

第3問 （選択問題） 次の文章を読み、後の問い(問1～3)に答えよ。(配点 35)

Kさんは、あみだくじを表示するプログラムを作ろうと考えた。どの文字も同じ幅で表示されることを仮定して、記号の「**I**」・「**┣**」・「**┫**」という文字と改行を使うことにした。文字の左右および行間に隙間のない表示をすれば、これらの記号がつながって、あみだくじの線に見える。

あみだくじには縦線が2本以上、横線が1本以上ある。プログラムを簡単にするため、横線は隣り合う縦線の間のみを結ぶとし、一つの行にはちょうど1本だけ横線があるとした。

例えば、縦線が3本で横線が4本であるあみだくじを、図1のように4行で表示する。この図で点線は文字の枠を示しており、各行の右端で改行している。このあみだくじの一番上の横線は左から2本目と3本目の縦線を結んでおり、「**I**」・「**┣**」・「**┫**」と改行をこの順に表示することで1行目を出力できる。上から2番目の横線は左から1本目と2本目の縦線を結んでおり、1行目の表示に続けて「**┣**」・「**┫**」・「**I**」と改行をこの順に表示することで2行目を出力できる。3行目と4行目も同様である。

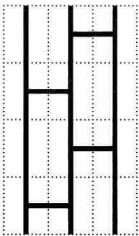


図1 表示されるあみだくじの例

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

表示したいあみだくじを指定するために、縦線の本数を変数 **tate** に、横線の位置の情報を整数の配列 **Yokosen** に、横線の本数を変数 **yoko** に入れることにした。配列の要素 **Yokosen[y]** が **x** であることは、上から **y** 番目の横線が左から **x** 番目の縦線と **x + 1** 番目の縦線を結ぶことを表す。

例えば、図1のあみだくじを表示するには、**tate** を **ア** , **yoko** を4と設定し、**Yokosen[1] ← 2** , **Yokosen[2] ← 1** , **Yokosen[3] ← イ** , **Yokosen[4] ← ウ** と設定する。以下では、配列の要素の並びを[ ]でくくって配列全体を表すことがある。例えば、上記のように設定された **Yokosen** は[2, 1, イ, ウ]と表せる。

このように **tate** , **yoko** , **Yokosen** が設定されているとき、あみだくじを表示する手続きとして、図2を作成した。

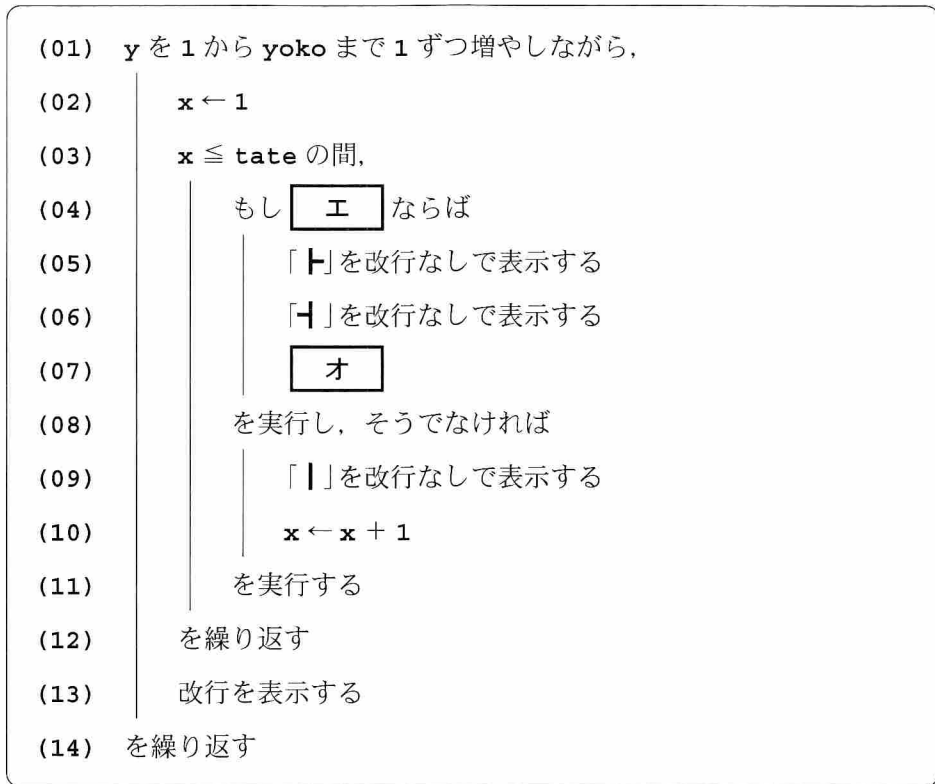


図 2 あみだくじを表示する手続き

<span style="border: 1px solid black; padding: 2px;">工</span> の解答群		
① Yokosen[x] ≠ y	① Yokosen[x] = y	② Yokosen[x] < y
③ Yokosen[y] ≠ x	④ Yokosen[y] = x	⑤ Yokosen[y] < x
⑥ Yokosen[x] ≠ x	⑦ Yokosen[x] = x	⑧ Yokosen[x] < x

