

課題Bを以下のプログラムで解決した。

```
// 2020.04.16 課題1-B
// task_1b.c
// Made by Taiki Watanabe(5SE-26)
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

typedef struct
{
    int id;
    char name[20];
    int math1;
    int math2;
    int sum;
} SCORE;

int compare_score(const void *p, const void *q)
{
    return (((SCORE *)q)->sum - ((SCORE *)p)->sum);
}

int main(void)
{
    FILE *fp;
    char fname[] = "SampleB.txt";
    SCORE data[50] = {0};
    SCORE *sp;
    sp = data;

    // ファイルの読み込みおよびエラー処理
    fp = fopen(fname, "r");

    if (fp == NULL)
    {
        printf("Can't open this file");
        exit(EXIT_FAILURE);
    }

    while (fscanf(fp, "%d\t%s\t%d\t%d", &sp->id, sp->name, &sp->math1, &sp->math2) != EOF)
    {
        sp->sum = sp->math1 + sp->math2;
        // printf("%d %s %d %d %d\n", sp->id, sp->name, sp->math1, sp->math2, sp->sum);
        sp++;
    }

    fclose(fp);

    // 構造体を点数が高い順にソート
    qsort(data, sizeof(data) / sizeof(data[0]), sizeof(SCORE), compare_score);

    // ソート後の1番目から1番高い点数を出力
    sp = data;
    printf("総合点が最も高かったのは、\n");
    while (data->sum == sp->sum)
    {
        printf("出席番号:%d番\t電気電子数学1:%d点\t電気電子数学2:%d点\t合計:%d点\n",
            sp->id, sp->math1, sp->math2, sp->sum);
        sp++;
    }

    return 0;
}
```

実行結果は以下ようになった。

```
~/Documents/Activities/学校/応用プログラミングB/1_20200416/b
> gcc task_1b.c -o task_1b

~/Documents/Activities/学校/応用プログラミングB/1_20200416/b
> ./task_1b
総合点が最も高かったのは、
出席番号:6番    電気電子数学1:100点    電気電子数学2:100点    合計:200点
```

## 考察

今回のプログラムでは、総合点が最も高いのは電気電子数学1, 2のどちらも満点の一人であった。しかし、最高点取得者が1人でない場合も考えられる。また、名前も同時に出力できるよう改良が必要になる場合も考えられる。

そこで、そういったケースに対応するため、構造体を利用し、総合点の高い順にソートする形で実装した。

ファイルからデータを読み取って、同じwhile文の中で比較する形に比べて、ソートの分だけ計算量が増えてしまうが、改良がしやすくデータの管理が容易になるよう実装できたのではないかと考察する。