


# 海洋情報技術〔プログラミング〕

## 指導書


### 令和7年度大学入学共通テスト『情報Ⅰ』問3を解説

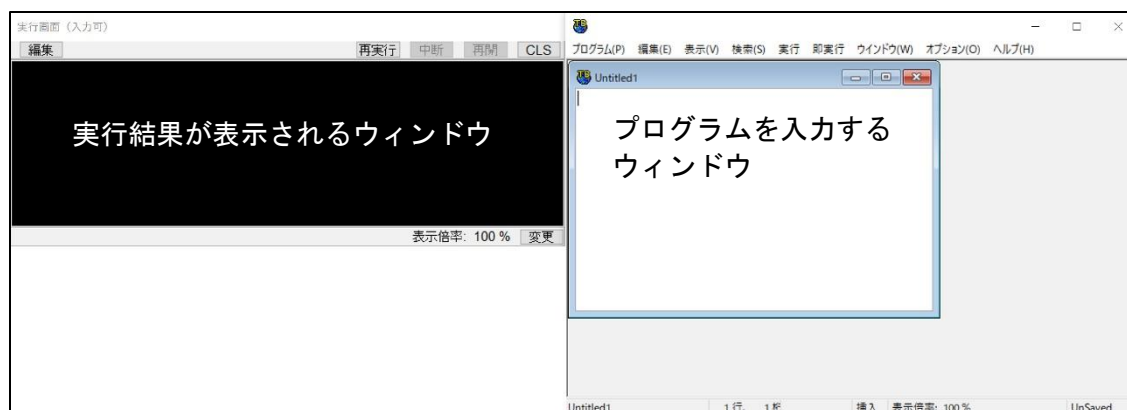
TinyBasic ソフトウェアのダウンロード先

<https://tbasic.org/>

ファイルをダウンロードし、展開したら、実行ファイル  TBasic を生徒の実習用フォルダや USB メモリにコピーして使用する。

## ソフトウェアの起動

 TBasic をダブルクリックすると、ソフトウェアが起動する。



## 大学入試用プログラミング言語

高校で学習するプログラミングは、プログラム言語の習得が目的ではなく、プログラマ的思考力やアルゴリズムの理解が目的となります。このため、各学校で用いられるプログラム言語は、Python や、VBA、C 言語、BASIC など様々なものとなります。しかし、特定の言語に特化した出題をすると、他の言語でプログラミングを習った人が不利となるため、大学入試では、共通テスト用プログラム表記である、**疑似コード**を用います。このプログラム表記は、何か特定のプログラミング言語を習ったことがあれば、容易に習得できる表記法になっているため、どの言語でプログラミングを学んでも十分理解することができます。

# 海洋情報技術〔プログラミング〕

## プログラミング問題 第3問

### 情報 I

第3問 次の文章を読み、後の問い(問1～3)に答えよ。(配点 25)

Kさんが所属する工芸部では毎年、文化祭に向けた集中製作合宿を開催し、複数の工芸品を部員全員で分担して製作している。Kさんは今年、工芸品を製作する担当の割当て作業を行うことになった。

問1 次の文章を読み、空欄 **ア** ～ **オ** に当てはまる数字をマークせよ。

表1は今年製作する各工芸品(1から順に番号を振る。)の製作日数である。製作日数は部員によって変わることはなく、例えば工芸品1の製作日数はどの部員が製作しても4日である。なお、一つの工芸品の製作は一人の部員が担当し、完了するまでその部員は他の工芸品の製作には取り掛からない。

表1 各工芸品の製作日数

工芸品	1	2	3	4	5	6	7	8	9
製作日数	4	1	3	1	3	4	2	4	3

Kさんは図1の割当図を作成し、今年の工芸部の部員3名について、工芸品の番号順に割当てを決めていくことにした。

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		1									
部員2	2		4								
部員3	3										

図1 割当図(工芸品4まで)

図1では、最上段に日付を合宿初日から順に1日目、2日目、…と表して記載している。その下に各部員(1から順に番号を振る。)に割り当てた工芸品の番号を、その製作期間を表す矢印とともに記載している。例えば、工芸品4は部員 **ア** が **イ** 日目から1日間製作することが、図1から読み取れる。

