|  |
| --- |
| 疑似言語資料 |
| データ構造とアルゴリズム |
| 愛媛県　高度資格取得講習会 |

|  |
| --- |
| 学校法人　上野学園  広島コンピュータ専門学校 |

# １．擬似言語の表記法

問題解決する為の手順（アルゴリズム）を記号で表したものです。

記述形式は国家試験の問題に付いています。

|  |  |  |
| --- | --- | --- |
| 記述形式 | | 説明 |
| ○ | | 手続き、変数などの名前、型などを宣言する。 |
| ／＊　文　＊／ | | 文に注釈を記述する |
| 処　理 | ・変数　←　式 | 変数に式の値を代入する |
| ・手続（　引数　， …　） | 手続きを呼出し、引数を受け渡す。 |
| 条件式  　　　　処理 | 単岐選択処理を示す。  　　条件式が真のときは処理を実行する。 |
| 条件式  　　　　処理１  　　　　処理２ | 双岐選択処理を示す。  　　条件式が真のときは処理１を実行し、偽の時は処理２を実行する。 |
| 条件式  　　　　処理 | 前判定繰返し処理を示す。  　　条件式が真の間、処理を繰り返し実行する。 |
| 処理  　　　条件式 | 後判定繰返し処理を示す。  　　処理を実行し、条件式が真の間、処理を繰り返し実行する。 |
| 変数：初期値 ,　条件式　，　増分  　　　　処理 | 繰返し処理を示す。  　　開始時点で変数に初期値（式で与えられる）が格納され、条件式が真の間、処理を繰り返す。また、繰り返すごとに、変数に増分（式で与えられる）を加える。 |

|  |  |  |
| --- | --- | --- |
| 演算子の種類 | 演算子 | 優先順位 |
| 単項演算 | ＋，－，ｎｏｔ | 高い |
| 乗除演算 | ×，÷，％ |  |
| 加減演算 | ＋，－ |  |
| 関係演算 | ＞，＜，≧，≦，＝，≠ |  |
| 論理積 | ａｎｄ |  |
| 論理和 | ｏｒ | 低い |

# ２．アルゴリズムの基本構造

アルゴリズムは「順次」・「選択」・「繰り返し」構造で表現をする。

## （１）　選択構造

問題１　変数ａに１０、変数ｂに１５が代入されている場合、真か偽かを答えなさい。

　ａ　＞　ｂ【ア　　　　　】　　ａ　≦　ｂ【イ　　　　　】　　ａ　≠　ｂ【ウ　　　　　】

問題２　変数ａに２０、変数ｂに２０を代入している場合、真か偽かを答えなさい。

　ａ　＞　ｂ【ア　　　　　】　　ａ　≦　ｂ【イ　　　　　】　　ａ　≠　ｂ【ウ　　　　　】

問題３　変数ｎの値が20以上であれば、変数cupに’B’、そうでなければ’J’を代入。

|  |  |
| --- | --- |
| ○関数：分岐の練習( 整数型：n )  ○文字型：cup  ▲【ア　　　　　　　】  　　・cup ← ‘ B ’  　　・cup ← ‘ J ’  ▼  ・出力(“cup=” ＋ cup ) | ≪実行例(ｎが21の場合)≫  cup=B |

問題４　変数ｎの値が20以上であれば、変数cupに’B’、そうでない場合、変数hが140未満であれば、変数cupに’M’、そうでなければ’J’を代入

|  |  |
| --- | --- |
| ○関数：複数分岐の練習( 整数型：n , h )  ○文字型：cup  ▲【ア　　　　　　　　】  　　・cup ← ‘ B ’  　　▲ ｈ ≧ 140  　　　　・cup ← ‘【イ　　　】’    　　　　・cup ← ‘【ウ　　　】’  　　▼  ▼  ・出力(“cup=” ＋ cup ) | ≪実行例(nが18,hが130)≫  cup=M |

