5．共通テスト用プログラム表記の例示

高等学校の「情報Ⅰ」の授業で使用するプログラミング言語は多様であることから，共通テスト『情報Ⅰ』の試作問題作成にあたり，共通テスト用のプログラム表記を使用します。以下，参考のためにその基本を例示します。しかしながら，問題文の記述を簡潔にするなどの理由で，この説明文書の記述内容に従わない形式で出題することもあります。したがって，共通テスト『情報Ⅰ』の受験に際しては，当該問題文の中の説明や指示に注意し，それらに沿って解答してください。なお，経過措置問題『旧情報（仮）』についても同様に扱うこととします。

１変数

通常の変数例：kosu，kingaku_kei
（変数名は英字で始まる英数字と「_」の並び）

配列変数の例：Tokuten[3]，Data[2,4]　（配列名は先頭文字が大文字）

※特に説明がない場合，配列の要素を指定する添字は0から始まる

２文字列

文字列はダブルクォーテーション（"）で囲む

```
moji = "I'll be back."
message = "祇園精舎の" + "鐘の声"          ※ +で連結できる
```

３代入文

```
kosu = 3 , kingaku = 300          ※複数文を１行で表記できる
kingaku_goukei = kingaku * kosu
namae = "Komaba"
Data = [10,20,30,40,50,60]
Tokutenのすべての値を0にする
nyuryoku =【外部からの入力】
```

４算術演算

加減乗除の四則演算は，『+』，『-』，『*』，『/』で表す

整数の除算では，商（整数）を『÷』で，余りを『%』で表す

べき乗は『**』で表す

５比較演算

『==』（等しい），『!=』（等しくない），『>』，『<』，『>=』，『<=』

６論理演算

『and』（論理積），『or』（論理和），『not』（否定）

７関数

値を返す関数例：kazu = **要素数**(Data)

saikoro = **整数**(**乱数**()*6)+1

値を返さない関数例：表示する(Data)

表示する(Kamoku[i],"の得点は",Tensu[i],"です")

※「表示する」関数はカンマ区切りで文字列や数値を連結できる

※「表示する」関数以外は基本的に問題中に説明あり

８制御文（条件分岐）

```
もし x < 3 ならば:
│  x = x + 1
└ y = y + 1

もし x == 3 ならば:
│  x = x - 1
そうでなければ:
└ y = y * 2
```

```
もし x >= 3 ならば:
│  x = x - 1
そうでなくもし x < 0 ならば:
│  x = x * 2
そうでなければ:
└ y = y * 2
```

※ │ と └で制御範囲を表し，└は制御文の終わりを示す

９制御文（繰返し）

```
x を 0 から 9 まで 1 ずつ増やしながら繰り返す:
└ goukei = goukei + Data[x]
```

※「減らしながら」もある

```
n < 10 の間繰り返す:
│  goukei = goukei + n
└ n = n + 1
```

※ │ と └で制御範囲を表し，└は制御文の終わりを示す

10 コメント

```
atai = 乱数()      #0以上1未満のランダムな小数をataiに代入する
```

※１行内において#以降の記述は処理の対象とならない

1　共通テスト用プログラム表記例（二分探索のアルゴリズム）

共通テスト用プログラム表記例

```
(1) Data=[3,18,29,33,48,52,62,77,89,97]
(2) kazu=要素数(Data)
(3) 表示する("0～99の数字を入力してください")
(4) atai=【外部からの入力】
(5) hidari=0 , migi=kazu-1
(6) owari=0
(7) hidari <= migi and owari==0 の間繰り返す:
(8) │  aida=(hidari+migi)÷2        #演算子÷は商の整数値を返す
(9) │  もし Data[aida]==atai ならば:
(10) │  │  表示する(atai,"は",aida,"番目にありました")
(11) │  │  owari=1
(12) │  そうでなくもし Data[aida]<atai ならば:
(13) │  │  hidari=aida+1
(14) │  そうでなければ:
(15) └  └  migi=aida-1
(16) もし owari==0 ならば:
(17) └  表示する(atai,"は見つかりませんでした")
(18) 表示する("添字"," ","要素")
(19) iを0からkazu-1まで1ずつ増やしながら繰り返す:
(20) └  表示する(i,"     ",Data[i])
```

