

情報処理 2 - 前期第 9 回課題

柴田健琉

提出日：2026 年 06 月 16 日
2026 年 06 月 16 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, Linux 7.0.12 x86_64
- CC: GCC 15.2.0
- CFLAGS: -g -O1 -Wall -Wpedantic
- デバッガ: GNU gdb (GDB) 17.1

2 課題 1 - 演習 10-1

教科書の演習 10-1 の `adjust_point` 関数を作成する.

Listing 1: 課題 1 のプログラム

```
1  #include <stdio.h>
2
3  void adjust_point(int *n) {
4      if (*n < 0) {
5          *n = 0;
6      } else if (*n > 100) {
7          *n = 100;
8      }
9  }
10
11 int main(void) {
12     int d = 0;
13
14     printf("Input integer: ");
15     scanf("%d", &d);
16
17     adjust_point(&d);
18     printf("Result: %d\n", d);
19
20     return 0;
21 }
```

2.1 実行結果

```
report cc cls09 $ ./a1
Input integer: 63
Result: 63
report cc cls09 $ ./a1
Input integer: -32
Result: 0
report cc cls09 $ ./a1
Input integer: 127
Result: 100
report cc cls09 $ ./a1
Input integer: 87
Result: 87
report cc cls09 $ █
```

図1: 課題 1 の実行結果

2.2 処理の流れ

1. 標準入力から整数を取得, 変数 **d** に格納
2. **adjust_point** 関数を変数 **d** のポインタを引数として呼び出す
3. ディレファレンスした引数の値をもとに, 0 未満であれば引数が指す値を 0 に, 100 より大きければ 100 に, それ以外の値では何もしない
4. 変数 **d** の値を標準出力に表示

2.3 書く際の難点

この課題では特に大きな疑問・難点を抱えずに素直に処理を記述できた.

2.4 想定されるバグ

`scanf` 関数の戻り値を確認していないので, 入力 that 正常かどうか不明のまま処理を行う. これにより, 意図しない結果が出る可能性がある.

3 課題 2 - 演習 10-2

教科書の演習 10-2 の `increment_date` と `decrement_date` 関数を作成する. なお, 閏年は考慮しないものとする.

Listing 2: 課題 2 のプログラム

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define BUFFSZ 32
5
6  const int DAYS_IN_MONTH[12] = {
7      31,
8      28,
9      31,
10     30,
11     31,
12     30,
13     31,
14     31,
15     30,
16     31,
17     30,
18     31
19 };
20
21 void increment_date(int *year, int *month, int *day) {
22     (*day)++;
23
24     if (*day > DAYS_IN_MONTH[*month - 1]) {
25         *day = 1;
26         (*month)++;
27     }
28
29     if (*month > 12) {
30         *month = 1;
31         (*year)++;
32     }
33 }
34
35 void decrement_date(int *year, int *month, int *day) {
36     (*day)--;
37 }
```

```

38     if (*day < 1) {
39         (*month)--;
40         if (*month < 1) {
41             *month = 12;
42             (*year)--;
43         }
44         *day = DAYS_IN_MONTH[*month - 1];
45     }
46 }
47
48 int main(void) {
49     char *buff = (char*)malloc(sizeof(char) * BUFFSZ);
50
51     int m, d, y;
52     printf("Input date in mm/dd/yyyy:");
53     fgets(buff, BUFFSZ, stdin);
54     sscanf(buff, "%d/%d/%d", &m, &d, &y);
55
56
57     char opr_in;
58     while (1) {
59         printf("Select operation [+/-]:");
60         opr_in = getc(stdin);
61         if (opr_in == '+') {
62             increment_date(&y, &m, &d);
63             printf("Tomorrow is %02d/%02d/%4d\n", m, d, y);
64             break;
65         } else if (opr_in == '-') {
66             decrement_date(&y, &m, &d);
67             printf("Yesterday was %02d/%02d/%4d\n", m, d, y);
68             break;
69         } else {
70             printf("Error: Please input correct operation mode.\n");
71         }
72     }
73
74
75     return 0;
76 }

```

3.1 実行結果

