

2016年度 プログラミングII レポート06

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合わせに答え、レポートを完成させなさい。

提出期限 : 2016年11月15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) クラスマイトのレポートを参考にしたり、クラスマ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マ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 : 2016年11月9日(水)

2016年度 プログラミングII レポート06

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.155 の記述にしたがって、ソースプログラム「平均から評価を求める」(rei5_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。

注意：void を適切に記述すること。

注意：戻り値がないので関数「getLank()」の中で結果を表示する必要がある。

注意：間違えずに関数で書き換えたプログラムの方のソースプログラムを添付すること。

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2 p.159 の記述にしたがって、ソースプログラム「平均から評価を求める」(rei5_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：void を適切に記述すること。

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3 p.165 の記述にしたがって、ソースプログラム「平均から評価を求める」(rei5_2.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：void を適切に記述すること。

注意：グローバル変数を用いると関数「main()」と関数「getLank()」で同じ変数を共有できる。

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。

参考 : p.167 の記述にしたがって、ソースプログラム「平均から評価を求める」(rei5_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。

問4 p.171 の記述にしたがって、ソースプログラム「平均から評価を求める」(rei5_3.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5 p.176の記述にしたがって、ソースプログラム「平均から評価を求める」(rei5_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：戻り値の型に注意すること。

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.182の記述にしたがって、ソースプログラム「配列の部分から最小値を求める」(rei5_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：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8行に誤りがあるので訂正すること。

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。

参考：p.186の記述にしたがって、ソースプログラム「基本選択法」(rei5_4b.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7 p.193の記述にしたがって、ソースプログラム「階乗を求める」(rei5_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：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8 p.196の記述にしたがって、ソースプログラム「階乗を求める」(rei5_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：キャスト演算子(double)を忘れずに記述すること。

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。

参考：p.200の記述にしたがって、ソースプログラム「二分探索法」(rei5_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。

問9 レポート03のソースプログラム「有理数体上の四則演算」(report03_01.c)を書き換え、下記ソースプログラムに続けて加減乗除の各演算をそれぞれ関数にしたソースプログラム「有理数体上の四則演算(関数：戻り値なし)」(report06_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「平均から評価を求める」(rei5_1a.c)を参考に、4つの関数を作成すること。

● 有理数体上の四則演算(関数：戻り値なし)

report06_01.c

```
1: #include <stdio.h>
2:
3: void add_q(int a, int b, int c, int d);
4: void sub_q(int a, int b, int c, int d);
5: void mul_q(int a, int b, int c, int d);
6: void div_q(int a, int b, int c, int d);
7:
8: int main(void)
9: {
10:     int a = 2, b = 3;
11:     int c = 1, d = 4;
12:
13:     //和
14:     add_q(a, b, c, d);
15:
16:     //差
17:     sub_q(a, b, c, d);
18:
19:     //積
20:     mul_q(a, b, c, d);
21:
22:     //商
23:     div_q(a, b, c, d);
24:
25:     return 0;
26: }
```

考察：問9に習って、ソースプログラム「複素数体上の四則演算」(report03_02.c)の加減乗除の各演算をそれぞれ関数にしたソースプログラム「複素数体上の四則演算(関数：戻り値なし)」(report06_01a.c)を作成してみましょう。

解答例 別紙を参照のこと。

問 10 ソースプログラム「有理数体上の四則演算(関数：戻り値なし)」(report06_01.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グローバル変数を用いたソースプログラム「有理数体上の四則演算(関数：グローバル変数)」(report06_02.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「平均から評価を求める」(rei5_2.c)を参考に、4つの関数を作成すること。

● 有理数体上の四則演算(関数：グローバル変数)

report06_02.c

```
1: #include <stdio.h>
2:
3: int x, y;                                ←グローバル変数
4:
5: void add_q(int a, int b, int c, int d);
6: void sub_q(int a, int b, int c, int d);
7: void mul_q(int a, int b, int c, int d);
8: void div_q(int a, int b, int c, int d);
9:
10: int main(void)
11: {
12:     int a = 2, b = 3;
13:     int c = 1, d = 4;
14:
15:     //和
16:     add_q(a, b, c, d);
17:     printf("(%d/%d)+(%d/%d)=(%d/%d)\n", a, b, c, d, x, y);
18:
19:     //差
20:     sub_q(a, b, c, d);
21:     printf("(%d/%d)-(%d/%d)=(%d/%d)\n", a, b, c, d, x, y);
22:
23:     //積
24:     mul_q(a, b, c, d);
25:     printf("(%d/%d)*(%d/%d)=(%d/%d)\n", a, b, c, d, x, y);
26:
27:     //商
28:     div_q(a, b, c, d);
29:     printf("(%d/%d)/(%d/%d)=(%d/%d)\n", a, b, c, d, x, y);
30:
31:     return 0;
32: }
```

