2016 年度 プログラミング II　レポート 08　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： 2016 年 12 月 13 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： 2016 年 12 月 7 日 (水)

- 切り取り線 -

2016 年度 プログラミング II　レポート 08　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.279 の記述にしたがって、ソースプログラム「1 人分の成績処理」(`rei7_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：`for` 文の式 1 の `data.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 2　p.282 の記述にしたがって、ソースプログラム「4 人分の成績処理」(`rei7_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：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 `data[j].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 3　p.285 の記述にしたがって、ソースプログラム「1 人分の成績処理」(`rei7_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。
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 `s.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 4　p.289 の記述にしたがって、ソースプログラム「1 人分の成績処理」(`rei7_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：`for` 文の式 1 の `p->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 5　p.293 の記述にしたがって、ソースプログラム「ポインタをメンバに持つ構造体」(`rei7_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 3 の `p++`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。

問 6　p.296 の記述にしたがって、ソースプログラム「リスト構造」(`rei7_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：第 44 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「リスト構造」(`rei7_5a_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 7　p.300 の記述にしたがって、ソースプログラム「ファイルのコピー」(`rei7_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。

参考：p.302 の記述にしたがって、ソースプログラム「成績処理」(`rei7_7.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、ソースプログラム「成績処理」(`rei7_7_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 8　ソースプログラム「複素数体上の四則演算」(report03_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構造体 (**変数を利用**) を用いたソースプログラム「複素数体上の四則演算 (関数：構造体：変数)」(report08_01.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「1 人分の成績処理」(rei7_3.c) を参考にしなさい。

● 複素数体上の四則演算 (関数：構造体：変数)　report08_01.c

```
 1: #include <stdio.h>
 2:
 3: typedef struct complex {
 4:     double re;
 5:     double im;
 6: } COMPLEX;
 7:
 8: COMPLEX add_c(COMPLEX x, COMPLEX y);
 9: COMPLEX sub_c(COMPLEX x, COMPLEX y);
10: COMPLEX mul_c(COMPLEX x, COMPLEX y);
11: COMPLEX div_c(COMPLEX x, COMPLEX y);
12:
13: int main(void)
14: {
15:     COMPLEX x = {2.0, 2.5};
16:     COMPLEX y = {-1.5, 4.5};
17:     COMPLEX z;
18:
19:     //和
20:     z = add_c(x, y);
21:     printf("((%f)+(%f)i)+((%f)+(%f)i)=((%f)+(%f)i)¥n",
22:                                     x.re, x.im, y.re, y.im, z.re, z.im);
23:
24:     //差
25:     z = sub_c(x, y);
26:     printf("((%f)+(%f)i)-((%f)+(%f)i)=((%f)+(%f)i)¥n",
27:                                     x.re, x.im, y.re, y.im, z.re, z.im);
```

解答例　別紙を参照のこと。

問 9　ソースプログラム「複素数体上の四則演算」(report03_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構造体 (**ポインタを利用**) を用いたソースプログラム「複素数体上の四則演算 (関数：構造体：ポインタ)」(report08_02.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「1 人分の成績処理」(rei7_3a.c) を参考にしなさい。

● 複素数体上の四則演算 (関数：構造体：ポインタ)　report08_02.c

```
 1: #include <stdio.h>
 2:
 3: typedef struct complex {
 4:     double re;
 5:     double im;
 6: } COMPLEX;
 7:
 8: COMPLEX *padd_c(COMPLEX *px, COMPLEX *py);
 9: COMPLEX *psub_c(COMPLEX *px, COMPLEX *py);
10: COMPLEX *pmul_c(COMPLEX *px, COMPLEX *py);
11: COMPLEX *pdiv_c(COMPLEX *px, COMPLEX *py);
12:
13: int main(void)
14: {
15:     COMPLEX x = {2.0, 2.5};
16:     COMPLEX y = {-1.5, 4.5};
17:     COMPLEX *px, *py, *pz;
18:
19:     px = &x;
20:     py = &y;
21:
22:     //和
23:     pz = padd_c(px, py);
24:     printf("((%f)+(%f)i)+((%f)+(%f)i)=((%f)+(%f)i)¥n",
25:                     px->re, px->im, py->re, py->im, pz->re, pz->im);
26:
27:     //差
28:     pz = psub_c(px, py);
29:     printf("((%f)+(%f)i)-((%f)+(%f)i)=((%f)+(%f)i)¥n",
30:                     px->re, px->im, py->re, py->im, pz->re, pz->im);
```

解答例　別紙を参照のこと。

問 10　下記ソースプログラム「行番号を付加するプログラム」(report08_03.c) は、任意のソースプログラムに対して各行の先頭に行番号を付加するプログラムである。次ページの実行例を参考にして、**このソースプログラム**「行番号を付加するプログラム」(report08_03.c) **自身をこのプログラム**「report08_03.exe」**で処理し、出力ファイルを印刷してレポートに添付しなさい。**
注意: このソースプログラムは、ソースプログラム「ファイルのコピー」(rei7_6a.c) の応用です。