## 解説

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員 1											
部員 2		2	4								
部員 3			3								

工芸品 4 は○の部分であるため、部員 2 が 2 日目から製作することとなる。

図 1 では工芸品 4 までが割り当てられており、部員 1 が 5 日目で割り当てがない。このことを、部員 1 は 5 日目で空きであるという。

K さんは各工芸品の担当と期間を割り当てていく際、次の規則を用いた。

最も早く空きになる部員(複数いる場合はそのうち最小の番号の部員)が、空きになった日付から次の工芸品を担当する。

K さんは、工芸品 5 以降についても上の規則を用いて割り当て、各工芸品の担当と期間を一覧にした図 2 のような文面のメールを部員全員に送信した。

工芸品 1 … 部員 1 : 1 日目 ~ 4 日目  
 工芸品 2 … 部員 2 : 1 日目 ~ 1 日目  
 工芸品 3 … 部員 3 : 1 日目 ~ 3 日目  
 工芸品 4 … 部員 ア : イ 日目 ~ イ 日目  
 工芸品 5 … 部員 ウ : エ 日目 ~ オ 日目

工芸品 9 … 部員 1 : 7 日目 ~ 9 日目

図 2 各工芸品の担当と期間を一覧にしたメールの文面

以上を手作業で作成するのが手間だと感じた K さんは、図 2 のような文面を自動的に表示するプログラムを作成しようと考えた。

## 解説

工芸品 1 … 部員 1 : 1 日目 ~ 4 日目

工芸品 2 … 部員 2 : 1 日目 ~ 1 日目

工芸品 3 … 部員 3 : 1 日目 ~ 3 日目

工芸品 4 … 部員 **ア** : **イ** 日目 ~ **イ** 日目

工芸品 5 … 部員 **ウ** : **エ** 日目 ~ **オ** 日目

工芸品 9 … 部員 1 : 7 日目 ~ 9 日目

最も早く空きになる部員から割り当てるため、

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員 1			1								
部員 2	2	4			5						
部員 3		3									

となる。

よって、工芸品 5 は、部員 **2** : **3** 日目 ~ **5** 日目、となる。

問 2 次の文章を読み、空欄 **カ**、**ク** に当てはまる数字をマークせよ。また、空欄 **キ** に入れるのに最も適当なものを、後の解答群のうちから一つ選べ。

Kさんはまず、次の規則(再掲)に従い、いくつかの工芸品がすでに割り当てられた状況で、その次の工芸品の担当部員を表示するプログラムを作ることにした。

最も早く空きになる部員(複数いる場合はそのうち最小の番号の部員)が、空きになった日付から次の工芸品を担当する。

最も早く空きになる部員の番号を求めるために、各部員が空きになる日付を管理する配列 **Akibi** を用意する。この配列の添字(1から始まる。)は部員の番号であり、要素はその部員が空きになる日付である。

例えば、図1の状況では、配列 **Akibi** は図3のようになる。図1で部員1は5日目に空きになるため、図3で要素 **Akibi**[1]は5となる。同様に要素 **Akibi**[3]は **カ** となる。

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員 1			1								
部員 2	2	4									
部員 3		3									

図1 割当図(工芸品4まで)(再掲)

添字	1	2	3
<b>Akibi</b>	5	3	<b>カ</b>

図3 図1の状況に対応する配列 **Akibi**

図3において、要素 **Akibi**[**ウ**]が配列 **Akibi** の最小の要素であることから、部員 **ウ** が最も早く空きになることがわかる。

## 解説

配列 **Akibi** を用意し、各部員の空きになる日付を代入しておく。

```
Akibi[ 1 ]  5  
Akibi[ 2 ]  3  
Akibi[ 3 ]  力
```

部員番号 空き日

よって、部員番号3が空き日となるのは、4日目であるので、力は4となる。

この考え方に基づき、Kさんは配列 **Akibi** の要素と、部員数が代入された変数 **buinsu** を用いて、次に割り当てる工芸品の担当部員を表示するプログラムを作成した(図4)。ここでは例として、(01)行目で図3のように配列 **Akibi** を設定している。

```
(01) Akibi = [5, 3, 力]  
(02) buinsu = 3  
(03) tantou = 1  
(04) buin を 2 から buinsu まで1ずつ増やしながら繰り返す:  
(05) | もし キ ならば:  
(06) | | tantou = buin  
(07) 表示する("次の工芸品の担当は部員", tantou, "です。")
```

