

2017年度 プログラミングII レポート13

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： 2018年1月23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： 2018年1月17日(水)

----- 切り取り線 -----

2017年度 プログラミングII レポート13

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 挿入ソートのアルゴリズムを用いて、配列 $a[11] = \{0, 23, 12, 48, -4, 22, 35, 88, 30, 15, 11\}$ を昇順にソートしなさい (空欄を埋め、下表を完成しなさい)。ただし、要素 $a[0] = 0$ はソートの範囲に含めなくて、要素の挿入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 と一致する)。

{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} Step 1
{0, 12, 23, 48, -4, 22, 35, 88, 30, 15, 11} Step 2
{0, -4, 12, 23, 48, 22, 35, 88, 30, 15, 11} Step 3
{0, -4, 12, 22, 23, 48, 35, 88, 30, 15, 11} Step 4
{0, -4, 12, 22, 23, 35, 48, 88, 30, 15, 11} Step 5
{0, -4, 12, 22, 23, 35, 48, 88, 30, 15, 11} Step 6
{0, -4, 12, 22, 23, 30, 35, 48, 88, 15, 11} Step 7
{0, -4, 12, 15, 22, 23, 30, 35, 48, 88, 11} Step 8
{0, -4, 11, 12, 15, 22, 23, 30, 35, 48, 88} Step 9

問 2 テキストを参考に、挿入ソートによって昇順にソートするソースプログラム「挿入ソート：昇順」(report13_01.c) を作成しなさい。ただし、要素 $a[0] = 0$ とし、ソート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、ソートの過程を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「挿入ソート：昇順」(report13_01.c) を書き換えて、降順にソートするプログラム「挿入ソート：降順」(report13_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 クイックソートのアルゴリズムを用いて、配列 $a[11] = \{0, 23, 12, 48, -4, 22, 35, 88, 30, 15, 11\}$ を昇順にソートしなさい (空欄を埋め、下表を完成しなさい)。ただし、空欄には、要素 $a[0]$ は除き、テキスト (p.169) のソースプログラム「クイックソートによるソートプログラム」(quicksort.c) の第 25 行および第 34 行の関数「printarray()」によって出力される配列を記述すること (テキストの Step とは関係ない)。

注意：分割によるソートの範囲と再帰的プログラムであることに注意すること。

{0, 23, 12, 48, -4, 22, 35, 88, 30, 15, 11} 第 25 行
{0, 11, 12, 48, -4, 22, 35, 88, 30, 15, 23} 第 34 行
{0, 11, 12, 15, -4, 22, 35, 88, 30, 48, 23} 第 34 行
{0, 11, 12, 15, -4, 22, 35, 88, 30, 48, 23} 第 25 行
{0, -4, 12, 15, 11, 22, 35, 88, 30, 48, 23} 第 34 行
{0, -4, 12, 15, 11, 22, 35, 88, 30, 48, 23} 第 25 行
{0, -4, 11, 15, 12, 22, 35, 88, 30, 48, 23} 第 34 行
{0, -4, 11, 15, 12, 22, 35, 88, 30, 48, 23} 第 25 行
{0, -4, 11, 12, 15, 22, 35, 88, 30, 48, 23} 第 34 行
{0, -4, 11, 12, 15, 22, 35, 88, 30, 48, 23} 第 25 行
{0, -4, 11, 12, 15, 22, 35, 88, 30, 48, 23} 第 25 行
{0, -4, 11, 12, 15, 22, 23, 88, 30, 48, 35} 第 34 行
{0, -4, 11, 12, 15, 22, 23, 30, 88, 48, 35} 第 34 行
{0, -4, 11, 12, 15, 22, 23, 30, 88, 48, 35} 第 25 行
{0, -4, 11, 12, 15, 22, 23, 30, 88, 48, 35} 第 25 行
{0, -4, 11, 12, 15, 22, 23, 30, 35, 48, 88} 第 34 行
{0, -4, 11, 12, 15, 22, 23, 30, 35, 48, 88} 第 25 行

問 5 テキスト (p.169) のソースプログラム「クイックソートによるソートプログラム」(quicksort.c) を書き換えて、降順にソートするプログラム「クイックソート:降順」(report_13_03.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 「挿入ソート：昇順」(report13_01.c)

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

問 3 の解答例 「挿入ソート：降順」(report13_02.c)

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

← 変更箇所

問 5 の解答例 「クイックソート：降順」 (report13_03.c)

```
1: #include <stdio.h>
2:
3: #define N 10
4:
5: void quicksort(int a[], int start, int end);
6: void printarray(int a[], int start, int end);
7:
8: int main(void)
9: {
10:     int a[N + 1] = {0, 4, 10, 5, 2, 1, 7, 8, 6, 3, 9};
11:
12:     quicksort(a, 1, N);
13:
14:     return 0;
15: }
16:
17: void quicksort(int a[], int start, int end)
18: {
19:     int i, j, x, tmp;
20:
21:     i = start;
22:     j = end;
23:     x = a[start];
24:
25:     printarray(a, start, end);
26:     while (1) {
27:         while (a[i] > x) i++;           ← 変更箇所
28:         while (a[j] < x) j--;         ← 変更箇所
29:         if (i >= j) break;
30:         tmp = a[i];
31:         a[i] = a[j];
32:         a[j] = tmp;
33:         i++; j--;
34:         printarray(a, start, end);
35:     }
36:     printf("%#n");
37:     if (start < i - 1) quicksort(a, start, i - 1);
38:     if (j + 1 < end) quicksort(a, j + 1, end);
39: }
40:
: ----- 関数「printarray()」は同じなので省略 -----
```