

プログラミング問題PCIモジュール

問題例とその作成方法

(①バブルソート問題)

作成するバブルソート問題

問題 配列Dataに格納されている値を昇順に並べ替えるバブルソート(隣接交換法)のプログラムを完成させよ。

配列Data

89,79,89,23,5,21,22,27,62,98,62,53,35,27,49,50,41,2,5,58,26,48,43,10,23

短冊型コード選択肢

🔄 n を saigo から m-1 まで
1 ずつ 減らしながら繰り返す:

もし Data[n] < Data[n-1]:

Data[n] = Data[n-1]

Data[n-1] = tmp

tmp = Data[n]

解答欄

Data = [89,79,89,23,5,...(省略)...
,43,10,23]

saigo = 要素数(Data)-1

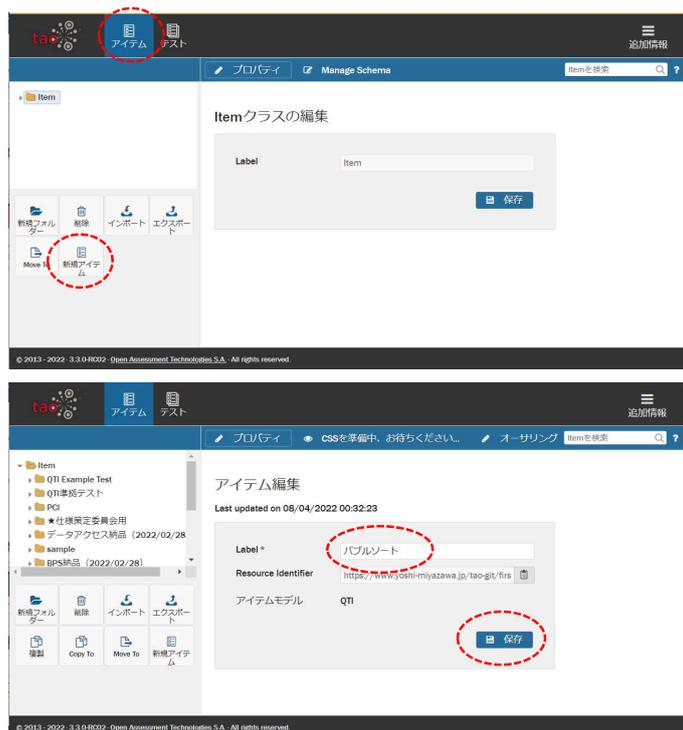
🔄 m を 0 から saigo まで
1 ずつ 増やしなが繰り返す:

🖥 表示する (Data)

出題者があらかじめ解答欄に配置する「固定短冊」には、そのことが分かる押しピンのアイコンが付されている。

プログラミング問題PCIモジュールを使用したバブルソート問題の作成方法

1. アイテム(問題)を新規作成する。



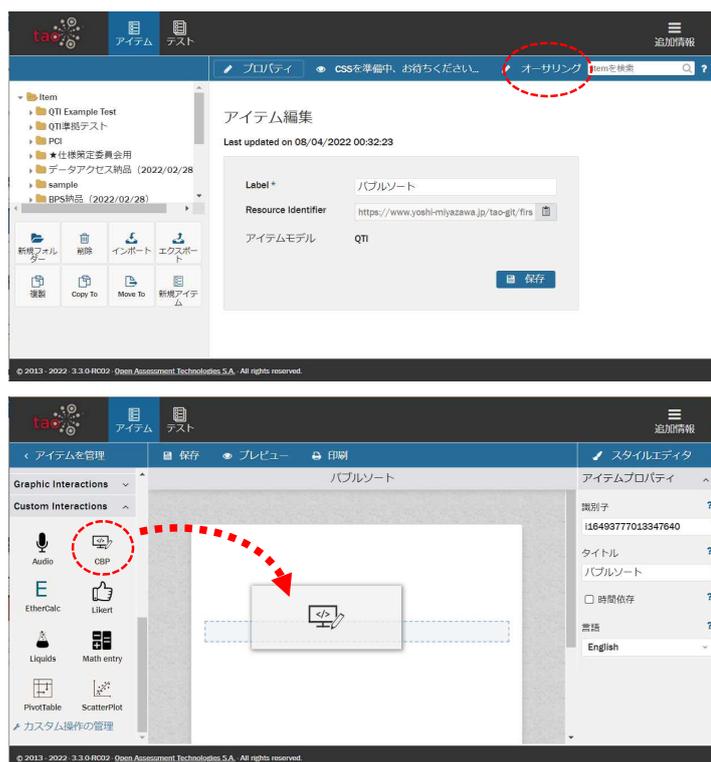
(1) ヘッダーの「アイテム」をクリックする。

(2) 左サイドバーの「新規アイテム」をクリックする。

(3) 「Label」欄にアイテムの名前を入力する。
(例「バブルソート」)

