

# 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

### (②シーザー暗号問題)

#### 作成するシーザー暗号問題

シーザー暗号に代表される古典的な暗号化の方法であるシフト暗号はアルファベットの文字を決まった文字数分シフトさせて（ずらして）置き換える極めて単純な暗号手段である。  
例えばアルファベットを5文字右にシフトした場合は、図1のように文字「a」は文字「f」に、文字「x」はまず2文字シフトして右端に達した後一番左端に戻り3文字シフトした文字「c」に置き換わる。



図1 5文字右シフトした場合の考え方

次の暗号文は、アルファベットを10文字左にシフトされている。つまり、先頭の文字「y」を復号するには、まず1文字シフトして右端に達した後一番左端に戻り9文字シフトした文字「i」ということになる。

"ycqwydu jxuhu'i de xuqlud yj'i uqio yv oek jho de xubb rubem ki qrelu ki edbo iao ycqwydu qbb jxu fuefbu bylydw veh jetqo..."

この暗号文を復号するプログラムを完成させて、復号した結果を表示せよ。  
ただし、次の関数を用いている。

#### 【関数の説明】

**要素数 (値)**…配列の要素数を返す。

例：Data=["M", "i", "s", "s", "i", "s", "s", "i", "p", "p", "i"]の時  
要素数(Data)は11を返す

**差分 (値)**…アルファベットの「a」との位置の差分を返す

値がアルファベット以外の文字であれば-1を返す

例：差分("e")は4を、差分("x")は23を返す  
差分("5")や差分(",")は-1を返す

**文字 (値)**…番号の値に対するアルファベットの文字を返す。

値が0以上25以下でなければ「アルファベットでない」を返す

例：文字(4)は「e」を、文字(23)は「x」を返す  
文字(-1)や文字(27)は「アルファベットでない」を返す

短冊型コード選択肢

bangou=差分 (Angoubun [m])

Hirabun [m] = 文字 [bangou+shift]

Hirabun [m] = 文字 [bangou+shift-26]

Hirabun [m] =  
Angoubun [ 選択してください ]

もし bangou != -1 ならば:

でなければ

もし bangou+shift <= ならば:

でなければ

解答欄

Angoubun = ["y","q","q","w","y",  
"d","u","n","j","."] (省略...)

配列変数Hirabun を初期化する

shift=10

も0 から 要素数 (Angoubun) -1 まで  
1 ずつ増やしながら繰り返す:

配列をつなげて表示する (Hirabun)

