

# 令和 6 年度 情報工学実験 I 報告書

## 実験題目

プログラミング演習 4

## 指導教員

丸山教員, 安細教員, 周教員

## 実験日

- 令和 6 年 10 月 02 日 (水) ~ 令和 6 年 10 月 16 日 (水)

## レポート

- 提出締切日: 令和 6 年 10 月 30 日 (水)
- 受理最終日: 令和 6 年 11 月 20 日 (水)
- 提出日: 令和 6 年 \_\_\_\_ 月 \_\_\_\_ 日 (\_\_\_\_)

## 報告者

2 年 31 番 氏名 橋本 千聡

## 共同実験者

川和 李圭, 鈴木 隆生, 安田 れん

## 1. 実験の目的

プログラムの共同開発演習を通して、議論などを伴うチームでのプログラム作成手法を理解する。

## 2. 実験の概要

- 1・2 週目: 実行環境の確認及び C 言語サンプル実行確認  
作成分担調整、分担一覧や全体構成の資料作成
- 3・4 週目: 各自の担当箇所を作成、単体動作確認
- 5・6 週目: 各自の作成の関数を統合して動作確認  
レポート報告内容のまとめ、レポート作成

## 3. 演習課題の報告

プログラム全体の概要

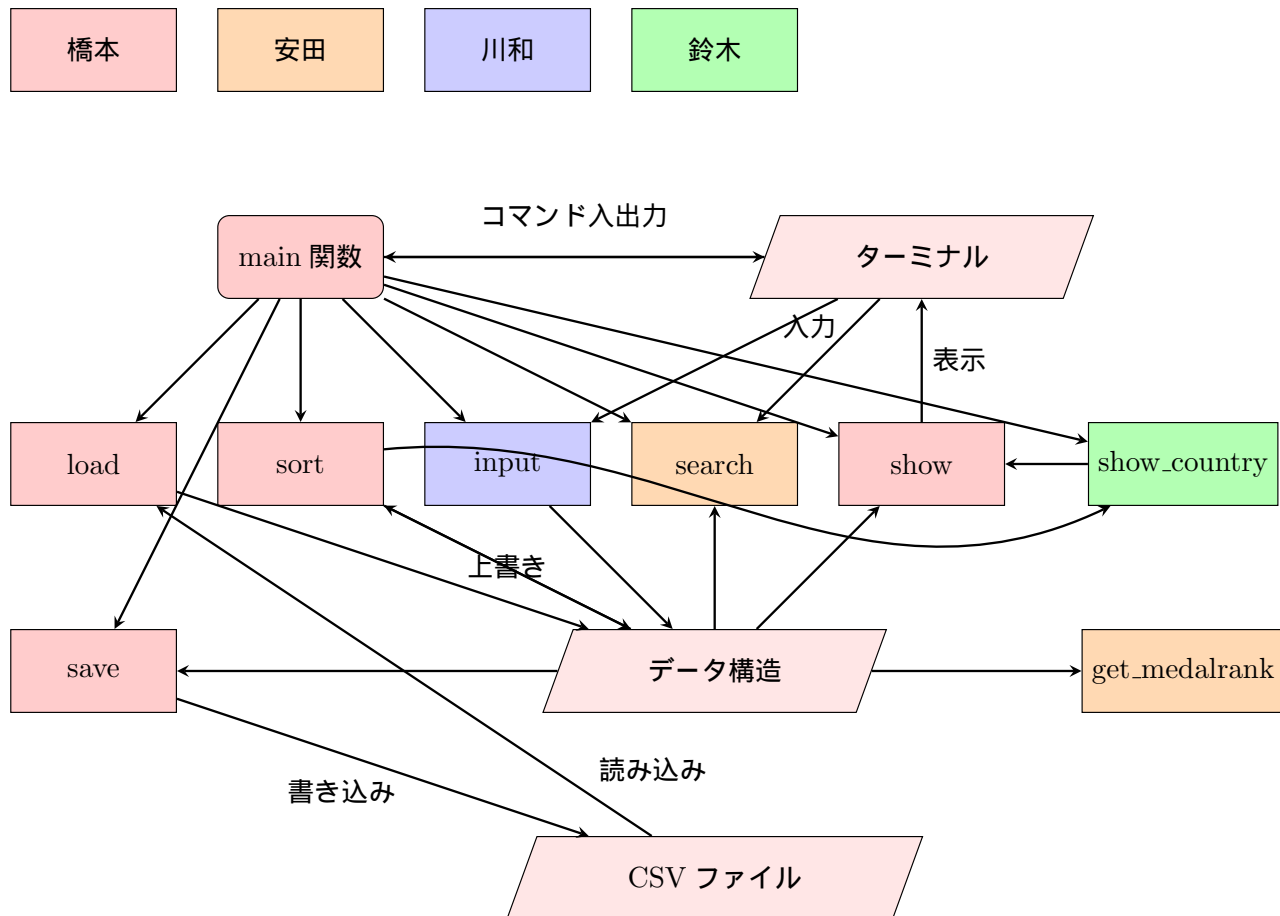


図 1: プログラムのフローチャート +α

### 担当した機能の構成と説明

sort 関数

- 関数の説明: データ構造を並び替える
- 関数の入力: モード
- 関数の出力: なし (データ構造上書き)
- 関数の処理内容: merge sort を行い、mode によって国名金銀銅総メダル数のいずれかで並び替える

```
1  #include "main.h"
2
3  // 安定な比較関数
4  int compare_by_mode(const country_data_type* a, const country_data_type* b,
5                      int mode) {
6      if (mode == 0) {
7          return strcmp(a->country, b->country); // 国名で昇順
8      } else if (mode == 1) {
9          return b->gold - a->gold; // 金メダル数で降順
10     } else if (mode == 2) {
11         return b->silver - a->silver; // 銀メダル数で降順
12     } else if (mode == 3) {
13         return b->bronze - a->bronze; // 銅メダル数で降順
14     } else if (mode == 4) {
15         return b->sum - a->sum; // 合計メダル数で降順
16     }
17     return 0; // デフォルトは等しいと見なす
18 }
19
20 // 安定なマージ関数
21 void merge(country_data_type arr[], int left, int mid, int right, int mode) {
22     int n1 = mid - left + 1;
23     int n2 = right - mid;
24
25     country_data_type L[n1], R[n2];
26
27     for (int i = 0; i < n1; i++)