(4) 保存をクリックしてアイテムを保存する。

2. アイテム(問題)にプログラミング問題PCIモジュールを挿入する。



(1) 上部の「オーサリング」をクリックしてアイテム(問題)の編集画面を開く。

(2) 左サイドバー「Custom Interactions」を展開し中にある「CBP」を右側の画面にドラッグ・アンド・ドロップする。

※あらかじめTAOにプログラミング問題PCIモジュールをインストールしておく必要があります。PCIモジュールのインストール方法については、本報告書の付録2-2を参照してください。

3. 「解答欄」に固定する短冊型コード(固定短冊)を作成する。



(1) 「短冊型コード選択肢」欄内の短冊タイプをクリックする。

- ・「process」を2回クリック
- ・「for」を1回クリック
- ・「print」を1回クリック



(「短冊型コード選択肢」欄内にそれぞれのひな型の短冊型コードが作成される。)



(2) ひな型短冊型コード(process)の「code#1」をクリックする。

(3) 右側の「選択肢プロパティ」に表示される項目「実行コード」「表示内容」にそれぞれ次のコードを入力する。

「実行コード」

```
let data = [89,79,89,23,5,21,22,27,62,98,62,53,35,27,49,50,41,2,5,58,26,48,43,10,23];
```

「表示内容」

```
Data = [89,79,89,23,5,...(省略)...,43,10,23]
```



(4) 「表示内容」がすべて短冊型コードに表示されるように適当な箇所で改行を入れる。

(例)

```
Data = [89,79,89,23,5,...(省略)...  
,43,10,23]
```

短冊型コードの表示

```
Data = [89,79,89,23,5,...(省略)...  
,43,10,23]
```

(5) 同様に各ひな型短冊型コードに「実行コード」と「表示内容」をそれぞれ設定する。

ひな型短冊	実行コード	表示内容
<p>process</p> 	<pre>saigo = data.length-1;</pre>	<pre>saigo = 要素数(Data)-1</pre> 
<p>for</p> 	<pre>m = 0; m <= saigo; m++</pre>	<p>m を 0 から saigo まで 1 ずつ 増やしながら繰り返す:</p> 
<p>print</p> 	<pre>data</pre>	<p>表示する (Data)</p> 

4. 短冊型コード(固定短冊)を配置する。

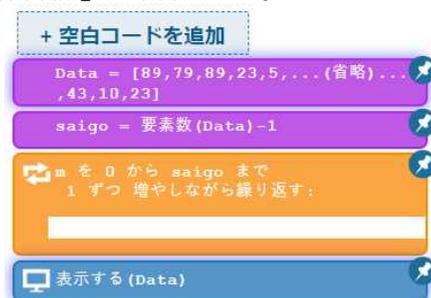


(1) 3. で作成した固定短冊をドラッグ・アンド・ドロップで解答欄に移す。

(2) 解答欄」に配置した短冊コードの右側にある固定ボタンをクリックする
(固定ボタンを押すと、当該短冊は「短冊型コード選択肢」欄から削除される。)



(3) 同様に、固定短冊にしたい他の短冊型コードを「解答欄」に配置する。



5. 解答欄に空白コードを配置する。



(1) 受験者がどこに短冊型コード選択肢を配置すれば良いか、分かりやすくするために「空白コード」を配置することができる（必ず配置しなければならないわけではない。）。「解答欄」の「+空白コードを追加」をクリックすると、「空白パーツ」が追加される



