

# 2016年度 プログラミングII レポート12

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 : 2017年1月17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 : 2017年1月11日(水)

# 2016年度 プログラミングII レポート12

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** 選択ソートのアルゴリズムを用いて、配列  $a[11] = \{0, 23, 12, 48, -4, 22, 35, 88, 30, 15, 11\}$  を昇順にソーティングしなさい(空欄を埋め、下表を完成しなさい)。ただし、要素  $a[0] = 0$  はソーティングの範囲に含めないで、要素の入れ替えが起こる度に次の行に新たな配列を記述すること(テキストの Step とは関係ない)。

{0, 23, 12, 48, -4, 22, 35, 88, 30, 15, 11} 初期値  
{0, -4, 12, 48, 23, 22, 35, 88, 30, 15, 11}  
{0, -4, 11, 48, 23, 22, 35, 88, 30, 15, 12}  
{0, -4, 11, 12, 23, 22, 35, 88, 30, 15, 48}  
{0, -4, 11, 12, 15, 22, 35, 88, 30, 23, 48}  
{0, -4, 11, 12, 15, 22, 23, 88, 30, 35, 48}  
{0, -4, 11, 12, 15, 22, 23, 30, 88, 35, 48}  
{0, -4, 11, 12, 15, 22, 23, 30, 35, 88, 48}  
{0, -4, 11, 12, 15, 22, 23, 30, 35, 48, 88}

**問 2** テキストを参考に、選択ソートによって昇順にソーティングするソースプログラム「選択ソート：昇順」(`report12_01.c`)を作成しなさい。ただし、要素  $a[0] = 0$  とし、ソーティングの範囲に含めないこと。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：問 1 の表のように、ソーティングの過程が観察できるようなプログラムにすること。

解答例 別紙を参照のこと。

**問 3** 問 2 のソースプログラム「選択ソート：昇順」(`report12_01.c`)を書き換えて、降順にソーティングするプログラム「選択ソート：降順」(`report12_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。

解答例 別紙を参照のこと。

**問 4** バブルソートのアルゴリズムを用いて、配列  $a[11] = \{0, 23, 12, 48, -4, 22, 35, 88, 30, 15, 11\}$  を昇順にソーティングしなさい（空欄を埋め、下表を完成しなさい）。ただし、要素  $a[0] = 0$  はソーティングの範囲に含めないで、要素の入れ替えが起こる度に新たな配列を記述すること（テキストの Step とは関係ない）。

{0, 23, 12, 48, -4, 22, 35, 88, 30, 15, 11} 初期値  
{0, 12, 23, 48, -4, 22, 35, 88, 30, 15, 11}  
{0, 12, 23, -4, 48, 22, 35, 88, 30, 15, 11}  
{0, 12, 23, -4, 22, 48, 35, 88, 30, 15, 11}  
{0, 12, 23, -4, 22, 35, 48, 88, 30, 15, 11}  
{0, 12, 23, -4, 22, 35, 48, 30, 88, 15, 11}  
{0, 12, 23, -4, 22, 35, 48, 30, 15, 88, 11}  
{0, 12, 23, -4, 22, 35, 48, 30, 15, 11, 88}  
{0, 12, -4, 23, 22, 35, 48, 30, 15, 11, 88}  
{0, 12, -4, 22, 23, 35, 48, 30, 15, 11, 88}  
{0, 12, -4, 22, 23, 35, 30, 48, 15, 11, 88}  
{0, 12, -4, 22, 23, 35, 30, 15, 48, 11, 88}  
{0, 12, -4, 22, 23, 35, 30, 15, 11, 48, 88}  
{0, -4, 12, 22, 23, 35, 30, 15, 11, 48, 88}  
{0, -4, 12, 22, 23, 30, 35, 15, 11, 48, 88}  
{0, -4, 12, 22, 23, 30, 15, 35, 11, 48, 88}  
{0, -4, 12, 22, 23, 30, 15, 11, 35, 48, 88}  
{0, -4, 12, 22, 23, 15, 30, 11, 35, 48, 88}  
{0, -4, 12, 22, 23, 15, 11, 30, 35, 48, 88}  
{0, -4, 12, 22, 15, 23, 11, 30, 35, 48, 88}  
{0, -4, 12, 22, 15, 11, 23, 30, 35, 48, 88}  
{0, -4, 12, 15, 22, 11, 23, 30, 35, 48, 88}  
{0, -4, 12, 15, 11, 22, 23, 30, 35, 48, 88}  
{0, -4, 12, 11, 15, 22, 23, 30, 35, 48, 88}  
{0, -4, 11, 12, 15, 22, 23, 30, 35, 48, 88}

**問 5** テキストを参考に、バブルソートによって昇順にソーティングするソースプログラム「バブルソート：昇順」(report12\_03.c)を作成しなさい。ただし、要素  $a[0] = 0$  とし、ソーティングの範囲に含めないこと。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：問 4 の表のように、ソーティングの過程が観察できるようなプログラムにすること。