● 行番号を付加するプログラム　report08_03.c

```
 1: #include <stdio.h>
 2: #include <stdlib.h>
 3:
 4: int main(int argc, char *argv[])
 5: {
 6:     FILE *fpi;
 7:     FILE *fpo;
 8:     char buff[256];
 9:     int i = 1;
10:
11:     if (argc != 3) {
12:         printf("引数の数が違います¥n");
13:         exit(EXIT_FAILURE);
14:     }
15:     if ((fpi = fopen(argv[1], "r")) != NULL
16:                             && (fpo = fopen(argv[2], "w")) != NULL) {
17:         while ((fgets(buff, 256, fpi)) != NULL) {
18:             fprintf(fpo, "   %3d: %s", i, buff);
19:             i++;
20:         }
21:     }
22:     else {
23:         printf("ファイルがオープンできません¥n");
24:         exit(EXIT_FAILURE);
25:     }
26:     fclose(fpi);
27:     fclose(fpo);
28:
29:     return 0;
30: }
```

実行例：

```
Z:¥src>report08_03.exe hello.c output.txt Enter

Z:¥src>more output.txt Enter
     1:  #include <stdio.h>
     2:
     3:  int main(void)
     4:  {
     5:          printf("Hello World¥n");
     6:
     7:          return 0;
     8:  }

Z:¥src>
```

解答例　前ページの行番号付ソースプログラムそのものなので省略。

```
Z:¥src>report08_03.exe report08_03.c output.txt Enter

Z:¥src>
```

問 8 の解答例　「複素数体上の四則演算 (関数：構造体：変数)」(report08_01.c)

```
 ⋮           -------- 続き --------

28:
29:     //積
30:     z = mul_c(x, y);
31:     printf("((%f)+(%f)i)*((%f)+(%f)i)=((%f)+(%f)i)¥n",
32:                                     x.re, x.im, y.re, y.im, z.re, z.im);
33:
34:     //商
35:     z = div_c(x, y);
36:     printf("((%f)+(%f)i)/((%f)+(%f)i)=((%f)+(%f)i)¥n",
37:                                     x.re, x.im, y.re, y.im, z.re, z.im);
38:
39:     return 0;
40: }
41:
42: COMPLEX add_c(COMPLEX x, COMPLEX y)
43: {
44:     COMPLEX z;
45:
46:     z.re = x.re + y.re;
47:     z.im = x.im + y.im;
48:
49:     return z;
50: }
51:
52: COMPLEX sub_c(COMPLEX x, COMPLEX y)
53: {
54:     COMPLEX z;
55:
56:     z.re = x.re - y.re;
57:     z.im = x.im - y.im;
58:
59:     return z;
60: }

 ⋮           -------- 続く --------
```

問 8 の解答例　「複素数体上の四則演算 (関数：構造体：変数)」(report08_01.c)

⋮　　-------- 続き --------

```
61:
62: COMPLEX mul_c(COMPLEX x, COMPLEX y)
63: {
64:     COMPLEX z;
65:
66:     z.re = x.re * y.re - x.im * y.im;
67:     z.im = x.re * y.im + x.im * y.re;
68:
69:     return z;
70: }
71:
72: COMPLEX div_c(COMPLEX x, COMPLEX y)
73: {
74:     COMPLEX z;
75:     double tmp;
76:
77:     tmp = y.re * y.re + y.im * y.im;
78:     z.re = (x.re * y.re + x.im * y.im) / tmp;
79:     z.im = (-1 * x.re * y.im + x.im * y.re) / tmp;
80:
81:     return z;
82: }
```

問 9 の解答例　「複素数体上の四則演算 (関数：構造体：ポインタ)」(report08_02.c)

```
 ⋮           -------- 続き --------

31:
32:     //積
33:     pz = pmul_c(px, py);
34:     printf("((%f)+(%f)i)*((%f)+(%f)i)=((%f)+(%f)i)¥n",
35:                     px->re, px->im, py->re, py->im, pz->re, pz->im);
36:
37:     //商
38:     pz = pdiv_c(px, py);
39:     printf("((%f)+(%f)i)/((%f)+(%f)i)=((%f)+(%f)i)¥n",
40:                     px->re, px->im, py->re, py->im, pz->re, pz->im);
41:
42:     return 0;
43: }
44:
45: COMPLEX *padd_c(COMPLEX *px, COMPLEX *py)
46: {
47:     COMPLEX z;
48:     COMPLEX *pz;
49:
50:     pz = &z;
51:
52:     pz->re = px->re + py->re;
53:     pz->im = px->im + py->im;
54:
55:     return pz;
56: }

 ⋮           -------- 続く --------
```

問 9 の解答例　「複素数体上の四則演算 (関数：構造体：ポインタ)」(report08_02.c)

⋮　　-------- 続き --------

```
57:
58: COMPLEX *psub_c(COMPLEX *px, COMPLEX *py)
59: {
60:     COMPLEX z;
61:     COMPLEX *pz;
62:
63:     pz = &z;
64:
65:     pz->re = px->re - py->re;
66:     pz->im = px->im - py->im;
67:
68:     return pz;
69: }
70:
71: COMPLEX *pmul_c(COMPLEX *px, COMPLEX *py)
72: {
73:     COMPLEX z;
74:     COMPLEX *pz;
75:
76:     pz = &z;
77:
78:     pz->re = px->re * py->re - px->im * py->im;
79:     pz->im = px->re * py->im + px->im * py->re;
80:
81:     return pz;
82: }
83:
84: COMPLEX *pdiv_c(COMPLEX *px, COMPLEX *py)
85: {
86:     COMPLEX z;
87:     COMPLEX *pz;
88:     double tmp;
89:
90:     pz = &z;
91:
92:     tmp = py->re * py->re + py->im * py->im;
93:     pz->re = (px->re * py->re + px->im * py->im) / tmp;
94:     pz->im = (-1 * px->re * py->im + px->im * py->re) / tmp;
95:
96:     return pz;
97: }
```