

2011 年度 情報科学&情報科学演習 レポート 5

学生用

学籍番号：_____

氏名：_____

下記の注意事項を守り、次ページ以降の問いに答え、レポートを完成させなさい。

提出期限：2011 年 5 月 31 日 (火) 13:00 まで

提出場所：理学部棟 正面玄関内に設置のレポートボックス

注意事項：

- (1) このページを印刷し、必要事項を記入の上 (学籍番号欄と氏名欄は2箇所あるので忘れずに記入すること)、レポートの表紙として提出すること。
- (2) 文章処理ソフトウェアや図形処理ソフトウェア等を駆使してレポートを作成し (問→解答→問→解答→…の順になるように記述すること)、A4 サイズの用紙に印刷して提出すること (手書きは不可)。
- (3) クラスメイトのレポートを参考にしたり、クラスメイトと協力してレポートを作成した場合は、教員控の協力者氏名欄にクラスメイトの氏名を記入すること。これらの場合も、自分の言葉で表現し直すこと。**コピー禁止**。
- (4) 情報科学&情報科学演習について、あなたの声を聞かせてください (教員控の意見・質問欄に記入のこと)。気軽にどうぞ (成績には一切影響しません)。

出題者：幸山 直人

出題日：2011 年 5 月 25 日 (水)

得点：

/6

----- 切り取り線 -----

2011 年度 情報科学&情報科学演習 レポート 5

教員控

学籍番号：_____

氏名：_____

協力者氏名：_____, _____, _____

レポート作成に要した時間：_____ 時間

得点：

/6

意見・質問：

問 1 演習 3 の **演習 2** で作成した「アルファベットの小文字は大文字に大文字は小文字に変換して表示するプログラム」(ファイル名「`exercise0306.c`」)を印刷してレポートに添付しなさい。(2 点)

解答例 添付書類を参考のこと。

評価基準 解答例に準じた解答であれば 2 点。

問 2 演習 3 の **演習 3** で作成した「 3×3 行列 A, B の積を計算するプログラム」(ファイル名「`exercise0307.c`」)を印刷してレポートに添付しなさい。(2 点)

解答例 添付書類を参考のこと。

評価基準 解答例に準じた解答であれば 2 点。

問 3 演習 3 の **演習 5** で作成した「ハノイの塔ー直接的解法ー(偶数個の円盤)」(ファイル名「`exercise0309.c`」)を印刷してレポートに添付しなさい。(2 点)

解答例 添付書類を参考のこと。

評価基準 解答例に準じた解答であれば 2 点。

問 1 の解答例 「アルファベットの小文字は大文字に大文字は小文字に変換して表示するプログラム」(ファイル名「`exercise0306.c`」)

```
1: #include <stdio.h>
2: #include <ctype.h>
3:
4: int main(void)
5: {
6:     int ch;
7:
8:     while ((ch = getchar()) != EOF) {
9:         if ((64 < ch) && (ch < 91)) {          ← 論理積を用いる
10:             ch = tolower(ch);
11:         }
12:         else if ((96 < ch) && (ch < 123)) {    ← 論理積を用いる
13:             ch = toupper(ch);
14:         }
15:         putchar(ch);
16:     }
17:
18:     return 0;
19: }
```

問 2 の解答例 (その 1) 「3×3 行列 A, B の積を計算するプログラム」 (ファイル名「`exercise0307.c`」)

【標準的な解答例】 第 14 行から第 18 行で `c[i][j]` の全ての値を 0 に初期化

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     int i, j, k;
6:     double a[3][3] = {{1, 2, 3},
7:                        {4, 5, 6},
8:                        {7, 8, 9}};
9:     double b[3][3] = {{1, 2, 3},
10:                       {4, 5, 6},
11:                       {7, 8, 9}};
12:     double c[3][3];
13:
14:     for (i = 0; i < 3; i++) {
15:         for (j = 0; j < 3; j++) {
16:             c[i][j] = 0;
17:         }
18:     }
19:
20:     for (i = 0; i < 3; i++) {
21:         for (j = 0; j < 3; j++) {
22:             for (k = 0; k < 3; k++) {
23:                 c[i][j] = c[i][j] + a[i][k] * b[k][j];
24:             }
25:             printf("%8.2f", c[i][j]);
26:         }
27:         printf("\n");
28:     }
29:
30:     return 0;
31: }
```

* 行列のサイズが小さければ下記のように初期化してもよい (第 14 行から第 18 行は不要)。ただし、数値計算などの分野では、比較的サイズの大きな行列を扱うため、上記のように初期化する。