関数の説明

要素数(配列)･･･配列の要素数を返す

例：Data=[10, 20, 30, 40, 50, 60, 70, 80]の時

　要素数(Data)は8を返す

配列

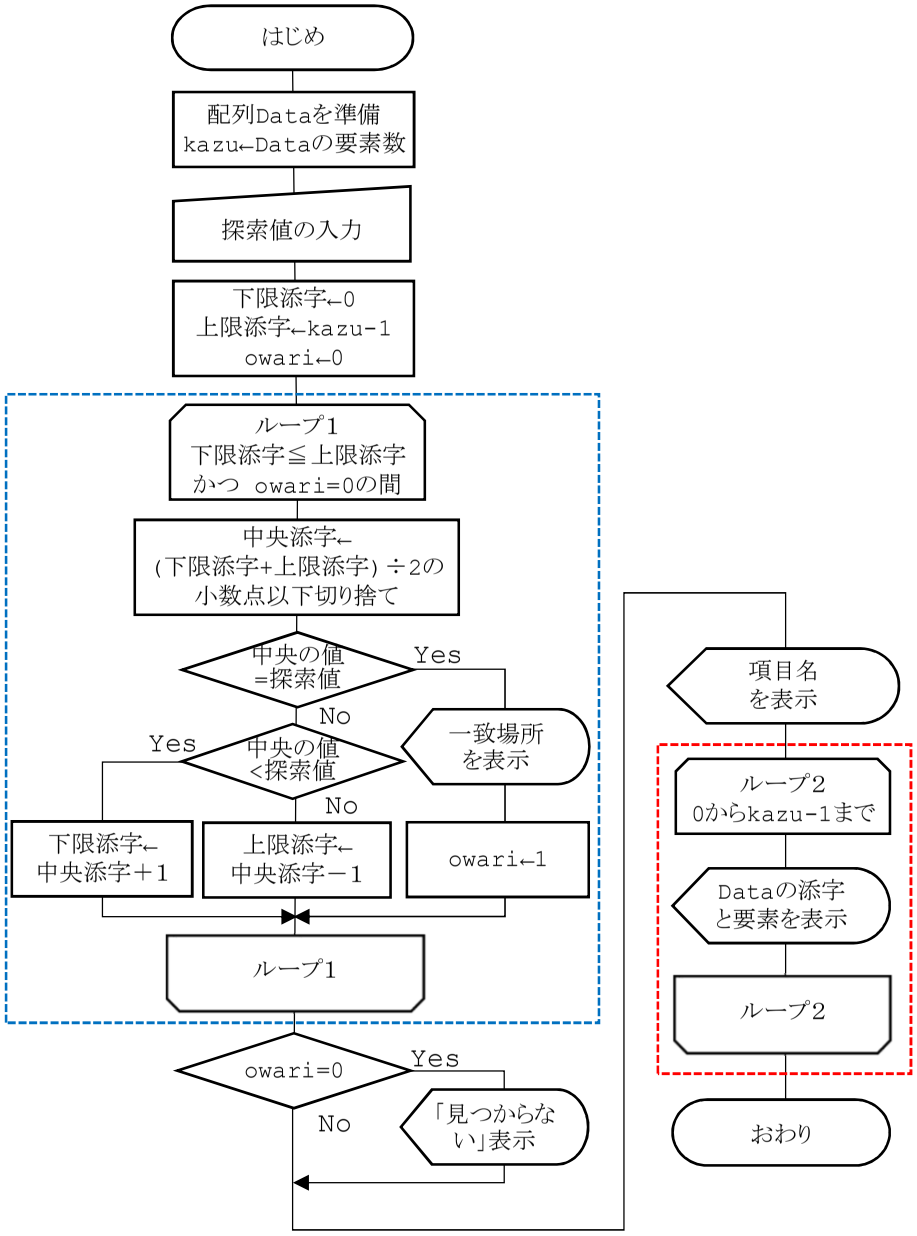
添字	0	1	2	3	4	5	6	7	8	9
Data	3	18	29	33	48	52	62	77	89	97