オ

 の解答群  

① $x \leftarrow 0$	① $x \leftarrow x + 1$	② $x \leftarrow x + 2$	③ $tate \leftarrow tate - 1$
④ $x \leftarrow y$	⑤ $x \leftarrow x - 1$	⑥ $x \leftarrow x - 2$	⑦ $tate \leftarrow tate - 2$

情報関係基礎

問 2 次の文章を読み、空欄 **カ** ～ **ケ** 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。

K さんは次に、あみだくじを引いた結果をコンピュータで求めることを考えた。まず、あみだくじの縦線のそれぞれの上端にコマを置く。コマを区別するため、それぞれに番号をつけておく。すべてのコマを同時に、縦線に沿って下に移動していき、横線があったら、横線がつなぐ二つの縦線の上にあるコマを入れ替えれば、あみだくじの結果を求めることができる(図 3)。

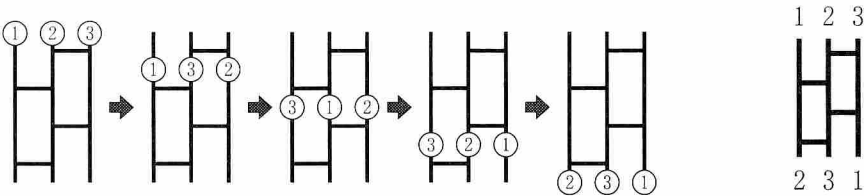


図 3 あみだくじの結果を求める様子

図 4 図 5 の手続きが表示するあみだくじ

コマの番号を順番に格納した配列 **Koma** が与えられ、**Koma** には最初、あみだくじの上端に置くコマの番号が左から順に格納されているものとする。すなわち、**Koma** の要素数とあみだくじの縦線の本数は等しい。

できた手続きを図 5 ～ 7 に示す。ここで用いている関数「要素数」はあらかじめ用意されたもので、配列を与えるとその要素数を返す。例えば **Koma** が [1, 2, 3] のとき、要素数(**Koma**) は 3 を返す。図 6 と図 7 はここで用いる新しい関数の定義である。関数を定義するときは、「関数」というキーワードと空白に続いて、関数名と、() でくくられた引数列を書き、「を」と「と定義する」までの間に関数の本体を書く。関数を呼び出すときは、関数名に続けて引数列を() でくくって書く。例えば、図 5 の (01) 行目は関数「配列を表示する」を呼び出しており、これを実行すると、図 6 の (02) ～ (05) 行目が実行される。

図 5 では、まず図 6 で定義した関数「配列を表示する」を呼び出し、最初に与えられた **Koma** を表示する。次に図 7 で定義した関数「あみだくじを表示する」を呼び出す。ここでは **カ** の値が **tate** に格納されて、図 2 と同じ処理をすることであみだくじを表示する。図 5 の (03) ～ (07) 行目では、コマを入れ替えることによって、あみだくじの結果を求めている。最後に (08) 行目で再び関数「配列を表示する」を呼び出して結果を表示している(図 4)。

- (01) 配列を表示する (Koma)
- (02) あみだくじを表示する ( 力 , Yokosen, 要素数 (Yokosen) )
- (03)  $y$  を 1 から 要素数 (Yokosen) まで 1 ずつ増やしなが
- (04) |  $t \leftarrow \text{Koma}[\text{Yokosen}[y]]$
- (05) | キ
- (06) | ク
- (07) を繰り返す
- (08) 配列を表示する (Koma)

図 5 あみだくじの結果を求める手続き

- (01) 関数 配列を表示する (Koma) を
- (02) |  $j$  を 1 から 要素数 (Koma) まで 1 ずつ増やしなが
- (03) | | ケ を改行なしで表示する
- (04) | を繰り返す
- (05) | 改行を表示する
- (06) と定義する