解答例 別紙を参照のこと。

問 11 ソースプログラム「有理数体上の四則演算(関数: グローバル変数)」(`report06_02.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配列を用いたソースプログラム「有理数体上の四則演算(関数: 配列)」(`report06_03.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：配列の受け渡しでは、実際には配列の先頭アドレスを渡すため、呼び出す側と呼び出される側で同じ配列が参照されることに注意する(`return`文を使う必要はない)。

● 有理数体上の四則演算(関数: 配列)

`report06_03.c`

```
1: #include <stdio.h>
2:
3: void add_q(int a[], int b[], int x[]);
4: void sub_q(int a[], int b[], int x[]);
5: void mul_q(int a[], int b[], int x[]);
6: void div_q(int a[], int b[], int x[]);
7:
8: int main(void)
9: {
10:     int a[2] = {2, 3};
11:     int b[2] = {1, 4};
12:     int x[2];
13:
14:     //和
15:     add_q(a, b, x);
16:     printf("(%d/%d)+(%d/%d)=(%d/%d)\n", a[0], a[1], b[0], b[1], x[0], x[1]);
17:
18:     //差
19:     sub_q(a, b, x);
20:     printf("(%d/%d)-(%d/%d)=(%d/%d)\n", a[0], a[1], b[0], b[1], x[0], x[1]);
21:
22:     //積
23:     mul_q(a, b, x);
24:     printf("(%d/%d)*(%d/%d)=(%d/%d)\n", a[0], a[1], b[0], b[1], x[0], x[1]);
25:
26:     //商
27:     div_q(a, b, x);
28:     printf("(%d/%d)/(%d/%d)=(%d/%d)\n", a[0], a[1], b[0], b[1], x[0], x[1]);
29:
30:     return 0;
31: }
```

解答例 別紙を参照のこと。

問 12 下記ソースプログラム「最大公約数(関数：戻り値あり)」(report06_04.c)は、ソースプログラム「最大公約数」(report05_02.c)を関数に書き換えたものである。これを使って、3変数 $x = 123$, $y = 456$, $z = 789$ の最大公約数を求めるソースプログラム「3変数の最大公約数」(report06_05.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。

ヒント：3変数の最大公約数 $\text{gcd}(x, y, z)$ は $\text{gcd}(x, \text{gcd}(y, z))$ によって与えられる(証明は各自で)。したがって、関数「`gcd()`」を変更する必要はありません。

● 最大公約数(関数：戻り値あり)

report06_04.c

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

解答例 別紙を参照のこと。

問 13 ソースプログラム「最大公約数(関数: 戻り値あり)」(`report06_04.c`)の関数「`gcd()`」が再帰呼び出しとなるように書き換えたソースプログラム「最大公約数(関数: 再帰呼び出し)」(`report06_06.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。

ヒント：ソースプログラム「階乗を求める」(`rei5_5a.c`)を参考に、再帰呼び出しとなるように関数「`gcd()`」を作成すること。

解答例 別紙を参照のこと。

問 14 ソースプログラム「有理数体上の四則演算(関数: 戻り値なし)」(`report06_01.c`)を実行すると下記の改良前のように既約でない有理数が現れる。ソースプログラム「最大公約数(関数)」(`report06_04.c`)の関数「`gcd()`」をソースプログラム「有理数体上の四則演算(関数: 戻り値なし)」(`report06_01.c`)に含めることによって、下記の改良後のように計算結果の値が既約となるソースプログラム「有理数体上の四則演算(関数: 戻り値なし: 最大公約数)」(`report06_07.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。

ヒント：ソースプログラム「最大公約数(関数)」(`report06_04.c`)の関数「`gcd()`」はそのまま使用する。

ヒント：計算の過程で関数「`gcd()`」の引数が負の整数になる場合があるので、標準ヘッダファイル「`math.h`」をインクルード(`#include <math.h>`)して、引数が正の整数となるように絶対値(absolute value)を求める関数「`int abs(int 変数名)`」を使用する。

改良前：

```
Z:¥src>report06_01.exe [Enter]
(2/3)+(1/4)=(11/12)
(2/3)-(1/4)=(5/12)
(2/3)*(1/4)=(2/12)                                     ←既約でない
(2/3)/(1/4)=(8/3)

Z:¥src>
```