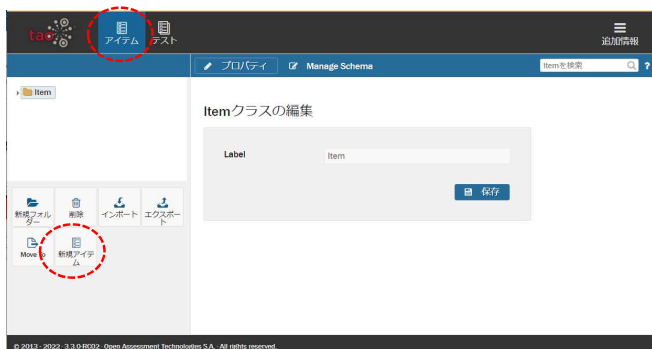
実行結果

リセット    実行

送信

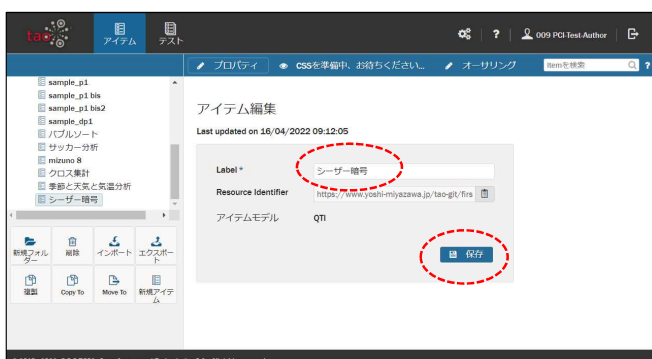
## プログラミング問題PCIモジュールを使用したシーザー暗号問題の作成方法

### 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。

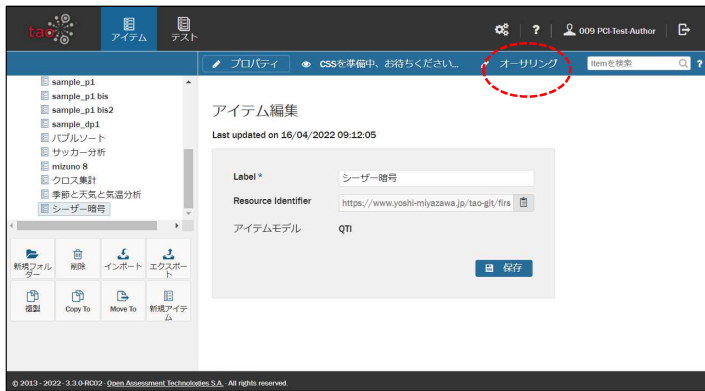
(2) 左サイドバーの「新規アイテム」をクリックする。



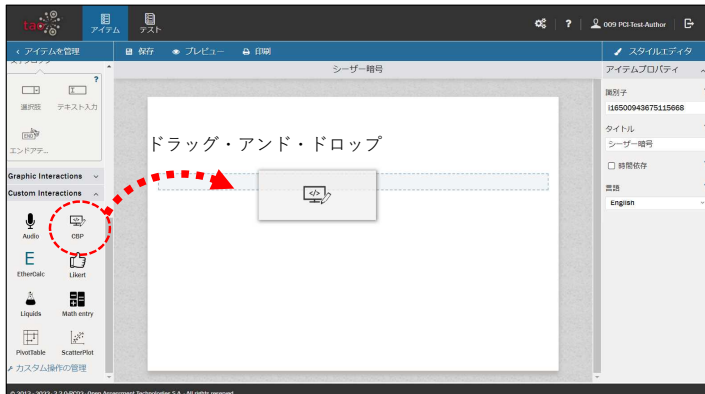
(3) 「Label」欄にアイテムの名前を入力する。  
(例「シーザー暗号」)

(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。

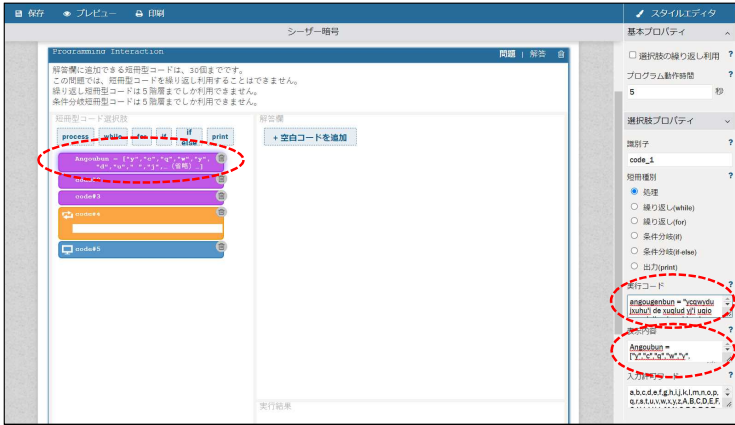


(1) 「短冊型コード選択肢」欄内の短冊タイプをクリックする。

- ・ 「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。)



この短冊には関係がないものでも、必要な処理を入れておく

(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」

```
angougenbun = "ycqwydu jxuhu'i de xuqlud yj'i uqio yv oek jho de xubb rubem ki qrelu ki edbo iao ycqwydu qbb jxu fuefbu bylydw veh jetqo...";
let array_angou = angougenbun.split("");
alphabet="abcdefghijklmnopqrstuvwxyz";
let array_alphabet = alphabet.split("");
```

「表示内容」

```
Angoubun = ["y", "c", "q", "w", "y", "d", "u", " ", "j", ... (省略) ...]
```

短冊型コードの表示


```
Angoubun = ["y", "c", "q", "w", "y", "d", "u", " ", "j", ... (省略) ...]
```

(4) 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。

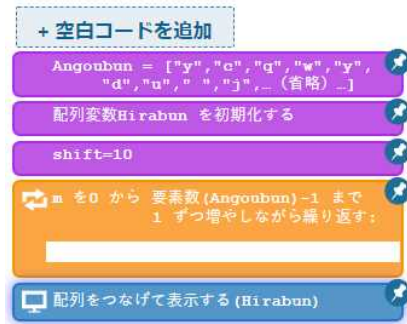
ひな型短冊	実行コード	表示内容
process code#2	<pre>var array_hirabun = new Array();</pre>	配列変数Hirabun を初期化する 配列変数Hirabun を初期化する
process code#3	<pre>shift=10;</pre>	shift=10 shift=10
for code#4	<pre>m=0;m&lt;=array_angou.length-1;m++</pre>	m を0 から 要素数(Angoubun)-1 まで 1 ずつ増やしながら: m を0 から 要素数(Angoubun)-1 まで 1 ずつ増やしながら繰り返す:
print code#5	<pre>array_hirabun.join("")</pre>	配列をつけて表示する(Hirabun) 配列をつなげて表示する (Hirabun)