解答例 別紙を参照のこと。

**問 6** バブルソートでは、ある Step から要素の交換が全く起こらなくなる場合がある（その配列はソーティングが完了し、最終 Step まで実行する必要が無い）。問 5 のソースプログラム「バブルソート：昇順」(report12\_03.c) を改良して、要素の交換が起らなくなた場合、以後の Step を省略するようなソースプログラム「バブルソート：昇順：改良」(report12\_04.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 別紙を参照のこと。

問2の解答例 「選択ソート：昇順」(report12\_01.c)

```
1: #include <stdio.h>
2:
3: #define N 10
4:
5: void selectionsort(int a[]);
6: void printarray(int a[]);
7:
8: int main(void)
9: {
10:     int a[N + 1] = {0, 4, 10, 5, 2, 1, 7, 8, 6, 3, 9};
11:
12:     printarray(a);
13:     selectionsort(a);
14:
15:     return 0;
16: }
17:
18: void selectionsort(int a[])
19: {
20:     int i, j, k, tmp;
21:
22:     for (i = 1; i <= N - 1; i++) {
23:         k = i;
24:         for (j = i + 1; j <= N; j++) {
25:             if (a[k] > a[j]) k = j;
26:         }
27:         tmp = a[i];
28:         a[i] = a[k];
29:         a[k] = tmp;
30:         if (i != k) printarray(a);
31:     }
32: }
33:
34: void printarray(int a[])
35: {
36:     int i;
37:
38:     for (i = 1; i <= N; i++) printf("%3d ", a[i]);
39:     printf("\n");
40: }
```

問3の解答例 「選択ソート：降順」(report12\_02.c)

```
1: #include <stdio.h>
2:
3: #define N 10
4:
5: void selectionsort(int a[]);
6: void printarray(int a[]);
7:
8: int main(void)
9: {
10:    int a[N + 1] = {0, 4, 10, 5, 2, 1, 7, 8, 6, 3, 9};
11:
12:    printarray(a);
13:    selectionsort(a);
14:
15:    return 0;
16: }
17:
18: void selectionsort(int a[])
19: {
20:    int i, j, k, tmp;
21:
22:    for (i = 1; i <= N - 1; i++) {
23:        k = i;
24:        for (j = i + 1; j <= N; j++) {
25:            if (a[k] < a[j]) k = j; ← 変更箇所
26:        }
27:        tmp = a[i];
28:        a[i] = a[k];
29:        a[k] = tmp;
30:        if (i != k) printarray(a);
31:    }
32: }
33:
34: void printarray(int a[])
35: {
36:    int i;
37:
38:    for (i = 1; i <= N; i++) printf("%3d ", a[i]);
39:    printf("\n");
40: }
```

問5の解答例 「バブルソート：昇順」(report12\_03.c)

```
1: #include <stdio.h>
2:
3: #define N 10
4:
5: void bubblesort(int a[]);
6: void printarray(int a[]);
7:
8: int main(void)
9: {
10:     int a[N + 1] = {0, 4, 10, 5, 2, 1, 7, 8, 6, 3, 9};
11:
12:     printarray(a);
13:     bubblesort(a);
14:
15:     return 0;
16: }
17:
18: void bubblesort(int a[])
19: {
20:     int i, j, tmp;
21:
22:     for (i = N; i > 1; i--) {
23:         for (j = 1; j < i; j++) {
24:             if (a[j] > a[j+1]) {
25:                 tmp = a[j];
26:                 a[j] = a[j+1];
27:                 a[j+1] = tmp;
28:                 printarray(a);
29:             }
30:         }
31:     }
32: }
33:
34: void printarray(int a[])
35: {
36:     int i;
37:
38:     for (i = 1; i <= N; i++) printf("%3d ", a[i]);
39:     printf("\n");
40: }
```

問6の解答例 「バブルソート：昇順：改良」(report12\_04.c)

```
1: #include <stdio.h>
2: #include <stdlib.h>      /* for exit() */
3:
4: #define N 10
5:
6: void bubblesort(int a[]);
7: void printarray(int a[]);
8:
9: int main(void)
10: {
11:     int a[N + 1] = {0, 4, 10, 5, 2, 1, 7, 8, 6, 3, 9};
12:
13:     printarray(a);
14:     bubblesort(a);
15:
16:     return 0;
17: }
18:
19: void bubblesort(int a[])
20: {
21:     int i, j, tmp, flag;                                ← 変更箇所
22:
23:     for (i = N; i > 1; i--) {
24:         flag = 0;                                       ← 変更箇所
25:         for (j = 1; j < i; j++) {
26:             if (a[j] > a[j+1]) {
27:                 tmp = a[j];
28:                 a[j] = a[j+1];
29:                 a[j+1] = tmp;
30:                 flag = 1;                                ← 変更箇所
31:             }
32:             printarray(a);                            ← 変更箇所
33:         }
34:         if (flag == 0) exit(EXIT_SUCCESS);            ← 変更箇所
35:     }
36: }
37:
38: void printarray(int a[])
39: {
40:     int i;
41:
42:     for (i = 1; i <= N; i++) printf("%3d ", a[i]);
43:     printf("\n");
44: }
```