

2013 年度 プログラミング II 練習問題 10

学籍番号： _____

氏名： _____

質問がある場合は、以下の注意事項を守り、

提出期限： 2014 年 1 月 8 日 (水) 16:30 までに

提出場所： 理学部 B 棟 2 階 幸山研究室 (B231 室)

の前に設置してある「質問箱 (提出用)」に、質問事項を記入し提出してください。
2 日～3 日後を目処に質問に対する回答を行い、「質問箱 (返却用)」に返却いたします。

注意事項：

- (1) 練習問題の提出は必修ではありません。質問がある場合のみ提出してください。
- (2) このページを印刷し、学籍番号・氏名を記入の上、質問事項の表紙として提出してください。
- (3) 解答は式の羅列ではなく、文章になるように記述してください。解答例を参照のこと。
- (4) 質問内容は明確に記入してください。

なお、授業に対する意見・質問等も受け付けています (成績には一切影響しません)。

授業に対する意見・質問等 (記入欄)：

出題者： 幸山 直人
出題日： 2013 年 12 月 14 日 (土)

問 1 「ハノイの塔 (再帰的解法)」 (`exercise1001.c`) について、以下の (1)～(3) の問いに答えなさい。なお、ソースプログラム「`exercise1001.c`」をコンパイルし、実行した結果は以下のとおりである (円盤の枚数が 3 枚の場合)。

```
z:¥src>exercise1001.exe Enter
```

```
円盤 1 を A から C に移す  
円盤 2 を A から B に移す  
円盤 1 を C から B に移す  
円盤 3 を A から C に移す  
円盤 1 を B から A に移す  
円盤 2 を B から C に移す  
円盤 1 を A から C に移す
```

```
z:¥src>
```

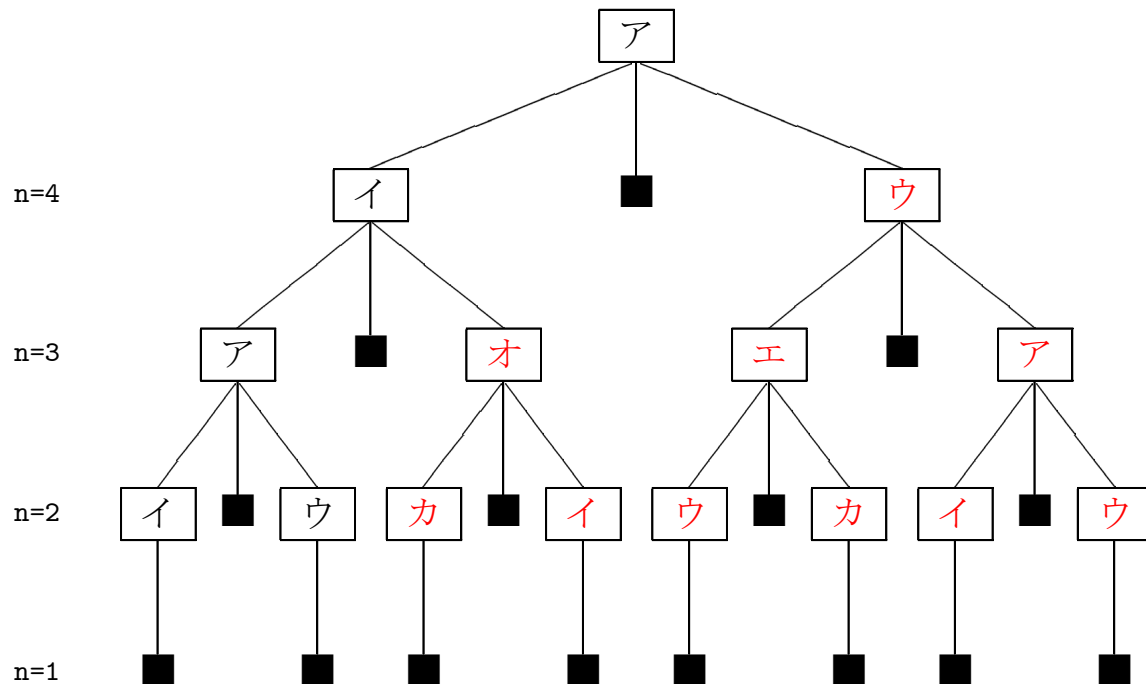
(1) 以下の「ハノイの塔 (再帰的解法)」 (`exercise1001.c`) を完成しなさい。

● ハノイの塔 (再帰的解法)

`exercise1001.c`

```
1: #include <stdio.h>  
2:  
3: void hanoi(int n, char x, char y, char z);  
4:  
5: int main(void)  
6: {  
7:     int n = 3;  
8:  
9:     hanoi(n, 'A', 'B', 'C');  
10:  
11:     return 0;  
12: }  
13:  
14: void hanoi(int n, char x, char y, char z)  
15: {  
16:     if (n > 1) hanoi(n - 1, x, z, y);  
17:     printf("円盤 %d を %c から %c に移す ¥n", n, x, z);  
18:     if (n > 1) hanoi(n - 1, y, x, z);  
19: }
```