## 4. 短冊型コード(固定短冊)を配置する。

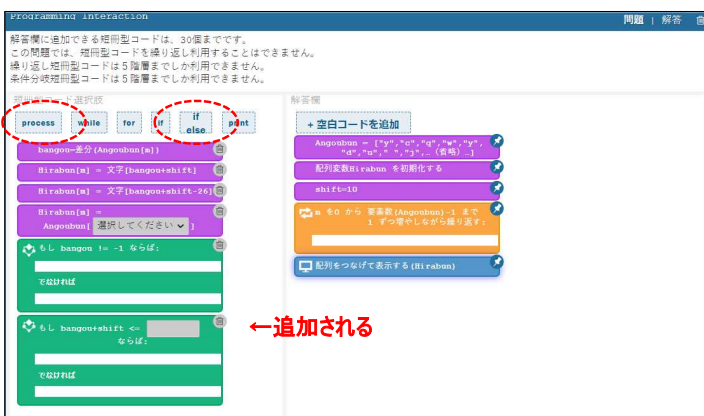


- (1) 3. で作成した固定短冊をドラッグ・アンド・ドロップで解答欄に移す。
- (2) 解答欄」に配置した短冊コードの右側にある固定ボタン  をクリックする  
(固定ボタンを押すと、当該短冊は「短冊型コード選択肢」欄から削除される。)

- (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. 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。
  - ・「process」を4回クリック
  - ・「if else」を2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
process code#6	<code>bangou=array_alphabet.indexOf(array_angou[m]);</code>	bangou=差分(Angoubun[m]) bangou=差分(Angoubun[m])
process code#7	<code>array_hirabun[m]=array_alphabet[bangou+shift];</code>	Hirabun[m] = 文字[bangou+shift] Hirabun[m] = 文字[bangou+shift]
process code#8	<code>array_hirabun[m]=array_alphabet[bangou+shift-26];</code>	Hirabun[m] = 文字[bangou+shift-26] Hirabun[m] = 文字[bangou+shift-26]
process code#9	<code>array_hirabun[m]=array_angou[¥selectBox_012];</code>	Hirabun[m] = Angoubun[¥selectBox_012[bangou、 shift、 m]] Hirabun[m] = Angoubun[ 選択してください ]

ひな型短冊	実行コード	表示内容
If else code#10 でなければ	<code>bangou!=-1</code>	もし bangou != -1 ならば: もし bangou != -1 ならば: でなければ
If else code#11 でなければ	<code>bangou+shift&lt;=¥textField_d_123</code>	もし bangou+shift <= ¥textField_123 ならば: もし bangou+shift <= [ ] ならば: でなければ

## 6. 正答を作成する。

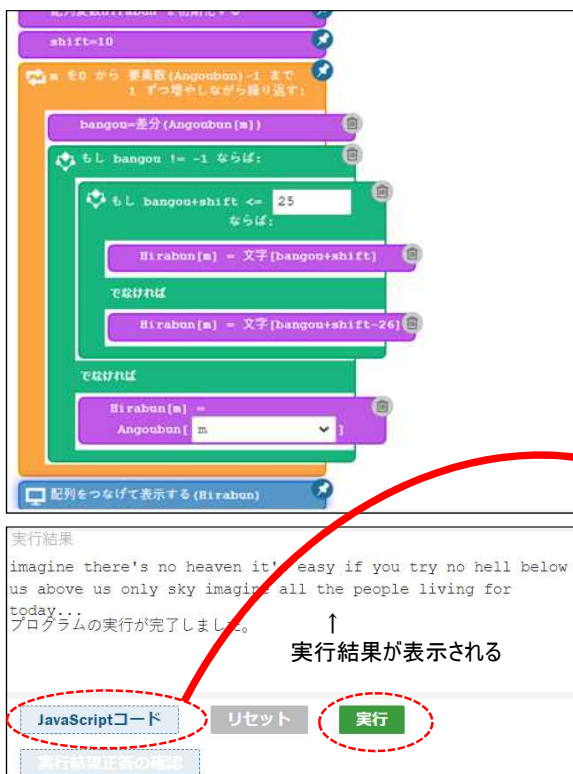


(1) 「解答」をクリックして解答画面を表示する。



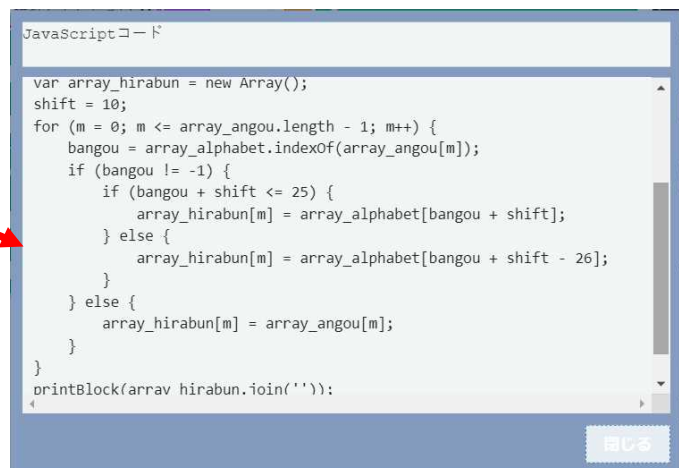
(2) 短冊型コード選択肢を解答欄にドラッグ・アンド・ドロップして正解プログラムを作成する。