図 6 関数「配列を表示する」の定義

- (01) 関数 あみだくじを表示する (tate, Yokosen, yoko) を
- (02-15) | (図 2 と同じ)
- (16) と定義する

図 7 関数「あみだくじを表示する」の定義

<span style="border: 1px solid black; padding: 0 5px;">力</span> , <span style="border: 1px solid black; padding: 0 5px;">ケ</span> の解答群	
① 0	② 要素数 (Yokosen)
③ $j$	④ $\text{Yokosen}[j]$
⑥ yoko	⑦ $\text{Koma}[j]$
	⑧ 要素数 (Koma) - $j + 1$

<span style="border: 1px solid black; padding: 0 5px;">キ</span> ・ <span style="border: 1px solid black; padding: 0 5px;">ク</span> の解答群	
① $\text{Koma}[\text{Yokosen}[y + 1]] \leftarrow \text{Koma}[\text{Yokosen}[y]]$	
② $\text{Koma}[\text{Yokosen}[y]] \leftarrow \text{Koma}[\text{Yokosen}[y] + 1]$	
③ $\text{Koma}[\text{Yokosen}[y] + 1] \leftarrow t$	④ $\text{Koma}[\text{Yokosen}[y]] \leftarrow t$
⑤ $t \leftarrow \text{Koma}[\text{Yokosen}[y + 1]]$	⑥ $t \leftarrow \text{Koma}[\text{Yokosen}[y]]$

## 情報関係基礎

問 3 次の文章を読み、空欄 **コ** ～ **ト** 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。ただし、同じものを繰り返し選んでもよい。

K さんが手続きを作るのを見ていた M さんは、昇順でないコマの並びを昇順に並べ替えるあみだくじを表示する手続きを作ることにした。配列 **Koma** の隣り合う要素の大小関係が逆転しているときに、これらを入れ替えればよいと考えて、図 8 の手続きを作った。この手続きでは、図 6 で定義された関数「配列を表示する」と図 7 で定義された関数「あみだくじを表示する」を用いている。なお、配列 **Yokosen** は十分な大きさを持ち、全要素が 0 で初期化されていると仮定する。

M さんが作った図 8 の手続きについて、K さんは具体的な例として **Koma** に [5, 2, 4, 3, 1] を入れた場合の動作を観察した。このとき表 1 を用意して、図 8 の (09) 行目の直後における **p**, **q**, **Yokosen**, **Koma** を記録し、その変化を見ることで実行の様子を追いかけた。最終的には図 9 のあみだくじが表示された。

```
(01) 配列を表示する (Koma)
(02) yoko ← 0
(03) p を 1 から 要素数 (Koma) - 1 まで 1 ずつ増やしながら、
(04)   q を 1 から 要素数 (Koma) - p まで 1 ずつ増やしながら、
(05)     もし Koma [q] > Koma [q + 1] ならば
(06)       (Koma [q] と Koma [q + 1] を入れ替える手続き)
(07)       yoko ← yoko + 1
(08)       Yokosen [yoko] ← q
(09)     を実行する
(10)   を繰り返す
(11) を繰り返す
(12) あみだくじを表示する ( 力 , Yokosen, yoko)
(13) 配列を表示する (Koma)
```

図 8 昇順に並べ替えるあみだくじを作って表示する手続き

表1 図8の手続き(09)行目の直後における  
p, q, Yokosen, Koma の値

p	q	Yokosen	Koma
1	1	1, 0, 0, ...	2, 5, 4, 3, 1
1	2	1, 2, 0, ...	コ
1	3		サ
1	4		シ
2	1		ス
2	2		
2	3		
3	1		
3	2		
4	1		

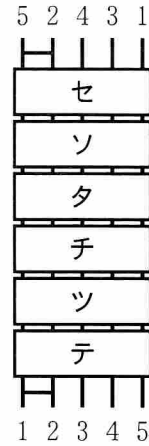


図9 図8の手続き  
が表示するあみ  
だくじ

図8の(04)行目の要素数(Koma) - p を要素数(Koma) - 1 としても同じあみだくじが表示される。しかし、の処理の結果としてp 番目に大きな値の位置(添字)が決まることで、Koma の要素が大きな値順にp 番目まで決まるため、要素数(Koma) - p まで繰り返せば十分である。

コ

ス

の解答群

①

2, 4, 3, 1, 5

②

2, 4, 5, 3, 1

③

2, 5, 4, 3, 1

④

4, 2, 3, 1, 5

⑤

5, 4, 2, 3, 1

セ

～

テ

の解答群

①

③

①

④

②

⑤

ト

の解答群

①

(06) 行目のみ

②

(07) 行目のみ

③

(08) 行目のみ

④

(04) ～ (10) 行目の繰り返し

⑤

(05) ～ (09) 行目の条件分岐

⑥

(06) ～ (08) 行目