

# 2015年度 プログラミングII レポート07

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： 2015年12月1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) コンピュータ端末室を利用する場合は、情報システム利用ガイドラインを厳守すること。**特に、コンピュータ端末室では飲食禁止である。**
- (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) プログラミングII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： 2015年11月18日(水)

----- 切り取り線 -----

# 2015年度 プログラミングII レポート07

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

意見・質問：

**問 1** p.216 の記述にしたがって、ソースプログラム「ポインタの宣言と利用」(rei6.1a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
注意：コピー禁止。自らの手で全てのソースプログラムを入力すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 2** p.218 の記述にしたがって、ソースプログラム「ポインタの宣言と利用」(rei6.1b.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
注意：コピー禁止。自らの手で全てのソースプログラムを入力すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 3** p.224 の記述にしたがって、ソースプログラム「合計と平均を求める」(rei6.2a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
注意：コピー禁止。自らの手で全てのソースプログラムを入力すること。  
注意：for 文の式 1 の total = 0 は繰り返しの変数に直接関係しないので、for 文の外に記述すること。  
注意：for 文の式 3 の pScore++ は繰り返しの変数に直接関係しないので、for 文の内に記述すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。なお、ソースプログラム「合計と平均を求める」(rei6.2a.k.c) は注意事項にしたがって記述されたソースプログラムである。

**問 4** p.228 の記述にしたがって、ソースプログラム「文字列の複写 (コピー)」(rei6.3a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
注意：コピー禁止。自らの手で全てのソースプログラムを入力すること。  
注意：第 17 行の for 文は while 文で記述した方が自然である。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。なお、ソースプログラム「文字列の複写 (コピー)」(rei6.3a.k.c) は注意事項にしたがって記述されたソースプログラムである。

**問 5** p.231 の記述にしたがって、ソースプログラム「合計と平均を求める」(rei6.2b.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
注意：for 文の式 1 の total = 0 は繰り返しの変数に直接関係しないので、for 文の外に記述すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 6** p.233 の記述にしたがって、ソースプログラム「合計と平均を求める」(rei6\_2c.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
**注意** : for 文の式 1 の `total = 0` は繰り返しの変数に直接関係しないので、for 文の外に記述すること (2つ)。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 7** p.237 の記述にしたがって、ソースプログラム「合計と平均を求める」(rei6\_4a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
**注意** : for 文の式 1 の `total = 0` は繰り返しの変数に直接関係しないので、for 文の外に記述すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 8** p.241 の記述にしたがって、ソースプログラム「合計と平均を求める」(rei6\_5a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
**注意** : for 文の式 1 の `total = 0` と `cnt = 0` は繰り返しの変数に直接関係しないので、for 文の外に記述すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 9** p.244 の記述にしたがって、ソースプログラム「ポインタの宣言と利用」(rei6\_6a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
**注意** : コピー禁止。自らの手で全てのソースプログラムを入力すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 10** p.247 の記述にしたがって、ソースプログラム「合計と平均を求める」(rei6\_5b.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。  
**注意** : for 文の式 1 の `total = 0` と `cnt = 0` は繰り返しの変数に直接関係しないので、for 文の外に記述すること。  
**注意** : for 文の式 3 の `pp++` と `ppName++` は繰り返しの変数に直接関係しないので、for 文の内に記述すること (`pp++` は 2つ)。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。なお、ソースプログラム「合計と平均を求める」(rei6\_5b\_k.c) は注意事項にしたがって記述されたソースプログラムである。

**問 11** p.248 の記述にしたがって、ソースプログラム「合計と平均を求める」(rei6\_5c.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。

**注意** : for 文の式 1 の `total = 0` と `cnt = 0` は繰り返しの変数に直接関係しないので、for 文の外に記述すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 12** p.253 の記述にしたがって、ソースプログラム「合計と平均を求める」(rei6\_2d.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。

**注意** : for 文の式 1 の `total = 0` は繰り返しの変数に直接関係しないので、for 文の外に記述すること。

**注意** : for 文の式 3 の `pp++` は繰り返しの変数に直接関係しないので、for 文の内に記述すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。なお、ソースプログラム「合計と平均を求める」(rei6\_2d.k.c) は注意事項にしたがって記述されたソースプログラムである。

**参考** : p.255 の記述にしたがって、ソースプログラム「基本選択法」(rei6\_7a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。

**参考** : p.260 の記述にしたがって、ソースプログラム「基本選択法」(rei6\_7b.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。

**問 13** p.263 の記述にしたがって、ソースプログラム「文字列管理」(rei6\_8a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。

**注意** : for 文の式 1 の `p[0] = str` は繰り返しの変数に直接関係しないので、for 文の外に記述すること。

**注意** : 第 43 行の for 文は while 文で記述した方が自然である。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。なお、ソースプログラム「文字列管理」(rei6\_8a.k.c) は注意事項にしたがって記述されたソースプログラムである。

**問 14** p.267 の記述にしたがって、ソースプログラム「コマンドライン引数による四則演算」(rei6\_9a.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 15** p.269 の記述にしたがって、ソースプログラム「コマンドライン引数による四則演算」(rei6\_9b.c)を作成しなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。