## 7. 実行結果を確認する。



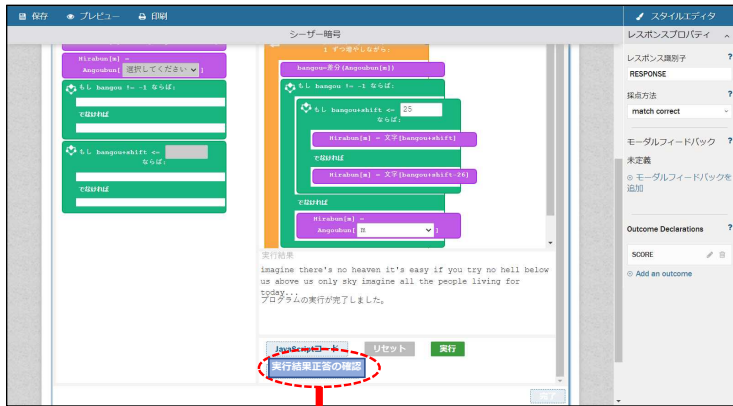
(1) 「実行」ボタンをクリックして実行結果を表示し、正しくソートされているかを確認する。

(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



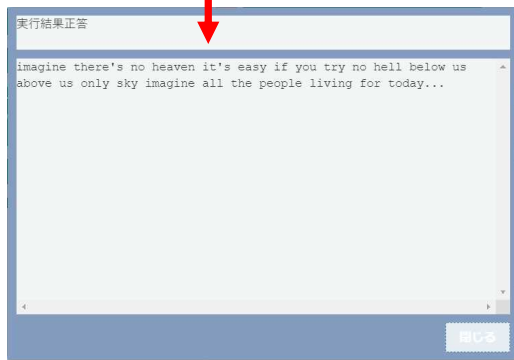
JavaScriptのソースコードが表示される

## 8. 実行結果正答(正答として登録される解答)を確認する。

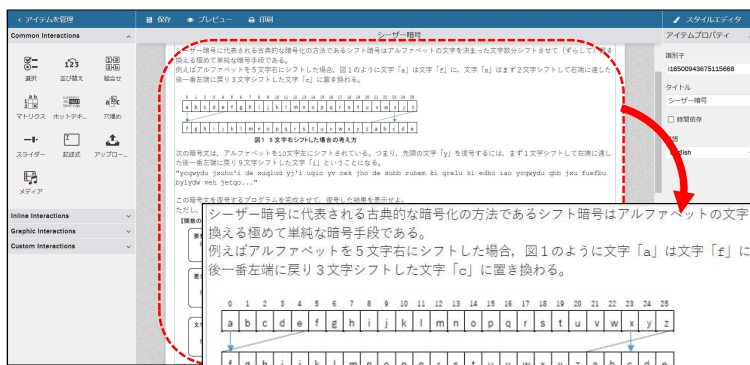


(1) 「実行結果正答の確認」ボタンをクリックすると、正答として登録される解答を確認することができる。

※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。



## 9. 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。

次の暗号文は、アルファベットを10文字左にシフトされている。つまり、先頭の文字「y」を復号するには、まず1文字シフトして右端に達した後一番左端に戻り9文字シフトした文字「i」ということになる。

"yqcwydu jxuhu'i de xquqlud yj'i uqio yv oek jho de xubb rubem ki qrelu ki edbo iao yqcwydu qbb jxu fuefbu bylydw veh jetgo..."

この暗号文を復号するプログラムを完成させて、復号した結果を表示せよ。ただし、次の関数を用いている。

**【関数の説明】**

- 要素数 (値)...**配列の要素数を返す。  
例: Data=["M","i","a","a","i","a","a","i","p","p","i"]の時  
要素数(Data)は11を返す
- 差分 (値)...**アルファベットの「a」との位置の差分を返す  
値がアルファベット以外の文字であれば-1を返す  
例: 差分("ae")は4を、差分("x")は23を返す  
差分("5")や差分("r","")は-1を返す
- 文字 (値)...**番号の値に対するアルファベットの文字を返す。  
値が0以上25以下でなければ「アルファベットでない」を返す  
例: 文字(4)は「e」を、文字(23)は「x」を返す  
文字(-1)や文字(27)は「アルファベットでない」を返す



## 【参考】作成したシーザー暗号問題の解答方法

短冊型コード選択肢

「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

```
Imagine there's no heaven it's easy if you try no hell below us above us only sky imagine all the people living for today...  
プログラムの実行が完了しました。
```

送信