた数値に対応する文字を返す。配列の添字は、 1 から始まるとする。

［プログラム］

◯文字型: calcCheckCharacter(文字型の配列: input, 整数型: len)

整数型: N, sum, i, value, check\_value

論理型: is\_even

N ← 30

sum ← 0

is\_even ← a1

for (i を len から 0 より大きい間 -1 ずつ減らす )

value ← getValue(input[i])

if ( is\_even が a2 と一致)

sum ← sum + value

else

sum ← sum + (value × 2) ÷ N + (value × 2) mod N

endif

is\_even ← not is\_even

endfor

check\_value ← b

return getChar(check\_value)

( b )解答群

|  |  |
| --- | --- |
| ア | N - sum mod N |
| イ | sum mod N |
| ウ | ( N - sum mod N ) mod N |
| エ | ( sum - N ) mod Ne |

( a1 )( a2 )解答群

|  |  |  |
| --- | --- | --- |
|  | a1 | a2 |
| ア | false | false |
| イ | false | true |
| ウ | true | false |
| エ | true | true |