

別紙 1 擬似言語の記述形式（IT パスポート試験用）

アルゴリズムを表現するための擬似的なプログラム言語（擬似言語）を使用した問題では、各問題文中に注記がない限り、次の記述形式が適用されているものとする。

〔擬似言語の記述形式〕

| 記述形式                                                                                                                                                       | 説明                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ○ <u>手順名又は関数名</u>                                                                                                                                          | 手順又は関数を宣言する。                                                                                                                                                                                                                                                                                                            |
| <u>型名</u> : <u>変数名</u>                                                                                                                                     | 変数を宣言する。                                                                                                                                                                                                                                                                                                                |
| /* <u>注釈</u> */                                                                                                                                            | 注釈を記述する。                                                                                                                                                                                                                                                                                                                |
| // <u>注釈</u>                                                                                                                                               |                                                                                                                                                                                                                                                                                                                         |
| <u>変数名</u> ← <u>式</u>                                                                                                                                      | 変数に <u>式</u> の値を代入する。                                                                                                                                                                                                                                                                                                   |
| <u>手順名又は関数名</u> ( <u>引数</u> , …)                                                                                                                           | 手順又は関数を呼び出し、 <u>引数</u> を受け渡す。                                                                                                                                                                                                                                                                                           |
| if ( <u>条件式 1</u> )<br><u>処理 1</u><br>elseif ( <u>条件式 2</u> )<br><u>処理 2</u><br>elseif ( <u>条件式 n</u> )<br><u>処理 n</u><br>else<br><u>処理 n + 1</u><br>endif | 選択処理を示す。<br><br><u>条件式</u> を上から評価し、最初に真になった <u>条件式</u> に対応する <u>処理</u> を実行する。以降の <u>条件式</u> は評価せず、対応する <u>処理</u> も実行しない。どの <u>条件式</u> も真にならないときは、 <u>処理 n + 1</u> を実行する。<br><br>各 <u>処理</u> は、0 以上の文の集まりである。<br><br>elseif と <u>処理</u> の組みは、複数記述することがあり、省略することもある。<br><br>else と <u>処理 n + 1</u> の組みは一つだけ記述し、省略することもある。 |
| while ( <u>条件式</u> )<br><u>処理</u><br>endwhile                                                                                                              | 前判定繰返し処理を示す。<br><br><u>条件式</u> が真の間、 <u>処理</u> を繰返し実行する。<br><br><u>処理</u> は、0 以上の文の集まりである。                                                                                                                                                                                                                              |
| do<br><u>処理</u><br>while ( <u>条件式</u> )                                                                                                                    | 後判定繰返し処理を示す。<br><br><u>処理</u> を実行し、 <u>条件式</u> が真の間、 <u>処理</u> を繰返し実行する。<br><br><u>処理</u> は、0 以上の文の集まりである。                                                                                                                                                                                                              |
| for ( <u>制御記述</u> )<br><u>処理</u><br>endfor                                                                                                                 | 繰返し処理を示す。<br><br><u>制御記述</u> の内容に基づいて、 <u>処理</u> を繰返し実行する。<br><br><u>処理</u> は、0 以上の文の集まりである。                                                                                                                                                                                                                            |

〔演算子と優先順位〕

| 演算子の種類 |     | 演算子         | 優先度                                                 |
|--------|-----|-------------|-----------------------------------------------------|
| 式      |     | ()          | <div>高</div> <div>↑</div> <div>↓</div> <div>低</div> |
| 単項演算子  |     | not + -     |                                                     |
| 二項演算子  | 乗除  | mod × ÷     |                                                     |
|        | 加減  | + -         |                                                     |
|        | 関係  | ≠ ≤ ≥ < = > |                                                     |
|        | 論理積 | and         |                                                     |
|        | 論理和 | or          |                                                     |

注記 演算子 mod は、剰余算を表す。

〔論理型の定数〕

true, false

〔配列〕

一次元配列において“{”は配列の内容の始まりを，“}”は配列の内容の終わりを表し、配列の要素は，“[”と“]”の間にアクセス対象要素の要素番号を指定することでアクセスする。

例 要素番号が1から始まる配列 exampleArray の要素が{11, 12, 13, 14, 15}のとき、要素番号4の要素の値（14）は exampleArray[4]でアクセスできる。

二次元配列において、内側の“{”と“}”に囲まれた部分は、1行分の内容を表し、要素番号は、行番号、列番号の順に“,”で区切って指定する。

例 要素番号が1から始まる二次元配列 exampleArray の要素が{{11, 12, 13, 14, 15}, {21, 22, 23, 24, 25}}のとき、2行目5列目の要素の値（25）は、exampleArray[2, 5]でアクセスできる。