**解答例** 著作権保護のため解答を記述していません。配布済みのファイルを参照してください。

**問 16** ソースプログラム「最大公約数(関数: 戻り値あり)」(report06\_04.c)を書き換え、下記ソースプログラムに続けて値の受け渡しにポインタを用いたソースプログラム「最大公約数(関数: 戻り値あり: ポインタ)」(report07\_01.c)を完成させなさい。さらに、ソースプログラムをコンパイルし、実行可能ファイルが正しく動作するか確認しなさい。なお、作成したソースプログラムは印刷してレポートに添付すること。

**ヒント:** ソースプログラム「文字列管理」(rei6\_8a.c)の関数「\*getNext()」を参考にしなさい。

● 最大公約数(関数: 戻り値あり: ポインタ)

report07\_01.c

```
1: #include <stdio.h>
2:
3: int *gcd(int *px, int *py);
4:
5: int main(void)
6: {
7:     int x = 1234, y = 56;
8:     int *px, *py;
9:
10:    px = &x;
11:    py = &y;
12:    printf("gcd(%d,%d)=", *px, *py);
13:    printf("%d\n", *gcd(px, py));
14:
15:    return 0;
16: }
```

**解答例** 別紙を参照のこと。

問 16 の解答例 「最大公約数 (関数：戻り値あり：ポインタ)」 (report07\_01.c)

```

:           ----- 続き -----

17:
18: int *gcd(int *px, int *py)
19: {
20:     int tmp;
21:
22:     while (*py != 0) {
23:         tmp = *px;
24:         *px = *py;
25:         *py = tmp % *px;
26:     }
27:
28:     return px;
29: }
```

考察：ポインタが変数のアドレスを渡すことに注意すれば、「最大公約数 (関数：戻り値あり：ポインタ)」 (report07\_01.c) は以下のように記述してもよい (戻り値「return 文」を用いない)。

「最大公約数 (関数：戻り値なし：ポインタ)」 (report07\_01a.c)

```

1: #include <stdio.h>
2:
3: void gcd(int *px, int *py);
4:
5: int main(void)
6: {
7:     int x = 1234, y = 56;
8:     int *px, *py;
9:
10:    px = &x;
11:    py = &y;
12:    printf("gcd(%d,%d)=", *px, *py);
13:    gcd(px, py);
14:    printf("%d\n", *px);
15:
16:    return 0;
17: }
18:
19: void gcd(int *px, int *py)
20: {
21:     int tmp;
22:
23:     while (*py != 0) {
24:         tmp = *px;
25:         *px = *py;
26:         *py = tmp % *px;
27:     }
28: }
```

**考察**：グローバル変数や配列を用いなくても、変数のアドレスを直接参照できるポインタを用いれば関数間（呼び出す側の関数と呼び出される側の関数）で**複数の変数を用いて複数の値のやり取りができる**。したがって、戻り値「return 文」のように呼び出す側から参照できる呼び出される側の変数が高々1つであるという制限に縛られません。

「有理数体上の四則演算（関数：戻り値なし：ポインタ）」(report07\_01b.c)

```
1: #include <stdio.h>
2:
3: void add_q(int a, int b, int c, int d, int *px, int *py);
4: void sub_q(int a, int b, int c, int d, int *px, int *py);
5: void mul_q(int a, int b, int c, int d, int *px, int *py);
6: void div_q(int a, int b, int c, int d, int *px, int *py);
7:
8: int main(void)
9: {
10:     int a = 2, b = 3;
11:     int c = 1, d = 4;
12:     int x, y;
13:     int *px, *py;
14:
15:     px = &x;
16:     py = &y;
17:
18:     //和
19:     add_q(a, b, c, d, px, py);
20:     printf("(%d/%d)+(%d/%d)=(%d/%d)%n", a, b, c, d, *px, *py); ← ポインタ参照
21:
22:     //差
23:     sub_q(a, b, c, d, px, py);
24:     printf("(%d/%d)-(%d/%d)=(%d/%d)%n", a, b, c, d, x, y); ← 変数名でも参照可
25:
26:     //積
27:     mul_q(a, b, c, d, px, py);
28:     printf("(%d/%d)*(%d/%d)=(%d/%d)%n", a, b, c, d, *px, *py);
29:
30:     //商
31:     div_q(a, b, c, d, px, py);
32:     printf("(%d/%d)/(%d/%d)=(%d/%d)%n", a, b, c, d, x, y);
33:
34:     return 0;
35: }
36:
37: void add_q(int a, int b, int c, int d, int *px, int *py)
38: {
39:     *px = a * d + b * c; ← 実際には関数「main()」の変数「x」の値を書き換えている
40:     *py = b * d;         ← 実際には関数「main()」の変数「y」の値を書き換えている
41: }
42:
43: void sub_q(int a, int b, int c, int d, int *px, int *py)
44: {
45:     *px = a * d - b * c;
46:     *py = b * d;
47: }

```

⋮ ----- 以下、同様なので省略 -----