改良後：

```
Z:¥src>report06_07.exe [Enter]
(2/3)+(1/4)=(11/12)
(2/3)-(1/4)=(5/12)
(2/3)*(1/4)=(1/6)                                     ←既約
(2/3)/(1/4)=(8/3)

Z:¥src>
```

解答例 別紙を参照のこと。

問9の解答例 「有理数体上の四則演算(関数: 戻り値なし)」(report06_01.c)

```
: ----- 続き -----  
27:  
28: void add_q(int a, int b, int c, int d)  
29: {  
30:     int x, y;  
31:  
32:     x = a * d + b * c;  
33:     y = b * d;  
34:     printf("(%d/%d)+(%d/%d)=(%d/%d)\n", a, b, c, d, x, y);  
35: }  
36:  
37: void sub_q(int a, int b, int c, int d)  
38: {  
39:     int x, y;  
40:  
41:     x = a * d - b * c;  
42:     y = b * d;  
43:     printf("(%d/%d)-(%d/%d)=(%d/%d)\n", a, b, c, d, x, y);  
44: }  
45:  
46: void mul_q(int a, int b, int c, int d)  
47: {  
48:     int x, y;  
49:  
50:     x = a * c;  
51:     y = b * d;  
52:     printf("(%d/%d)*(%d/%d)=(%d/%d)\n", a, b, c, d, x, y);  
53: }  
54:  
55: void div_q(int a, int b, int c, int d)  
56: {  
57:     int x, y;  
58:  
59:     x = a * d;  
60:     y = b * c;  
61:     printf("(%d/%d)/(%d/%d)=(%d/%d)\n", a, b, c, d, x, y);  
62: }
```

問 10 の解答例 「有理数体上の四則演算 (関数 : グローバル変数)」 (report06_02.c)

```
: ----- 続き -----  
33:  
34: void add_q(int a, int b, int c, int d)  
35: {  
36:     x = a * d + b * c;  
37:     y = b * d;  
38: }  
39:  
40: void sub_q(int a, int b, int c, int d)  
41: {  
42:     x = a * d - b * c;  
43:     y = b * d;  
44: }  
45:  
46: void mul_q(int a, int b, int c, int d)  
47: {  
48:     x = a * c;  
49:     y = b * d;  
50: }  
51:  
52: void div_q(int a, int b, int c, int d)  
53: {  
54:     x = a * d;  
55:     y = b * c;  
56: }
```

考察：問 10 に習って、ソースプログラム「複素数体上の四則演算 (関数 : 戻り値なし)」 (report06_01a.c) を書き換え、値の受け渡しにグローバル変数を用いたソースプログラム「複素数体上の四則演算 (関数 : グローバル変数)」 (report06_02a.c) を作成してみましょう。

問 11 の解答例 「有理数体上の四則演算 (関数：配列)」 (report06_03.c)

```
: ----- 続き -----  
32:  
34: void add_q(int a[], int b[], int x[])
35: {
36:     x[0] = a[0] * b[1] + a[1] * b[0];
37:     x[1] = a[1] * b[1];
38: }
39:  
40: void sub_q(int a[], int b[], int x[])
41: {
42:     x[0] = a[0] * b[1] - a[1] * b[0];
43:     x[1] = a[1] * b[1];
44: }
45:  
46: void mul_q(int a[], int b[], int x[])
47: {
48:     x[0] = a[0] * b[0];
49:     x[1] = a[1] * b[1];
50: }
51:  
52: void div_q(int a[], int b[], int x[])
53: {
54:     x[0] = a[0] * b[1];
55:     x[1] = a[1] * b[0];
56: }
```

注意：今回のソースプログラムでは配列の長さが 2 に固定されているため配列の終端を気にする必要はないが、一般的な配列による値の受け渡しでは配列の先頭アドレスしか渡さないため、プログラマの責任において配列の長さや終端値を設けるなど配列の終端を適切に処理できるようにプログラムを記述する必要がある（バッファオーバーフローの原因となる）。

考察：問 11 に習って、ソースプログラム「複素数体上の四則演算 (関数：グローバル変数)」(report06_02a.c) を書き換え、値の受け渡しに配列を用いたソースプログラム「複素数体上の四則演算 (関数：配列)」(report06_03a.c) を作成してみましょう。

