

情報処理 2 - 第 13 回課題

柴田健琉

提出日：2026 年 07 月 19 日
2026 年 07 月 19 日

1 はじめに

この課題のプログラムは以下の環境での動作が確認されている.

- OS: NixOS 26.05 Yarara, GNU/Linux 7.1.3 x86_64
- CC: GCC 15.2.0
- CFLAGS: -g -O1 -Wall -Wpedantic

2 課題 1

単純選択ソートと単純挿入ソートの 2 通りで授業サンプルで示された **Student** 構造体の **height** を昇順で並べ替えるプログラムを作成する.

課題 1 のソースコード

```
1  #include <ctype.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  #define STUDENT_NUM 5
6  #define NAME_LEN 64
7
8  struct Student {
9      char name[NAME_LEN];
10     int height;
11 };
12
13 typedef void (*sort_alg)(struct Student[], int);
14
15 void insertion_sort(struct Student students[], int n) {
16     struct Student tmp = {0};
17     for (int i = 1; i < n; i++) {
18         for (int j = 0; j < i; j++) {
19             if (students[j].height > students[i].height) {
20                 tmp = students[i];
21                 students[i] = students[j];
22                 students[j] = tmp;
23             }
24         }
25     }
26 }
27
28 void selection_sort(struct Student students[], int n) {
29     struct Student tmp = {0};
30     for (int i = 0; i < n; i++) {
31         int min_idx = i;
32         for (int j = i; j < n; j++) {
33             if (students[j].height < students[min_idx].height) {
```

```

34         min_idx = j;
35     }
36 }
37 tmp = students[i];
38 students[i] = students[min_idx];
39 students[min_idx] = tmp;
40 }
41 }
42
43 int main(int argc, char** argv) {
44     if (argc < 2) {
45         printf("Please specify sort algorithm (0: insertion, 1: selection)\n");
46         return 1;
47     }
48
49     sort_alg alg = NULL;
50     sort_alg algs[3] = {insertion_sort, selection_sort, NULL};
51
52     if (isdigit(argv[1][0])) {
53         int idx = argv[1][0] - '0';
54         alg = algs[idx < 2 ? idx : 2];
55     } else {
56         printf("Please input correct value (0: insertion, 1: selection)\n");
57         return 1;
58     }
59
60     if (alg == NULL) {
61         printf("Please input correct value (0: insertion, 1: selection)\n");
62         return 1;
63     }
64
65     struct Student students[STUDENT_NUM] = {
66         {"Sato", 178},
67         {"Sanaka", 175},
68         {"Takao", 173},
69         {"Mike", 165},
70         {"Masaki", 179}
71     };
72
73     printf("Initial State:\n");
74     for (int i = 0; i < STUDENT_NUM; i++) {
75         printf("%2d: %-8s%4d\n", i+1, students[i].name, students[i].height);
76     }
77
78     (*alg)(students, STUDENT_NUM);
79
80     printf("Sorted State:\n");
81     for (int i = 0; i < STUDENT_NUM; i++) {
82         printf("%2d: %-8s%4d\n", i+1, students[i].name, students[i].height);
83     }
84
85     return 0;
86 }

```

2.1 実行結果

```
report cc ~/Documents/nit-info-proc-2-S1/src/build/cls13 $ ./main 0
Initial State:
1: Sato      178
2: Sanaka    175
3: Takao     173
4: Mike      165
5: Masaki    179
Sorted State:
1: Mike      165
2: Takao     173
3: Sanaka    175
4: Sato      178
5: Masaki    179
report cc ~/Documents/nit-info-proc-2-S1/src/build/cls13 $ ./main 1
Initial State:
1: Sato      178
2: Sanaka    175
3: Takao     173
4: Mike      165
5: Masaki    179
Sorted State:
1: Mike      165
2: Takao     173
3: Sanaka    175
4: Sato      178
5: Masaki    179
report cc ~/Documents/nit-info-proc-2-S1/src/build/cls13 $
```

図1: 課題 1 の実行結果

2.2 処理の流れ

1. プログラムの引数の数を確認し, 2 以下であればエラー終了させる
2. 関数ポインタ **alg** と関数の配列 **algs** を宣言・初期化する
3. プログラムの第 1 引数の 1 文字目が数であれば **argv[1]** の値によって **algs** からどれか一つを **alg** に代入させ, そうでなければプログラムをエラー終了させる
4. **alg** が未だ **NULL** を指していれば, プログラムをエラー終了させる
5. 構造体の配列 **students** を宣言・初期化する
6. 標準出力にソート前の配列の状態を表示させる
7. **alg** が指している関数を引数とともに呼び出す
8. ソート後の配列の状態を標準出力に表示させる
9. プログラムを正常終了させる

2.3 書く際の難点

シンプルなプログラムだったので特に難しいと感じずにプログラムを組めた。

2.4 想定されるバグ

第 1 引数の 1 文字目のみをテストしているので後続の文字列は何でも良いことになっているが、他の使用者からすれば意図しない動作だと感じられてしまう。