

情報処理 2 - 前期第 4 回課題

柴田健琉

提出日：2026 年 05 月 12 日
2026 年 05 月 11 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 25.11 Xantusia, Linux 7.0.3 x86_64
- CC: GCC 15.2.0
- CFLAGS: -g -O1 -Wall -Wpedantic
- ファイルエンコーディング: UTF-8

2 補助関数群

今回の課題では前期第3回課題で作成した関数を使用する. 再利用性を高めるために `reduce.h` を作成した.

`reduce.h`

```
1 #ifndef __REDUCE_H__
2 #define __REDUCE_H__
3
4 int gcm(int a, int b) {
5     int x = a, y = b;
6     if (x < y) {
7         x = b;
8         y = a;
9     }
10    int r = a % b;
11    if (r == 0) return b;
12    return gcm(b, r);
13 }
14
15 int lcm(int a, int b) {
16     int g = gcm(a, b);
17     return b * a / g;
18 }
19
20 void reduce(int x[]) {
21     int a = x[0], b = x[1];
22     int g = gcm(a, b);
23     x[0] = a / g;
24     x[1] = b / g;
25 }
26
27
28 #endif // __REDUCE_H__
```

また、分数の入出力等に関する関数を `utils.h` に記述した.

utils.h

```
1  #ifndef __UTILS_H__
2  #define __UTILS_H__
3
4  #include <stdio.h>
5
6  void inputFrac(int x[]) {
7      printf("Input fraction(a/b): ");
8      scanf("%d/%d", &x[0], &x[1]);
9  }
10
11 void formatNeg(int x[]) {
12     size_t sz = sizeof(int);
13     int cnt = 0;
14     for (size_t i = 0; i < 2; i++) {
15         cnt += x[i] < 0 ? 1 : 0;
16         x[i] = x[i] < 0 ? x[i] * -1 : x[i];
17     }
18
19     if (cnt == 1)
20         x[0] = -1 * x[0];
21 }
22
23 void printFrac(int x[]) {
24     if (x[1] == 1) {
25         printf("%d", x[0]);
26     } else {
27         printf("%d/%d", x[0], x[1]);
28     }
29 }
30
31 #endif // __UTILS_H__
```

3 課題 1

reduce 関数を用いて分数どうしの加算を行うプログラム.

課題 1 のプログラム

```
1  #include <stdio.h>
2
3  #include "reduce.h"
4  #include "utils.h"
5
6  int addFrac(int x[], int y[]) {
7      if (x[1] == 0 || y[1] == 0) {
8          return -1;
9      }
10
11     int l = lcm(x[1], y[1]);
12     int z[] = {
13         (x[0] * l)/x[1] + (y[0] * l)/y[1],
14         l
15     };
16
17     reduce(z);
18     formatNeg(z);
19
20     x[0] = z[0];
21     x[1] = z[1];
22     return 0;
23 }
24
25 int main(void) {
26     int x[2] = {0};
27     int y[2] = {0};
28
29     inputFrac(x);
30     inputFrac(y);
31
32     int ret = addFrac(x, y);
33
34     if (ret == 0) {
35         printFrac(x);
36         putchar('\n');
37     } else {
38         fprintf(stderr, "Error: _detected_zero_div\n");
39     }
40
41     return 0;
42 }
```

3.1 実行結果

```
wakugumi-hitoroku% ./a1
Input fraction(a/b): 3/4
Input fraction(a/b): 9/5
51/20
wakugumi-hitoroku% ./a1
Input fraction(a/b): 1/2
Input fraction(a/b): -2/9
5/18
wakugumi-hitoroku% ./a1
Input fraction(a/b): 6/5
Input fraction(a/b): 2/0
Error: detected zero div
wakugumi-hitoroku% █
```

図1: 課題 1 の実行結果

4 課題 2

reduce 関数を用いて分数どうしの減算を行うプログラム.