28         L[i] = arr[left + i];
29     for (int j = 0; j < n2; j++)
30         R[j] = arr[mid + 1 + j];
31
32     int i = 0, j = 0, k = left;
33
34     while (i < n1 && j < n2) {
35         if (compare_by_mode(&L[i], &R[j], mode) <= 0) {
36             arr[k] = L[i];
37             i++;
38         } else {
39             arr[k] = R[j];
40             j++;
41         }
42         k++;
43     }
44
45     while (i < n1) {
46         arr[k] = L[i];
47         i++;
48         k++;
49     }
50
51     while (j < n2) {
```

```

51         arr[k] = R[j];
52         j++;
53         k++;
54     }
55 }
56
57 // マージソート関数
58 void mergeSort(country_data_type arr[], int left, int right, int mode) {
59     if (left < right) {
60         int mid = left + (right - left) / 2;
61
62         mergeSort(arr, left, mid, mode);
63         mergeSort(arr, mid + 1, right, mode);
64
65         merge(arr, left, mid, right, mode);
66     }
67 }
68
69 // ソート関数
70 void sort(int mode) {
71     mergeSort(data, 0, data_size - 1, mode);
72 }

```

---

#### load 関数

- 関数の説明: CSV ファイルからデータを読み込む
- 関数の入力: ファイル名
- 関数の出力: なし (データ構造上書き)
- 関数の処理内容: 引数のファイル名から CSV ファイルを読み込み、データ構造に格納する

#### save 関数

- 関数の説明: データ構造を CSV ファイルに書き込む
- 関数の入力: ファイル名
- 関数の出力: なし
- 関数の処理内容: 引数のファイル名にデータ構造を書き込む

#### show 関数

- 関数の説明: データ構造を表示する
- 関数の入力: なし
- 関数の出力: なし
- 関数の処理内容: printf のフォーマット機能を駆使してテーブル形式でデータ構造を表示する

#### main 関数

- 関数の説明: メイン関数
- 関数の入力: 引数 DEBUG の有無
- 関数の出力: 標準出力
- 関数の処理内容: bash をベースとしたターミナルでのコマンド入力を受け付け、各関数を呼び出す

- その他: デバッグモードを有効にすると、各関数の詳細情報を表示する

担当した機能の単体テストの方法と結果

統合したプログラムの結合テストの方法と結果

#### 4. 実験の感想