問題５　明日が日曜日(‘S’)で晴れ(‘F’)たらピクニックに行き、それ以外家から出ない。

|  |  |
| --- | --- |
| ○関数：ピクニック  ○文字列型：day,weather  ・入力( day ) ／＊曜日＊／  ・入力( weather ) ／＊天気＊／  ▲【ア　　　　　　　　　】  　　▲【イ　　　　　　　　　】  　　　　・ピクニックに行く    　　　　・家から出ない  　　▼  　　・家から出ない  ▼ | ○関数：ピクニック  ○文字列型：day , weather  ・入力( day ) ／＊曜日＊／  ・天気( weather ) ／＊天気＊／  ▲【ウ　　　　　　　　　　　　　　　　　　　　】  　　・ピクニックに行く  　　・家から出ない  ▼ |

##### 問題６　ａが１０以上かｂが１０以上であればａとｂに１を加算し、そうでなければ何もしない。

|  |
| --- |
| ○関数：複合条件判定の練習  ○整数型：a , b  ・a ← 10  ・b ← 5  ▲ a ≧ 10 【ア　　　　】 b ≧ 10  　　・a ← a ＋ 1  　　・b ← b ＋ 1  ▼ |

## （２）　繰り返し

問題１　ｃｎｔが５未満であれば繰り返す。また、最終的なｃｎｔの値は？

|  |  |
| --- | --- |
| ○関数：5回繰り返す  ○整数型：cnt  ・cnt ← 0  ■ cnt ＜ 【ア　　　】  　　・cnt ← cnt ＋ 1  ■ | ｃｎｔ【イ　　　　】 |

問題２　ｃｎｔが５未満であれば繰り返す。また、最終的なｃｎｔの値は？

|  |  |
| --- | --- |
| ○関数：5回繰り返す  ○整数型：cnt  ・cnt ← 0  ■ cnt ≦ 【ア　　　】  　　・cnt ← cnt ＋ 1  ■ | ｃｎｔ【イ　　　　】 |

問題３　比較演算に使用される数値は繰り返す回数にする場合、初期値によって比較演算子が変わります。１０回繰り返すとして【　　　】に入る比較演算子を考えてみましょう。

**カウンタが　０　から始まる場合　　ｃｎｔ　【ア**　**】　１０**

**カウンタが　１　から始まる場合　　ｃｎｔ　【イ**　**】　１０**

##### 問題４　「処理」の部分を１０回繰り返す繰り返しの判定を埋めましょう。また、αとβの行は何度通るでしょうか？

|  |  |
| --- | --- |
| 〇関数：10回繰り返す  〇整数型：cnt  ・cnt ← 0  ■ cnt 【ア　　　】 10 　←　α  ・処理  　　・cnt ← cnt ＋ 1 ←　β  ■ | 〇関数：10回繰り返す  〇整数型：cnt  ・cnt ← 1  ■ cnt 【イ　　　】 10  　　・処理  　　・cnt ← cnt ＋ 1  ■ |

　α【ウ　　　　】回　　，　β【エ　　　　】回

問題５　１からＮの整数のうち、偶数を出力する。

|  |  |
| --- | --- |
| ○関数：偶数を出力する( 整数型：N )  ○整数型： i  ・ i ← 1  ■ i ＜ 【ア　　　　　　　】  ▲【イ　　　　　　　】 ＝ 0  　　　　・出力( i )  　　▼  　・出力(改行)  ・ i ← i ＋ 1  ■ | ≪実行例(Nが6の場合)≫  2  4  6 |

問題６　５段の直角三角形を表示する（トレース表）

０行目の時は※、１行目の時は※※・・・

|  |  |
| --- | --- |
| 〇関数：５段の直角三角形を表示  〇整数型：i , j  　・i ← 0  ■【ア　　　　　　　　】  　　・j ← 0  　　■【イ　　　　　　　　】  　　　　・出力( ‘＊’ )  　　　　・j ← j ＋ 1  　　■  　　・出力( 改行 )  　　・i ← i ＋ 1  ■ | ≪実行例≫  ＊  ＊＊  ＊＊＊  ＊＊＊＊  ＊＊＊＊＊ |

問題７　ｃｎｔを使い　「処理」　をＮ回繰り返す

|  |
| --- |
| ○関数：処理をＮ回繰り返す（ 整数型：N ）  ○整数型：cnt  ■cnt：0 ，【ア　　　　　　　　】，【イ　　　】  　　・処理  ■ |