```
report cc cls09 $ ./a2
Input date in mm/dd/yyyy: 01/01/1970
Select operation [+/-]: -
Yesterday was 12/31/1969
report cc cls09 $ ./a2
Input date in mm/dd/yyyy: 3/3/2022
Select operation [+/-]: +
Tomorrow is 03/04/2022
report cc cls09 $ ./a2
Input date in mm/dd/yyyy: 8/31/2017
Select operation [+/-]: +
Tomorrow is 09/01/2017
report cc cls09 $ ./a2
Input date in mm/dd/yyyy: 12/31/2026
Select operation [+/-]: +
Tomorrow is 01/01/2027
report cc cls09 $ █
```

図2: 課題 2 の実行結果

3.2 処理の流れ

3.2.1 main 関数

1. 標準入力から日付を含む行を取得し、文字列バッファに格納する
2. 文字列バッファから `sscanf` 関数を用いて年月日それぞれの値を取得する
3. 標準入力から 1 文字取得し、`opr_in` 変数に格納する
4. '+' を受取れば `increment_date` 関数を、 '-' を受取れば `decrement_date` 関数を呼び出す、それ以外の文字であれば手順 3 に戻る
5. 標準出力に操作の結果を表示する

3.2.2 increment_date 関数

1. ディレファレンスした変数 `day` をインクリメントする
2. `day` が指す値が月の日数より大きければ、`day` が指す値を 1 にし、`month` が指す値をイン

クリメントする

3. **month** が指す値が年の月数より大きければ, **month** が指す値を 1 にし, **year** が指す値をインクリメントする

3.2.3 decrement_date 関数

1. ディレファレンスした変数 **day** をデクリメントする
2. **day** が指す値が 0 以下の時,
 - (a) **month** が指す値をデクリメントする
 - (b) **month** が指す値が 0 以下の時, その値を 12 にし, **year** が指す値をデクリメントする
 - (c) **day** が指す値を **month** の日数にする

3.3 書く際の難点

日付入力を受け付ける際, **scanf** 関数を使用した直後に別の標準入力にアクセスする関数を実行すると日付入力に入った改行文字が次の関数で読み取られ, **main** 関数の手順 4 の操作指定用の文字以外の文字が入力された状態になってしまうことが発生した. この問題は **fgets** 関数で行単位で標準入力から文字列バッファにコピーし, バッファに対し **sscanf** 関数を使用することで回避した.

3.4 想定されるバグ

日付入力のサニタイズ (検疫) を行っていないので異常値 (例: 15/01/1970) が入力された際, 両関数では正常な演算が実行できず, 時には **DAYS_IN_MONTH** 配列の範囲外へアクセスする恐れがある.

4 課題 3 - 演習 10-3

教科書の演習 10-3 の **sort3** 関数を作成する.

Listing 3: 課題 3 のプログラム

```
1 #include <stdio.h>
2
3 void swap(int *a, int *b) {
4     int t = *a;
5     *a = *b;
6     *b = t;
7 }
8
9 void sort3(int *x, int *y, int *z) {
10     if (*x > *y) swap(x, y);
```

```

11     if (*y > *z) swap(y, z);
12     if (*x > *y) swap(x, y);
13 }
14
15 int main(void) {
16     int a,b,c;
17
18     printf("Input 3 integers separated by ',' :");
19     scanf("%d,%d,%d", &a, &b, &c);
20
21     sort3(&a, &b, &c);
22
23     printf("Result: %d, %d, %d\n", a, b, c);
24
25     return 0;
26 }

```

4.1 実行結果

```

report cc cls09 $ ./a3
Input 3 integers separated by ',': 1,3,2
Result: 1, 2, 3
report cc cls09 $ ./a3
Input 3 integers separated by ',': 6,2,1
Result: 1, 2, 6
report cc cls09 $ ./a3
Input 3 integers separated by ',': 4,8,3
Result: 3, 4, 8
report cc cls09 $ ./a3
Input 3 integers separated by ',': 7,5,9
Result: 5, 7, 9
report cc cls09 $ ./a3
Input 3 integers separated by ',': 3,4,5
Result: 3, 4, 5
report cc cls09 $ █

```

図3: 課題 3 の実行結果

4.2 処理の流れ

4.2.1 main 関数

1. 標準入力から 3 つの整数値を取得し, 変数 **a**, **b**, **c** にそれぞれ格納する

2. 取得した値に対し **sort3** 関数を適用する
3. 標準出力に結果を表示する

4.2.2 sort3 関数

1. **x** が指す値が **y** が指す値より大きければ **swap** 関数で 2 つの値を交換する
2. **y** が指す値が **z** が指す値より大きければ **swap** 関数で 2 つの値を交換する
3. **x** が指す値が **y** が指す値より大きければ **swap** 関数で 2 つの値を交換する

4.3 書く際の難点

最初は配列やループを使用せずにどうやって並び換えを行うか一瞬分からなかったが, 3 要素のバブルソートの処理を直接記述することで解決した.

4.4 想定されるバグ

scanf 関数の戻り値である変換に成功した値の数を確認していないので 1 つ以上の変数が初期化されていない状態が発生しうる. この状態では意図しない値が並び換えの順序に影響してしまう.