```
12:     double c[3][3] = {{0, 0, 0},
13:                        {0, 0, 0},
14:                        {0, 0, 0}};
```

問 2 の解答例 (その 2) 「3×3 行列 A, B の積を計算するプログラム」 (ファイル名「`exercise0307d.c`」)

【Excellent な解答例】 第 16 行のみで `c[i][j]` の全ての値を 0 に初期化

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     int i, j, k;
6:     double a[3][3] = {{1, 2, 3},
7:                        {4, 5, 6},
8:                        {7, 8, 9}};
9:     double b[3][3] = {{1, 2, 3},
10:                       {4, 5, 6},
11:                       {7, 8, 9}};
12:     double c[3][3];
13:
14:     for (i = 0; i < 3; i++) {
15:         for (j = 0; j < 3; j++) {
16:             c[i][j] = 0;           ← ココに入れると上手く初期化できる
17:             for (k = 0; k < 3; k++) {
18:                 c[i][j] = c[i][j] + a[i][k] * b[k][j];
19:             }
20:             printf("%8.2f", c[i][j]);
21:         }
22:         printf("\n");
23:     }
24:
25:     return 0;
26: }
```



```

42:             if (j == 0) t = 'A';
43:             if (j == 1) t = 'C';           ← 変更箇所
44:             if (j == 2) t = 'B';           ← 変更箇所
45:             printf("円盤 %d を %c から %c に移す\n", x[i][k], s, t);
46:             x[j][l+1] = x[i][k];
47:             x[i][k] = 0;
48:             y[i] = y[i] - 1;
49:             y[j] = y[j] + 1;
50:         }
51:     }
52: }
53:     i = i + 2;
54:     i = i % 3;
55: }
56:
57:     return 0;
58: }

```

問 3 の解答例 (その 2) 「ハノイの塔ー直接的解法ー (偶数個の円盤)」 (ファイル名「**exercise0309d.c**」)

【回転方向を反時計回りに変更した解答例】ただし、棒 B と棒 C はそのまま (入れ替えなし)

```

1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     char s, t;
6:     int i = 0, j, k, l, m;
7:     int x[3][5] = {{1000, 4, 3, 2, 1},           ← 変更箇所
8:                   {1000, 0, 0, 0, 0},           ← 変更箇所
9:                   {1000, 0, 0, 0, 0}};           ← 変更箇所
10:    int y[3] = {4, 0, 0};                          ← 変更箇所
11:
12:    while (x[2][4] != 1) {                            ← 変更箇所
13:        if (y[i] != 0) {
14:            k = y[i];
15:            m = x[i][k] % 2;
16:            if (m == 1) {
17:                j = (i + 1) % 3;                        ← 変更箇所
18:                l = y[j];
19:                if (x[i][k] < x[j][l]) {
20:                    if (i == 0) s = 'A';
21:                    if (i == 1) s = 'B';
22:                    if (i == 2) s = 'C';

```

```

23:             if (j == 0) t = 'A';
24:             if (j == 1) t = 'B';
25:             if (j == 2) t = 'C';
26:             printf("円盤 %d を %c から %c に移す\n", x[i][k], s, t);
27:             x[j][l+1] = x[i][k];
28:             x[i][k] = 0;
29:             y[i] = y[i] - 1;
30:             y[j] = y[j] + 1;
31:             i = i + 1;
32:             i = i % 3;
33:         }
34:     }
35:     else {
36:         j = (i + 2) % 3;
37:         l = y[j];
38:         if (x[i][k] < x[j][l]) {
39:             if (i == 0) s = 'A';
40:             if (i == 1) s = 'B';
41:             if (i == 2) s = 'C';
42:             if (j == 0) t = 'A';
43:             if (j == 1) t = 'B';
44:             if (j == 2) t = 'C';
45:             printf("円盤 %d を %c から %c に移す\n", x[i][k], s, t);
46:             x[j][l+1] = x[i][k];
47:             x[i][k] = 0;
48:             y[i] = y[i] - 1;
49:             y[j] = y[j] + 1;
50:         }
51:     }
52: }
53: i = i + 1;
54: i = i % 3;
55: }
56:
57: return 0;
58: }

```

← 変更箇所

← 変更箇所

← 変更箇所