# ３．配列

## （１）　１次元配列

〔配列の宣言〕

|  |
| --- |
| ○整数型：ary[ 要素数 ] |

要素数５の場合

[ 0 ]

[ 1 ]

[ 2 ]

[ 3 ]

[ 4 ]

**ary**

|  |
| --- |
| 部屋の番号を「添え字」という。添え字でそれぞれの要素を識別する。  添え字は０から始まる。  配列の最後の要素は、「添え字」が「要素数―１」となる。 |

問題１　ａｒｙ［０］へ０を代入

|  |
| --- |
| ・【ア　ａｒｙ［０］　】 ← 0 |

問題２　ａｒｙ［０］より値を取り出し、変数ａに代入する。

|  |
| --- |
| ・a ← 【イ　ａｒｙ［０］　】 |

問題３　配列の要素をすべて「０」にしなさい。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ○関数：配列を初期化する  ○整数型：ary[ 5 ]，cnt  ■cnt：0，【ア　　　　　　　】，【イ　　　】  　・【ウ　　　　　　　　　　　　】 ← 0  ■ | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  | |

##### 問題４　５つの要素を持つ１次元配列Ｔに１～５までの連番を代入する。配列は０から始まるものとする。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ○関数：配列に値を入れる繰り返し  ○整数型：T[ 5 ]，cnt  ■cnt：0，【ア　　　　　　　】，【イ　　　】  　・T[ cnt ]← 【ウ　　　　　　　　　】  ■ | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  | |

## （２）　２次元配列

〔宣言〕

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ○整数型：ary[行数][列数] | | | | | | |
|  | [ 0 ] | [ 1 ] | [ 2 ] | [ 3 ] | [ 4 ] |
| [ 0 ] |  |  |  |  |  |
| [ 1 ] |  |  |  |  |  |
| [ 2 ] |  |  |  |  |  |

問題１　２次元配列（Ｔ［３］［４］）を０で初期化する。添え字は０から始まるものとする。

|  |
| --- |
| ○関数：配列の初期化  ○整数型：T[ 3 ][ 4 ], i , j  ・i ← 0  ■i ＜ 3  　　・【ア　　　　　　　　　】  　　■j ＜ 4  　　　　・【イ　　　　　】 ← 0  　　　　・j ← j ＋ 1  　　■  　　・i ← i ＋ 1  ■ |

# ３．基本アルゴリズム

## （１）　探索アルゴリズム

配列中に目的の値が存在するかを探索するアルゴリズムです。

### ⅰ）　線形探索

配列の先頭から順番に探索値が存在するかを比較するアルゴリズムです。

問題１　Ｔ配列には値が代入されており同じ値は２つと代入されていない。Ｎには配列の要素数が代入されている。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | [0] | [1] | [2] | [3] | [4] |  |  |
| T | 50 | 32 | 68 | 93 | 24 | N | 5 |

Mに入力した値がTに存在すればその添え字を出力し、存在しなければ-1を表示する

|  |  |
| --- | --- |
| ○関数：線形探索（ 整数型：T[ 5 ]，N ）  ○整数型：M ， i， j  ・出力（ “値を入力してください：” ）  ・入力（ M ）  ・i ← 0  ・j ← 【ア　　　】  ■ i ＜ N  　　▲ T[ i ] ＝ M  　　　　・j ← 【イ　　　】  　　▼  　　・i ← i + 1  ■  ・出力（ j ） | ≪実行例１≫  値を入力してください：68  2  ≪実行例２≫  値を入力してください：15  -1 |

### ⅱ）　２分探索

配列内の値が、整列されていなければ使用できません。配列要素を半分にしていきながら目的の要素を見つけます。

手順）

1. ちょうど中央の添え字の番号を調べ、その要素の値（以降中央値　小数点以下切り捨て）と探索値を比較する。

②中央値と探索値を比べ探索値のほうが小さければ、探索値は[0]～[中央値-1]に入っていることになり、中央値よりも大きければ、探索値は[中央値+1]～[要素数-1]に入っていることになる。