(2) 以下の図「関数の再帰的呼び出しの模式図」は、(1)のプログラムを実行した際に関数「`hanoi()`」が再帰的に呼び出されている様子を模式図にしたものである。空欄 にあてはまる適切な関数を選択肢から選び、ア～カの記号で答え、図を完成しなさい。ただし、円盤の枚数は4枚とし、プログラムは上から下へ、右から左へ実行されるものとする。なお、図中の ■ は(1)のプログラムの第17行の関数「`printf()`」による出力を表す。



選択肢：

ア. `hanoi(n - 1, 'A', 'B', 'C')`

ウ. `hanoi(n - 1, 'B', 'A', 'C')`

オ. `hanoi(n - 1, 'C', 'A', 'B')`

イ. `hanoi(n - 1, 'A', 'C', 'B')`

エ. `hanoi(n - 1, 'B', 'C', 'A')`

カ. `hanoi(n - 1, 'C', 'B', 'A')`

(3) 以下は、円盤の枚数が5枚の場合の(1)のプログラムの実行結果である。空欄を適切な語句で埋め、実行結果を完成しなさい。

```
z:¥src>exercise1001.exe Enter
```

```
円盤 1 を A から C に移す  
円盤 2 を A から B に移す  
円盤 1 を C から B に移す  
円盤 3 を A から C に移す  
円盤 1 を B から A に移す  
円盤 2 を B から C に移す  
円盤 1 を A から C に移す  
円盤 4 を A から B に移す  
円盤 1 を C から B に移す  
円盤 2 を C から A に移す  
円盤 1 を B から A に移す  
円盤 3 を C から B に移す  
円盤 1 を A から C に移す  
円盤 2 を A から B に移す  
円盤 1 を C から B に移す  
円盤 5 を A から C に移す  
円盤 1 を B から A に移す  
円盤 2 を B から C に移す  
円盤 1 を A から C に移す  
円盤 3 を B から A に移す  
円盤 1 を C から B に移す  
円盤 2 を C から A に移す  
円盤 1 を B から A に移す  
円盤 4 を B から C に移す  
円盤 1 を A から C に移す  
円盤 2 を A から B に移す  
円盤 1 を C から B に移す  
円盤 3 を A から C に移す  
円盤 1 を B から A に移す  
円盤 2 を B から C に移す  
円盤 1 を A から C に移す
```

```
z:¥src>
```

問 2 以下のソースプログラム「ハノイの塔 (直接的解法)」(exercise1002.c) はハノイの塔を直接的に解くことができるプログラムである (円盤の枚数が 5 枚の場合)。「ハノイの塔 (再帰的解法)」(exercise1001.c) と実行結果が同じになることを確かめなさい。また、ソースプログラム「exercise1002.c」をトレースしなさい。

● ハノイの塔 (直接的解法)

exercise1002.c

```
1: #include <stdio.h>
2:
3: #define N 64
4:
5: int main(void)
6: {
7:     int n = 5;
8:     int x[3][N];
9:     int p[3];
10:    int i, j;
11:    char c[3];
12:
13:    //円盤の並びを設定
14:    for (i = 0; i < n; i++) {
15:        x[0][i] = n - i;
16:        x[1][i] = 0;
17:        x[2][i] = 0;
18:    }
19:
20:    //最上位の円盤の位置
21:    p[0] = n - 1;
22:    p[1] = -1;
23:    p[2] = -1;
24:
25:    //棒の位置
26:    if (n % 2 == 1) {
27:        c[0] = 'A';
28:        c[1] = 'B';
29:        c[2] = 'C';
30:    }
31:    else {
32:        c[0] = 'A';
33:        c[1] = 'C';
34:        c[2] = 'B';
35:    }
```

```

36:
37:     i = 0;
38:     while (x[2][n - 1] != 1 && x[1][n - 1] != 1) {
39:         i = i % 3;
40:         if (p[i] != -1) {
41:             if (x[i][p[i]] % 2 == 1) {
42:                 j = (i + 2) % 3;
43:                 if (p[j] == -1 || x[i][p[i]] < x[j][p[j]]) {
44:                     printf("円盤 %d を %c から %c に移す✕n",
45:                             x[i][p[i]], c[i], c[j]);
46:                     x[j][p[j] + 1] = x[i][p[i]];
47:                     x[i][p[i]] = 0;
48:                     p[i] = p[i] - 1;
49:                     p[j] = p[j] + 1;
50:                     i = i + 1;
51:                 }
52:                 else {
53:                     i = i + 2;
54:                 }
55:             }
56:             else {
57:                 j = (i + 1) % 3;
58:                 if (p[j] == -1 || x[i][p[i]] < x[j][p[j]]) {
59:                     printf("円盤 %d を %c から %c に移す✕n",
60:                             x[i][p[i]], c[i], c[j]);
61:                     x[j][p[j] + 1] = x[i][p[i]];
62:                     x[i][p[i]] = 0;
63:                     p[i] = p[i] - 1;
64:                     p[j] = p[j] + 1;
65:                     i = i + 2;
66:                 }
67:                 else {
68:                     i = i + 2;
69:                 }
70:             }
71:         }
72:         else {
73:             i = i + 2;
74:         }
75:     }
76:
77:     return 0;
78: }

```