(2) 配置された「空白パーツ」を、配置したい場所にドラッグ・アンド・ドロップで移動させる。この問題の場合は、短冊型コードのforの繰り返し部分に配置する。

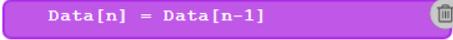
6. 受験者が解答時に並べ替える短冊型コード選択肢を作成する。



(1) 「短冊型コード選択肢」欄内の短冊タイプをクリックする。

- ・ 「for」を1回クリック
- ・ 「if」を1回クリック
- ・ 「process」を3回クリック

(2) 各ひな型短冊型コードに「実行コード」と「表示内容」をそれぞれ設定する。

ひな型短冊	実行コード	表示内容
<p>for</p> 	$n = \text{saigo}; n \geq m; n--$	<p>n を saigo から m-1 まで 1 ずつ減らしながら繰り返す:</p> 
<p>if</p> 	$\text{data}[n] < \text{data}[n-1]$	<p>もし $\text{Data}[n] < \text{Data}[n-1]$:</p> 
<p>process</p> 	$\text{data}[n] = \text{data}[n-1];$	$\text{Data}[n] = \text{Data}[n-1]$ 
<p>process</p> 	$\text{data}[n-1] = \text{tmp};$	$\text{Data}[n-1] = \text{tmp}$ 
<p>process</p> 	$\text{tmp} = \text{data}[n];$	$\text{tmp} = \text{Data}[n]$ 

7. 正答を作成する。



(1) 「解答」をクリックして解答画面を表示する。



(2) 短冊型コード選択肢を解答欄にドラッグ・アンド・ドロップして正解プログラムを作成する。

8. 実行結果を確認する。

解答欄

```
Data = [89, 79, 89, 23, 5, ... (省略) ... , 43, 10, 23]
saigo = 要素数(Data) - 1
for (m = 0 から saigo まで 1 ずつ 増やしながら繰り返す:
  for (n を saigo から m-1 まで 1 ずつ 減らしながら繰り返す:
    もし Data[n] < Data[n-1]:
      tmp = Data[n]
      Data[n] = Data[n-1]
      Data[n-1] = tmp
  )
)
表示する (Data)
```

実行結果

[2, 5, 5, 10, 21, 22, 23, 23, 26, 27, 27, 35, 41, 43, 48, 49, 50, 53, 58, 62, 62, 77, 77, 89, 98, 98, 98]

プログラムの実行が完了しました。

JavaScriptコード リセット 実行

↑
実行結果が表示される

(1) 「実行」ボタンをクリックして実行結果を表示し、正しくソートされているかを確認する。

(2) エラーが発生した場合は、「JavaScriptコード」ボタンをクリックしてソースを確認する。

```
JavaScriptコード
```

```
let data = [89, 79, 89, 23, 5, 21, 22, 27, 62, 98, 62, 53, 35, 27, 49, 50, 41, 2, 5, 58, 26, 48, 43, 10, 23];
saigo = data.length - 1;
for (m = 0; m <= saigo; m++) {
  for (n = saigo; n >= m; n--) {
    if (data[n] < data[n - 1]) {
      tmp = data[n];
      data[n] = data[n - 1];
      data[n - 1] = tmp;
    }
  }
}
printBlock(data);
```

閉じる

JavaScriptのソースコードが表示される

9. 実行結果正答(正答として登録される解答)を確認する。

実行結果正答の確認

(1) 「実行結果正答の確認」ボタンをクリックすると、正答として登録される解答を確認することができる。

※正答管理タブ画面で使用している短冊型コードを編集した場合は、正答が初期化されます。

実行結果正答

[2, 5, 5, 10, 21, 22, 23, 23, 26, 27, 27, 35, 41, 43, 48, 49, 50, 53, 58, 62, 62, 79, 89, 77, 77, 98, 98, 98]

閉じる

10. 問題文を作成する。



(1) 問題文をInline Interactionsの文字ブロックなどを使用して作成する。

※Programming Interaction (CBP) 内を編集すると正答が初期化されるので、別のInteractionsを利用して問題文を作成する。

配列Dataに格納されている値を昇順に並べ替えるバブルソート（隣接交換法）の手続きを、短冊型コード選択肢から適切なものを選び完成させよ。なお、配列の要素を指定する添え字は0から始まり、手続きは次のような関数を用いている。

要素数 (値) …配列の要素数を返す。
 例：Data=[10, 20, 30, 40, 50, 60, 70]の時
 要素数 (Data) は7を返す。

【参考】作成したバブルソート問題の解答方法

「短冊形コード選択肢」(左側)から「解答欄」にドラッグ・アンド・ドロップで短冊を移動させて解答する。

画面下の **実行** ボタンを押すと、作成したプログラムの実行結果が実行結果エリアに表示される

解答を終えたら **送信** ボタンを押して解答を提出する。

実行結果
 [2, 5, 5, 10, 21, 22, 23, 23, 26, 27, 27, 35, 41, 43, 48, 49, 50, 53, 58, 62, 62,
 プログラムの実行が完了しました。

実行

送信