③見つかるまで①②を繰り返す。

問題１　配列Ｔから入力値Ｘの「２７」を二分探索で見つけなさい。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |  |  |  |  |
| T | 12 | 18 | 25 | 27 | 31 | 35 | 48 | 49 | 51 | 55 | N | 10 | X | 27 |

■探索値：２７

・初期値　　最小値：０　　最大値：９　　中央値：【ア　　　】

・１回目終了時　　最小値：【イ　　　】　　最大値：【ウ　　　】　中央値：【エ　　　】

・２回目終了時　　最小値：【オ　　　】　　最大値：【カ　　　】　中央値：【キ　　　】

・３回目終了時　　最小値：【ク　　　】　　最大値：【ケ　　　】　中央値：【コ　　　】

問題２　配列Ｔから入力値Ｘの「５４」を二分探索で見つけなさい。

■探索値：５４

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |  |  |  |  |
| T | 12 | 18 | 25 | 27 | 31 | 35 | 48 | 49 | 51 | 55 | N | 10 | X | 54 |

・初期値　　最小値：０　　最大値：９　　　中央値：４

・１回目終了時　　最小値：【ア　　　】　　最大値：【イ　　　】　　中央値：【ウ　　　】

・２回目終了時　　最小値：【エ　　　】　　最大値：【オ　　　】　　中央値：【カ　　　】

・３回目終了時　　最小値：【キ　　　】　　最大値：【ク　　　】　　中央値：【ケ　　　】

・４回目終了時　　最小値：【コ　　　】　　最大値：【サ　　　】　　中央値：【シ　　　】

問題３　２分探索のアルゴリズムを完成させなさい。

|  |  |
| --- | --- |
| ○関数：２分探索（整数型：T[ ]，整数型：N ）  ○整数型：X，Min，Max，Mid，  ・出力（“探索値を入力：”）  ・入力（X）  ・Min ← 0  ・Max ← N － 1  ・Mid ← （Min ＋ Max ）÷ 2  ■T[ Mid ] ≠ X and 【ア　　　　】  　　▲T[ Mid ] ＜ X  　　　　・Min ← 【イ　　　　　　　　】    　　　　・Max ← 【ウ　　　　　　　　】  　　▼  　　Mid ← （ Min ＋ Max ）÷ 2  ■  ▲ Min ＞ Max  　　・出力（－1）  　　・出力（ Mid ）  ▼ | ＜実行例1＞  探索値を入力：27  3  ＜実行例2＞  探索値を入力：50  —1 |

## （２）　ソートアルゴリズム

値を並べ替えるアルゴリズムです。

### ⅰ）　基本交換法

隣り合う要素の比較交換を繰り返すことで整列していく方法です。

問題１　配列の値を基本交換法で昇順に並べかえなさい。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | [0] | [1] | [2] | [3] | [4] | | T | 54 | 32 | 49 | 63 | 21 | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | ／＊T[]にはすでに値が代入されている＊／  ／＊nにはTの要素数が代入されている＊／  ○基本交換法（整数型：T[]，n）  ○整数型：i，j，temp  ■i ： n － 1 ， i ＞ 0 －1  　　■ j ： 0，j ＜ i，1  　　　　▲T[ j ] ＞ T[ j + 1 ]  　　　　　　・temp ← T[ j ]  　　　　　　・T[ j ] ← T[ j + 1 ]  　　　　　　・T[ j + 1 ] ← temp  　　　　▼  　　■  ■ |

### ⅱ）　基本選択法

未整列の左端から残りの要素を比較し、交換対象の要素が見つかったら交換をしていく方法です。

問題１　配列の値を基本選択法で昇順に並べかえ、さらに擬似言語を完成させなさい。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | [0] | [1] | [2] | [3] | [4] | | T | 54 | 32 | 49 | 63 | 21 | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | ／＊T[]にはすでに値が代入されている＊／  ／＊nにはTの要素数が代入されている＊／  ○関数：選択ソート昇順（整数型：T[]，n）  ○整数型：i，j，temp  ■i：0，i ＜【ア　　　　　】，1  　　■j：【イ　　　　　】，j ＜ n ，1  　　　　▲T[i]【ウ　　　　　】T[j]  　　　　　　・temp ← T[ i ]  　　　　　　・T[ i ] ← T[ j ]  　　　　　　・T[ j ] ← temp  　　　　▼  　　■  ■ |

