問16　次の記述中( a )・( b )・( c ) ・( d )に入れる正しい答えを解答群の中から選べ。　※基数変換

　　RadixConv 関数は、 M 進数字列（ 2 ≦ M ≦ 16 ）を、 N 進数字列（ 2 ≦ N ≦ 16 ）に基数変換するプログラムである。

1. M 進数字列および N 進数字列は、数字を表す文字だけで構成され、空白文字は含まない。 11 ～ 16 進数の 10 以上の数字には、 “A” ～ “F” のアルファベットを用いる。
2. RadixConv 関数は、 M 進数字列を整数値に変換した後、その整数値を N 進数字列に変換する。その際に、 M 進数字列を整数値に変換する MToInt 関数と、整数値を N 進数字列に変換する IntToN 関数を使う。
3. MToInt 関数と IntToN 関数は、数字を整数に変換して返す ToInt 関数、整数値を数字に変換する ToStr 関数、文字列の長さを返す Length 関数、および文字列の一部を切り出して返す SubStr 関数を使う。
4. このプログラムで使われる関数の引数と戻り値の仕様を表 1 ～ 7 に示す。

表 1　RadixConv 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| Frdx | 整数型 | 変換元の数字列の基数（ 2 ≦ Frdx ≦ 16 ） |
| Fnum | 文字列 | 変換元の数字列 |
| Trdx | 整数型 | 変換後の数字列の基数（ 2 ≦ Trdx ≦ 16 ） |
| 戻り値 | 文字列型 | 変換後の数字列 |

表 2　MToInt 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| Rdx | 整数型 | 変換元の数字列の基数（ 2 ≦ Rdx ≦ 16 ） |
| Num | 文字列型 | 変換元の数字列 |
| 戻り値 | 整数型 | 変換後の整数 |

表 3　IntToN 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| Val | 整数型 | 変換元の整数 |
| Rdx | 整数型 | 変換後の数字列の基数（ 2 ≦ Rdx ≦ 16 ） |
| 戻り値 | 文字列型 | 変換後の数字列 |

表 4　ToInt 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| P | 文字型 | 変換元の数字（ “0”, “1”, …, “F” ） |
| 戻り値 | 整数型 | 変換後の整数 |

表 5　ToStr 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| Q | 整数型 | 変換元の整数（ 0 ≦ Q ≦ 15 ） |
| 戻り値 | 文字型 | 変換後の数字 |

表 6　Length 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| S | 文字列型 | 長さを求める文字列 |
| 戻り値 | 整数型 | 文字列の長さ |

表 7　SubStr 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| S | 文字列型 | 切り出し元の文字列 |
| Idx | 整数型 | 切り出しを始める位置（先頭を 1 とする） |
| N | 整数型 | 切り出す文字数 |
| 戻り値 | 文字列型 | 文字列型S の先頭から Idx の位置から N 文字を返す |

　[プログラム]

　　◯文字列型: RadixConv(整数型: Frdx, 文字列型: Fnum, 整数型: Trdx)

return IntToN(MToInt(Frdx, Fnum), Trdx)

◯整数型: MToInt(整数型: Rdx, 文字列型: Num)

整数型: Idx, Val

Val ← 0

for (Idx を 1 から Length(Num) 以下まで 1 つずつ増加)

Val ← a + ToInt(Substr(Num, Idx, 1))

endfor

a に関する解答群

ア　Rdx

イ　Val

ウ　Val × Rdx

エ　Val ÷ Rdx

return Val

◯文字列型: IntToN(整数型: Val, 整数型: Rdx)

整数型: Quo /\* 商 \*/

整数型: Rem /\* 剰余 \*/

文字列型: Num

ｂ～ｄ に関する解答群

ア　Quo ← Quo ÷ Rdx

イ　Quo ← Quo ÷ Rem

ウ　Quo ← Rdx

エ　Quo ← Rem ÷ Rdx

オ　Quo ← Val

カ　Rem ← Rdx

キ　Rem ← Val

ク　Num ← ToStr(Quo) + Num

ケ　Num ← ToStr(Rem) + Num

b

Num ← ""

while (Quo が Rdx 以上)

Rem ← Quo mod Rdx

　　/\* + は文字列を連結する演算子 \*/

Num ← ToStr(Rem) + Num

c

endwhile

d

return Num