問 12 の解答例 「3 変数の最大公約数 (関数 : 戻り値あり)」 (report06_05.c)

```
1: #include <stdio.h>
2:
3: int gcd(int x, int y);
4:
5: int main(void)
6: {
7:     int x = 123, y = 456, z = 789;
8:
9:     printf("gcd(%d,%d,%d)=%d\n", x, y, z, gcd(gcd(x, y), z));
10:
11:    return 0;
12: }
13:
14: int gcd(int x, int y)
15: {
16:     int tmp;
17:
18:     while (y != 0) {
19:         tmp = x;
20:         x = y;
21:         y = tmp % x;
22:     }
23:
24:     return x;
25: }
```

問 13 の解答例 「最大公約数 (関数 : 再帰呼び出し)」 (report06_06.c)

: ----- 第 1 行から第 12 行までは report06_04.c と同じなので省略 -----

```
13:
14: int gcd(int x, int y)
15: {
16:     if (y != 0) {
17:         return gcd(y, x % y);
18:     }
19:     else {
20:         return x;
21:     }
22: }
```

問14の解答例 「有理数体上の四則演算(関数: 戻り値なし: 最大公約数)」(report06_07.c)

```
1: #include <stdio.h>
2: #include <math.h>
3:
4: void add_q(int a, int b, int c, int d);
5: void sub_q(int a, int b, int c, int d);
6: void mul_q(int a, int b, int c, int d);
7: void div_q(int a, int b, int c, int d);
8: int gcd(int x, int y);
9:
10: int main(void)
11: {
12:     int a = 2, b = 3;
13:     int c = 1, d = 4;
14:
15:     //和
16:     add_q(a, b, c, d);
17:
18:     //差
19:     sub_q(a, b, c, d);
20:
21:     //積
22:     mul_q(a, b, c, d);
23:
24:     //商
25:     div_q(a, b, c, d);
26:
27:     return 0;
28: }
29:
30: void add_q(int a, int b, int c, int d)
31: {
32:     int x, y, tmp;
33:
34:     x = a * d + b * c;
35:     y = b * d;
36:     tmp = gcd(abs(x), abs(y));
37:     printf("(%d/%d)+(%d/%d)=(%d/%d)\n", a, b, c, d, x / tmp, y / tmp);
38: }
```

: ----- 続く -----

問 14 の解答例 「有理数体上の四則演算 (関数 : 戻り値なし : 最大公約数)」 (report06_07.c)

: ----- 続き -----

```
39:
40: void sub_q(int a, int b, int c, int d)
41: {
42:     int x, y, tmp;
43:
44:     x = a * d - b * c;
45:     y = b * d;
46:     tmp = gcd(abs(x), abs(y));
47:     x = x / tmp;
48:     y = y / tmp;
49:     printf("(%d/%d)-(%d/%d)=(%d/%d)%n", a, b, c, d, x, y);
50: }
51:
52: void mul_q(int a, int b, int c, int d)
53: {
54:     int x, y;
55:
56:     x = (a * c) / (gcd(abs(a * c), abs(b * d)));
57:     y = (b * d) / (gcd(abs(a * c), abs(b * d)));
58:     printf("(%d/%d)*(%d/%d)=(%d/%d)%n", a, b, c, d, x, y);
59: }
60:
61: void div_q(int a, int b, int c, int d)
62: {
63:     int x, y;
64:
65:     x = a * d;
66:     y = b * c;
67:     printf("(%d/%d)/(%d/%d)=(%d/%d)%n", a, b, c, d,
68:            x / gcd(abs(x), abs(y)), y / gcd(abs(x), abs(y)));
69: }
```

: ----- 続く -----

問 14 の解答例 「有理数体上の四則演算 (関数 : 戻り値なし : 最大公約数)」 (report06_07.c)

```
: ----- 続き -----  
70:  
71: int gcd(int x, int y)  
72: {  
73:     int tmp;  
74:  
75:     while (y != 0) {  
76:         tmp = x;  
77:         x = y;  
78:         y = tmp % x;  
79:     }  
80:  
81:     return x;  
82: }
```

考察：問 14 に習って、ソースプログラム「有理数体上の四則演算 (関数: グローバル変数)」 (report06_02.c) の計算結果の値が規約となるソースプログラム「有理数体上の四則演算 (関数: グローバル変数: 最大公約数)」 (report06_07a.c) を作成してみましょう。

考察：問 14 に習って、ソースプログラム「有理数体上の四則演算 (関数: 配列)」 (report06_02.c) の計算結果の値が規約となるソースプログラム「有理数体上の四則演算 (関数: 配列: 最大公約数)」 (report06_07b.c) を作成してみましょう。