図4 次に割り当てる工芸品の担当部員を表示するプログラム

仮に部員数が変わったとしても、配列 **Akibi** と変数 **buinsu** を適切に設定すれば、このプログラムを用いることができる。部員が5名に増えた場合、(01)行目を例えば **Akibi = [5, 6, 4, 4, 4]** に、(02)行目を **buinsu = 5** に変更して図4のプログラムを実行すると、(06)行目の代入がク回行われ、「次の工芸品の担当は部員3です。」と表示される。

キの解答群

- |                 |                               |
|-----------------|-------------------------------|
| ① buin < tantou | ① Akibi[buin] < Akibi[tantou] |
| ② buin > tantou | ③ Akibi[buin] > Akibi[tantou] |

## 解説

この考え方に基づくと、配列 Akibi の中から、最も小さい値が入っている添字(部員番号)を探せばよいこととなる。

全体の部員数を `buinsu` に格納し、  
次の工芸品を担当する部員番号を `tantou` に格納する。

図4のプログラムを、BASIC プログラムで記述してみます。

<pre> Cls Dim Akibi(3) For I=1 To 3 Step 1   Read Akibi(I) Next Data 5,3,4 buinsu = 3 tantou = 1  For buin = 2 To buinsu Step 1   If Akibi(buin) &lt; Akibi(tantou) Then     tantou = buin   End If Next         </pre>	<p>配列を宣言します。 配列 Akibi に値を格納 ↓ ↓ ↓ buinsu は 3 人 初期値として担当者を 1 にしておく</p> <p>次の部員から最後の部員まで繰り返す 次の部員の空き日が、担当者の空き日 より小さければ、次の部員を担当者と する。</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

※`tantou`(担当者)の初期値を 1 人目にしているため、繰り返し `For` の初期値は、次の部員である 2 として大丈夫である。

`For buin = 1 To buinsu Step 1` としても結果には問題はないが、1 回目の比較が、  
`If Akibi(1) < Akibi(1) Then` となり、意味のない比較をすることとなる。

よって、図4のプログラムのキは、

```

(01) Akibi = [5, 3, 2]
(02) buinsu = 3
(03) tantou = 1
(04) buin を 2 から buinsu まで 1 ずつ増やしながら繰り返す:
(05) | もし キ ならば:
(06) | | tantou = buin
(07) 表示する ("次の工芸品の担当は部員", tantou, "です。")
        
```

① `Akibi[buin]<Akibi[tantou]`

となる。

問 3 次の文章を読み、空欄 **ケ** ~ **シ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

次に K さんは、工芸部の部員数と、表 1 のような各工芸品の製作日数を用いて、図 2 のような一覧を表示するプログラムを作ることにした。

表 1 各工芸品の製作日数(再掲)

工芸品	1	2	3	4	5	6	7	8	9
製作日数	4	1	3	1	3	4	2	4	3

工芸品 1	…	部員 1	：	1 日目	～	4 日目
工芸品 2	…	部員 2	：	1 日目	～	1 日目
工芸品 3	…	部員 3	：	1 日目	～	3 日目
工芸品 4	…	部員 <b>ア</b>	：	<b>イ</b> 日目	～	<b>イ</b> 日目
工芸品 5	…	部員 <b>ウ</b>	：	<b>エ</b> 日目	～	<b>オ</b> 日目
~~~~~						
工芸品 9	…	部員 1	：	7 日目	～	9 日目

図 2 各工芸品の担当と期間を一覧にしたメールの文面(再掲)

表 1 をプログラムで扱うために、K さんは工芸品の番号順に製作日数を並べた配列 **Nissu** (添字は 1 から始まる。) を用意した。さらに、工芸品数 9 が代入された変数 **kougeihinsu**、各部員が空きになる日付を管理する配列 **Akibi**、部員数 3 が代入された変数 **buinsu** を用いて、図 2 の一覧を表示するプログラムを作成した(図 5)。最初はどの部員も合宿初日すなわち 1 日目で空きであるため、(03)行目で配列 **Akibi** の各要素を 1 に設定している。

工芸品の番号を表す変数 **kougeihin** を用意し、(05)～(11)行目で各工芸品に対して順に担当と期間を求めていく。破線で囲まれた(06)～(09)行目は問 2 における図 4 の(03)～(06)行目と同じもので、次に割り当てる工芸品の担当部員の番号を変数 **tantou** に代入する処理を行う。(10)行目で図 2 の 1 行分を表示し、(11)行目で担当部員が空きになる日付を更新する。



