

茨城高専 2I 情報工学実験 第4回 第5班

要件

最大100名分の名簿情報(出席番号、氏名、出身中学)を入力して蓄積することができ、次の機能を有するC言語プログラムを作成しなさい。また、そのプログラムを演習室サーバ内(CUI環境)で実装しなさい。

- 出席番号順(昇順)に名簿情報を一覧表
→ `order_of_attendance_no()` にて実装
- 氏名順(辞書順,昇順)に名簿情報を一覧表示
→ `order_of_name()` にて実装
- 出席番号を入力すると氏名を表示
→ `find_name_from_number()` にて実装
- 氏名を入力すると出席番号を表示
→ `find_number_from_name()` にて実装
- 出身中学を入力すると,該当する人物の出席番号・氏名を一覧表示
→ `find_all_students_from_jhs_name()` にて実装

共通のおねがい（非コード）

できればGitを使ってほしい気持ちがありますが、今回はGitおよびGitHubの学習コストを考慮し、GoogleDriveを用いてソースコードを共有します。

今回は、前期の「プログラミングI」のみの知識で書けるように調整しました。

わからなかったら聞いてほしい気持ちは山々ですが、まずは自分で振り返ってみてもらえると力がつくと思います。

共通のおねがい（コード）

1

まず、構造体を使った方法で実装するか、それとも複数の配列を使う方法で実装したいかを教えて下さい。

構造体を使った方法のほうが、より簡潔に記述することができるのでおすすめですが、今まで習った知識だけでつくりたい、という場合は複数の配列を使う方法で実装しても構いません。

上に構造体自体も難しいものではないので、個人的には構造体をおすすめします。

2

必ず、すべてのメッセージは英語（半角文字）のみにしてください。

全角文字はすべてのコード中で使用禁止です。

文字化けを回避するためです。文字コードを設定すれば問題ないですが、トラブルを避けるために禁止します。

3

一つ一つの機能は、すべて関数に分けられています。

実際のコーディングの際は、`src/functions/<各自の機能の名前>.c` を編集してください。

他のファイルを編集すると、コンパイルが通らなくなる可能性があります。

4

検索系の機能で、検索しても見つからなかった場合は、エラーメッセージを出力し、必ず `return -1` として、負の値を返してください。

5

みなさんをお願いするすべての関数の中では、必ず次の文が入っています。

```
// ===== 変更しない =====  
int student_ids[101];  
char family_names[101][16];  
char family_name_readings[101][32];  
char given_names[101][16];  
char given_name_readings[101][32];  
char jhs_names[101][32];  
  
EXTRACT_ALL_COLUMNS;  
// ===== 変更しない =====
```

ここは、変更しないでください。変更するとコンパイルに通らなくなります。

6

みなさんにお願ひするすべての関数の中では、次の変数を使うことができます。

```
int number_of_students;
int student_ids[101];
char family_names[101][16];
char family_name_readings[101][32];
char given_names[101][16];
char given_name_readings[101][32];
char jhs_names[101][32];
```

それぞれを詳しく説明します。

- `number_of_students` (int)
読み込んだ生徒の数の合計の数です。
- `student_ids` (int[])
ファイルに保存されている生徒の出席番号が上から順に格納されています。**必ずしも出席番号の順番通りに格納されているとは限りません。**
- `family_names` (char[][])
ファイルに保存されている生徒ごとの漢字姓が上から順に格納されています。
- `family_name_readings` (char[][])
ファイルに保存されている生徒ごとの半角カナ姓が上から順に格納されています。
- `given_names` (char[][])
ファイルに保存されている生徒ごとの漢字名が上から順に格納されています。
- `given_name_readings` (char[][])
ファイルに保存されている生徒ごとの半角カナ名が上から順に格納されています。
- `jhs_names` (char[][])
ファイルに保存されている生徒ごとの出身中学校名が上から順に格納されています。
- `target_table` (struct student[101])
(構造体を扱う場合のみ必要) 生徒一人ひとりの構造体が配列として格納されています。

イメージとしては、次のようになっています。

	student_ids	family_names	family_name_readings	given_names	given_name_readings	jhs_names (インデックス)	
ファイルの1番目の生徒	2	山田	ヤマダ	次郎	ジロウ	A中	0
ファイルの2番目の生徒	4	佐藤	サトウ	三郎	サブロウ	A中	1
ファイルの3番目の生徒	1	鈴木	スズキ	四郎	シロウ	B中	2
ファイルの4番目の生徒	3	後藤	ゴトウ	裕翔	ユウト	A中	3
ファイルの5番目の生徒	5	南	ミナミ	優	ユウ	C中	4

このうち、□で囲われている列ごとがそれぞれの配列と対応しています。

例: ファイルの1番目の生徒の漢字姓名を出力する

```
// 二つの配列を使う例
printf("%s %sさん", family_names[0], given_names[0]);
// 構造体を使う例
printf("%s %sさん", target_table[0].family_name, target_table[0].given_name);
```

7

構造体のメンバについて

```
struct student {  
    int student_id;  
    char family_name[16];  
    char given_name[16];  
    char family_name_reading[32];  
    char given_name_reading[32];  
    char jhs_name[32];  
};
```

以上のメンバが利用できます。

8

コンパイル及び実行は、

```
gcc main.c
```

でおこなってください。リンクは自動で行われるため、今回はこのコマンドで問題ありません。

動作確認は、

```
gcc ./a.out
```

で行えます。

9

Windowsにgccが入っていない場合（前期の授業でpaiza.ioを使っていた場合）、Windowsにgccをインストールしてください。

参考: [windowsで快適なC/C++開発環境を作る - Qiita](#)

個別のお願い

find_all_students_from_jhs_name()

入力された値を基に、該当する人物の**出席番号・氏名**を一覧表示してください。

どう表示するかは任せます。また、検索は、**完全一致**としてください。

- `query (str[])`
ユーザから入力された中学校の名前が入っています。

find_name_from_number()

入力された値を基に、該当する人物の**氏名**を一覧表示してください。

どう表示するかは任せます。

- `target_id (int)`
ユーザから入力された出席番号が入っています。

find_number_from_name()

入力された値を基に、該当する人物の**出席番号**を一覧表示してください。

どう表示するかは任せます。

- `query (str[])`
ユーザから入力された半角カタカナの名前が入っています。
なお、**姓名の間はスペースで区切られています。**
注意してください。

int order_of_attendance_no()

出席番号順(昇順)に**出席番号、氏名、出身中学**一覧表示してください。

どう表示するかは任せますが、**氏名と出身中学校の配列は、必ずしも出席番号通りの並び方ではないことに注意してください。**

前期レポート課題の「並び替え」を参考にしてください。

int order_of_name()

名前順(昇順)に**出席番号、氏名、出身中学**一覧表示してください。

どう表示するかは任せますが、**氏名と出身中学校の配列は、必ずしも出席番号通りの並び方ではないことに注意してください。**

また、**並び替えの際は半角カタカナ**の文字コードを参考にしてください。

前期レポート課題の「並び替え」を参考にしてください。

姓のみの判断では並び替えとして不十分であることに注意してください。（同姓が居た場合。）