

# 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ではありますが、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。

前期の「プログラミングI」のみの知識で書けることを確認しています。

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）

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。全角文字の使用は禁止です。文字化けを回避するためです。

みなさんをお願い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タ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タ姓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。

イメージとしては、次のようになっています。

	<code>student_ids</code>	<code>family_names</code>	<code>family_name_readings</code>	<code>given_names</code>	<code>given_name_readings</code>	<code>jhs_names</code>	(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。

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。

コンパイル及び実行は、

```
gcc main.c
gcc ./a.out
```

でおこなってください。

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

## 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。

- `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**。

### 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。

姓のみでは並び替えができないことに注意してください。（同姓が居た場合。）