```

(01) Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]
(02) kougaihinsu = 9
(03) Akibi = [1, 1, 1]
(04) buinsu = 3
(05)  を 1 から  まで 1 ずつ増やしながら繰り返す:
(06) | tantou = 1
(07) | buin を 2 から buinsu まで 1 ずつ増やしながら繰り返す:
(08) | | もし  ならば:
(09) | |   tantou = buin
(10) | 表示する ("工芸品", kougaihin, " ... ",
      | "部員", tantou, ":",
      | Akibi[tantou], "日目~",
      | Akibi[tantou] + , "日目")
(11) |  Akibi[tantou] = Akibi[tantou] + 

```

図 5 各工芸品の担当と期間の一覧を表示するプログラム

・  の解答群

① buin	① kougaihin	② tantou
③ buinsu	④ kougaihinsu	

・  の解答群

① Nissu[kougaihin]	① Nissu[tantou]
② Nissu[kougaihin] - 1	③ Nissu[tantou] - 1
④ Nissu[kougaihin - 1]	⑤ Nissu[tantou - 1]

## 解説

図5のプログラムを、BASICプログラムで記述してみます。

```
Cls
Dim Nissu(9),Akibi(3)
(01) For I=1 To 9 Step 1
      Read Nissu(I)
      Next
      Data 4,1,3,1,3,4,2,4,3
(02) kougeihinsu = 9
(03) For I=1 To 3 Step 1
      Read Akibi(I)
      Next
      Data 5,3,4
(04) buinsu = 3
(05) For kougeihin = 1 To kougeihinsu Step 1
(06)   tantou = 1
(07)   For buin = 2 To buinsu Step 1
(08)     If Akibi(buin) < Akibi(tantou) Then
(09)       tantou = buin
      End If
    Next
    Print "工艺品" > kougeihin ; "…";
      "部員"; tantou ; "：";
      Akibi(tantou) ; "日目～";
      Akibi(tantou) + Nissu(kougeihin)-1 ; "日目")
(11) Akibi(tantou) = Akibi(tantou) + Nissu(kougeihin)
  Next
```

## ケコ

今回行う処理は、工艺品1～9をどの部員が担当し、製作日数が何日目～何日目かを表示することである。

よって、変数 `kougeihin` を `kougeihinsu` まで1ずつ増やしながら繰り返す必要がある。変数 `kougeihin` を使用することは、表示する部分で、工艺品番号の変数に `kougeihin` を使用していることから分かる。

## サ

配列 Akibi(部員番号) には、その担当の空き日となる日が格納されている。

配列 Nissu(工艺品番号) には、その工艺品の製作日数が格納されている。

仮に、3日目から製作日数4日かかるとすれば、

表示は、『3日目～6日目』となる。

よって、式は、`Akibi(tantou)+Nissu(kougeihin)-1` となる。

シ

担当に工芸品を割り当てた後、配列 Akibi (部員番号) を、次の空き日に修正する必要がある。

仮に、部員 1 番の空き日が 3 日目で、工芸品 6 の製作日数 4 日を割り当てたとすれば、次の空き日は 7 日目となる。

	0	1	2	3	4	5	6	7
部員 1	□	□	□	□	□	□	□	□
部員 2	□	□	□	□	□	□	□	□
部員 3	□	□	□	□	□	□	□	□

よって式は、部員番号の空き日に、製作日数を加算すればよいこととなる。

$$\text{Akibi (tantou)} = \text{Akibi (tantou)} + \text{Nissu (kougeihin)}$$