実行結果の表示例

```
0～99の数字を入力してください
52 ←キーボードから入力
52 は 5 番目にありました
添字      要素
0         3
1         18
2         29
3         33
4         48
5         52
6         62
7         77
8         89
9         97
```

```
0～99の数字を入力してください
85 ←キーボードから入力
85 は見つかりませんでした
添字      要素
0         3
1         18
2         29
3         33
:          :
```

2　フローチャートと共通テスト用プログラム表記例の対比（例：二分探索）



共通テスト用プログラム表記例

```
(1) Data=[3,18,29,33,48,52,62,77,89,97]
(2) kazu=要素数(Data)
(3) 表示する("0～99の数字を入力してください")
(4) atai=【外部からの入力】
(5) hidari=0 , migi=kazu-1
(6) owari=0
(7) hidari <= migi and owari==0 の間繰り返す:
(8)     aida=(hidari+migi)÷2        #演算子÷は商の整数値を返す
(9)     もし Data[aida]==atai ならば:
(10)        表示する(atai,"は",aida,"番目にありました")
(11)        owari=1
(12)    そうでなくもし Data[aida]<atai ならば:
(13)        hidari=aida+1
(14)    そうでなければ:
(15)        migi=aida-1
(16) もし owari==0 ならば:
(17)    表示する(atai,"は見つかりませんでした")
(18) 表示する("添字"," ","要素")
(19) iを0からkazu-1まで1ずつ増やしながら繰り返す:
(20)    表示する(i,"     ",Data[i])
```

関数の説明

要素数(配列)･･･配列の要素数を返す

例：Data = [10, 20, 30, 40, 50, 60, 70, 80]の時

　要素数(Data)は8を返す

3　Python3と共通テスト用プログラム表記例の対比（例：二分探索）

Python3記述例

```
1  data=[3,18,29,33,48,52,62,77,89,97]
2  kazu=len(data)
3  atai=int(input('0～99の数字を入力してください'))
4  hidari=0;migi=kazu-1
5  owari=0
6  while hidari <= migi and owari==0:
7    aida=(hidari+migi)//2      #演算子//は商の整数部分
8    if data[aida]==atai:
9      print(atai,'は',aida,'番目にありました')
10     owari=1
11   elif data[aida]<atai:
12     hidari=aida+1
13   else:
14     migi=aida-1
15 if owari==0:
16   print(atai,'は見つかりませんでした')
17 print('添字','  ','要素')
18 for i in range(0,kazu,1):
19   print(i,'    ',data[i])
```

共通テスト用プログラム表記例

```
(1)  Data=[3,18,29,33,48,52,62,77,89,97]
(2)  kazu=要素数(Data)
(3)  表示する("0～99の数字を入力してください")
(4)  atai=【外部からの入力】
(5)  hidari=0 , migi=kazu-1
(6)  owari=0
(7)  hidari <= migi and owari==0 の間繰り返す:
(8)  │  aida=(hidari+migi)÷2        #演算子÷は商の整数値を返す
(9)  │  もし Data[aida]==atai ならば:
(10) │  │  表示する(atai,"は",aida,"番目にありました")
(11) │  │  owari=1
(12) │  そうでなくもし Data[aida]<atai ならば:
(13) │  │  hidari=aida+1
(14) │  そうでなければ:
(15) │  └  migi=aida-1
(16) もし owari==0 ならば:
(17) └  表示する(atai,"は見つかりませんでした")
(18) 表示する("添字"," ","要素")
(19) iを0からkazu-1まで1ずつ増やしながら繰り返す:
(20) └  表示する(i,"    ",Data[i])
```

関数の説明

要素数(配列)･･･配列の要素数を返す

例：Data = [10, 20, 30, 40, 50, 60, 70, 80]の時

　要素数(Data)は8を返す