### ⅲ）　基本挿入法

配列の要素を１つずつ整列済みに確定していく方法です。確定していない要素を正しい位置に挿入しながら並べ替えます。

問題１　配列の値を基本挿入法で昇順に並べかえ、さらに擬似言語を完成させなさい。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | [0] | [1] | [2] | [3] | [4] | | T | 54 | 32 | 49 | 63 | 21 | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | T |  |  |  |  |  | |  |  |  |  |  |  | | ／＊T[]にはすでに値が代入されている＊／  ／＊nにはTの要素数が代入されている＊／  ○関数：基本挿入法（整数型：T[]，n）  ○整数型：i，j，temp  ■i ： 1，i ＜ n ，1  ■j ： i，j ＞ 0　and【ア　　　　　　　　　】，－1  　　　　・temp ← T[j－1]  　　　　・T[ j － 1 ]← T [ j ]  　　　　・T[ j ] ← temp  　　■  ■ |

# 解答

# ２．アルゴリズムの基本構造

## （１）　選択構造

問題１　【ア　偽】　【イ　真】　【ウ　真】　　問題２　【ア　偽】　【イ　真】　【ウ　偽】

問題３　【ア　ｎ　≧　２０】　　問題４　【ア　ｎ　≧　２０】　【イ　ｊ】　【ウ　Ｍ】

問題５　【ア　ｄａｙ＝‘Ｓ’　】　【イ　ｗｅａｔｈｅｒ＝‘F’　】

　　　　【ウ　ｄａｙ＝‘Ｓ’　ａｎｄ　ｗｅａｔｈｅｒ＝‘F’　】

問題６　【ア　ｏｒ　】

## （２）　繰り返し

問題１【ア　５　】【イ　５　】　問題２【ア　４　】【イ　５　】　問題３　【ア　＜　】　【イ　≦　】

問題４　【ア　＜　】　【イ　≦　】　【ウ　１１　】　【エ　１０　】

問題５　【ア　Ｎ　＋　１】　【イ　ｉ　％　２　】　問題６　【ア　ｉ　＜　５　】【イ　ｊ　≦　ｉ　】

問題７　【ア　ｃｎｔ＜Ｎ　】【イ　１　】

# ３．配列

## （１）　１次元配列

問題１　【ア　ａｒｙ［０］　】　　問題２　【イ　ａｒｙ［０］　】

問題３　【ア　ｃｎｔ＜５　】【イ　１　】【ウ　ａｒｙ［ｃｎｔ］】

問題４　【ア　ｃｎｔ＜５　】【イ　１　】【ウ　ｃｎｔ＋１　】

## （２）　２次元配列

問題１　【ア　ｊ←０　】【イ　Ｔ［ｉ］［ｊ］】

# ３．基本アルゴリズム

## （１）　探索アルゴリズム

### ⅰ）線形探索

問題１　【ア　－１　】　【イ　ｉ　】

### ２）２分探索

問題１　【ア　４　】　【イ　０　】　【ウ　３　】　【エ　１　】　【オ　２　】　【カ　３　】

【キ　２　】　【ク　３　】　【ケ　３　】　【コ　３　】

問題２　【ア　５　】　【イ　９　】　【ウ　７　】　【エ　８　】　【オ　９　】　【カ　８　】

　　　　【キ　９　】　【ク　９　】　【ケ　９　】　【コ　９　】　【サ　８　】　【シ　８　】

問題３　【ア　Ｍｉｎ≦Ｍａｘ　】　【イ　Ｍｉｄ　＋　１　】　【ウ　Ｍｉｄ　－　１　】

## （２）　ソートアルゴリズム

### ⅱ）基本選択法

問題１　【ア　ｎ－１　】【イ　ｉ＋１　】【ウ　＞　】

### ⅲ）基本挿入法

問題２　【ア　Ｔ［ｊ］　＜　Ｔ［ｊ－１］　】