課題 2 のプログラム

```
1  #include <stdio.h>
2
3  #include "reduce.h"
4  #include "utils.h"
5
6  int subFrac(int x[], int y[]) {
7      if (x[1] == 0 || y[1] == 0) {
8          return -1;
9      }
10
11     int l = lcm(x[1], y[1]);
12     int z[] = {
13         (x[0] * l)/x[1] - (y[0] * l)/y[1],
14         l
15     };
16
17     reduce(z);
18     formatNeg(z);
19
20     x[0] = z[0];
21     x[1] = z[1];
22
23     return 0;
24 }
25
26 int main(void) {
27     int x[2] = {0};
28     int y[2] = {0};
29
30     inputFrac(x);
31     inputFrac(y);
32
33     int ret = subFrac(x, y);
34
35     if (ret == 0) {
36         printFrac(x);
37         putchar('\n');
38     } else {
39         fprintf(stderr, "Error: _detected_zero_div\n");
40     }
41
42     return 0;
43 }
```

4.1 実行結果

```
wakugumi-hitoroku% ./a2
Input fraction(a/b): 1/2
Input fraction(a/b): 1/3
1/6
wakugumi-hitoroku% ./a2
Input fraction(a/b): 6/7
Input fraction(a/b): 8/3
-38/21
wakugumi-hitoroku% ./a2
Input fraction(a/b): 3/4
Input fraction(a/b): 1/5
11/20
wakugumi-hitoroku% ./a2
Input fraction(a/b): 5/0
Input fraction(a/b): 4/9
Error: detected zero div
wakugumi-hitoroku% █
```

図2: 課題 2 の実行結果

5 課題 3

reduce 関数を用いて分数どうしの乗算を行うプログラム.

課題 3 のプログラム

```
1  #include <stdio.h>
2
3  #include "reduce.h"
4  #include "utils.h"
5
6  int mulFrac(int x[], int y[]) {
7      if (x[1]*y[1] == 0) {
8          return -1;
9      }
10
11     int z[] = {
12         x[0] * y[0],
13         x[1] * y[1]
14     };
15
16     reduce(z);
17     formatNeg(z);
18
19     x[0] = z[0];
20     x[1] = z[1];
21
22     return 0;
23 }
24
25 int main(void) {
26     int x[2] = {0};
27     int y[2] = {0};
28
29     inputFrac(x);
30     inputFrac(y);
31
32     int ret = mulFrac(x, y);
33
34     if (ret == 0) {
35         printFrac(x);
36         putchar('\n');
37     } else {
38         fprintf(stderr, "Error: Detected zero div\n");
39     }
40
41     return 0;
42 }
```

5.1 実行結果

```
wakugumi-hitoroku% ./a3
Input fraction(a/b): 1/3
Input fraction(a/b): 5/2
5/6
wakugumi-hitoroku% ./a3
Input fraction(a/b): -8/7
Input fraction(a/b): 2/6
-8/21
wakugumi-hitoroku% ./a3
Input fraction(a/b): 6/5
Input fraction(a/b): 10/0
Error: Detected zero div
wakugumi-hitoroku% ./a3
Input fraction(a/b): 4/3
Input fraction(a/b): 9/4
3
wakugumi-hitoroku% █
```

図3: 課題 3 の実行結果

6 課題 4

標準入力から自然数を取得する関数 `getnum` を作成する.

課題 4 のプログラム

```
1  #include <stdio.h>
2
3  int getnum(void) {
4      int ch, n = 0;
5
6      int stack[10] = {0}; // ceil(log_10(2^31))
7      int i = 0;
8      while ((ch = getchar()) != '\n') {
9          int x = ch - '0';
10         if (x < 0 || x > 9) {
11             continue;
12         }
13         if (i <= 9) stack[i++] = x; // stack push
14     }
15
16     int d = 1;
17     for (int j = i - 1; j >= 0; j--) {
18         n += stack[j] * d; // stack pop
19         d = d * 10;
20     }
21
22     return n;
23 }
24
25 int main(void) {
26     int ret = getnum();
27
28     printf("got: %d\n", ret);
29
30     return 0;
31 }
```

6.1 実行結果

```
wakugumi-hitoroku% ./a4
512
got: 512
wakugumi-hitoroku% ./a4
j534ljb2
got: 5342
wakugumi-hitoroku% ./a4
00008461
got: 8461
wakugumi-hitoroku% ./a4
2147483648
got: -2147483648
wakugumi-hitoroku% ./a4
2147483647
got: 2147483647
wakugumi-hitoroku% ./a4
12345678901234567890
got: 1234567890
wakugumi-hitoroku% █
```

図4: 課題 4 の実行結果