

Buatlah sebuah program yang dapat melakukan *selection sort* dengan ketentuan yang terdapat pada PPT Exercise No. 4

Code

```
#include<stdio.h>
#include<string.h>
struct mhs{
    long long int nim; float gpa; char name[20];
} maha[20], gpaSort, nameSort;

void sort(int x){
    //Here's the difference between bubble & selection
    for(int i=0; i < x; i++){ //Moving each element
        for(int j=i+1; j < x; j++){ //Search for min. element
            if(maha[j].gpa < maha[i].gpa){
                //Swapping each element
                gpaSort = maha[j]; maha[j] = maha[i]; maha[i] = gpaSort;
            } else if(maha[i].gpa == maha[j].gpa){
                /*According to the Terms & Condition, so
                We used if nest to sort by name*/
                if(strcmp(maha[i].name, maha[j].name) > 0){
                    //Use strcmp to compare between char by char inside of array
                    nameSort = maha[j]; maha[j] = maha[i]; maha[i] = nameSort;
                }
            }
        }
    }
}

int main(){
    //Headline
    printf("\t\t\t\t\t==== C Program Student Sorting by GPA & Name =====\n");
    printf("\t\t\t\t\t\t\t\t\tCreated by: Pristian Budi D\n");
    printf("\t\t\t\t\t\t\t\t\tVer 2.19.20\n\n\n");

    //Variable declaring & Total Students
    int x, i; scanf("%d", &x, printf("How many students do u want to input? "));
    getchar();

    //Input Data
    for(i=0; i < x; i++){
        printf("\nStudent # %d: \n", i+1);
        scanf("%[^\\n]*%c", &maha[i].name, printf("Input Student Name\t\t: ")); getchar();
        scanf("%lld", &maha[i].nim, printf("Input Student ID\t\t: ")); getchar();
        scanf("%f", &maha[i].gpa, printf("Input Student GPA\t\t: ")); getchar();
    } printf("\n\n\t\t\t\t\tSorting...\n");

    //Calling Function
    sort(x);

    //Display Data
```

```
printf("\n\n=====\\n");
printf("Here's your student that have been sorted by GPA then by NAME :\\n");
printf("<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>\\n\\n");
printf("NO\\tNAME\\t\\t\\tID\\t\\t    GPA\\n");
printf("-----\\n\\n");
for(i=0; i < x; i++){
    printf("%d\\t%s\\t\\t\\t%lld\\t    %.2f\\n",i+1, maha[i].name, maha[i].nim, maha[i].gpa);
} printf("\\n\\n\\n"); return 0;
}
```

Program Output

```
D:\SUNIB\B25\SEMESTER 1\KOOLEEUH\ALGO N PROG\SESSION 21 & 22\Tugas\Proj.exe
===== C Program Student Sorting by GPA & Name =====
Created by: Pristian Budi D
Ver 2.19.20

How many students do u want to input? 10

Student #1:
Input Student Name      : Suzuki Kyoko
Input Student ID        : 2501983105
Input Student GPA       : 4.00

Student #2:
Input Student Name      : Adrian Richard
Input Student ID        : 2598127592
Input Student GPA       : 3.00

Student #3:
Input Student Name      : Andrea Weaso
Input Student ID        : 2502938230
Input Student GPA       : 3.50

Student #4:
Input Student Name      : Fujioka Kanae
Input Student ID        : 2512305923
Input Student GPA       : 3.25

Student #5:
Input Student Name      : Rich Meggie
Input Student ID        : 2514981275
Input Student GPA       : 4.00

Student #6:
Input Student Name      : Hasegawa Akira
Input Student ID        : 2502190480
Input Student GPA       : 3.87

Student #7:
Input Student Name      : Akasuki Chisaki
Input Student ID        : 2510923850
```

```
D:\SUNIB\B25\SEMESTER 1\KOOLEEUH\ALGO N PROG\SESSION 21 & 22\Tugas\Proj.exe
Input Student Name      : Akasuki Chisaki
Input Student ID        : 2510923850
Input Student GPA       : 3.87

Student #8:
Input Student Name      : Kawasaki Honda
Input Student ID        : 2510235123
Input Student GPA       : 4.00

Student #9:
Input Student Name      : Joko Anwar
Input Student ID        : 2412935700
Input Student GPA       : 4.00

Student #10:
Input Student Name      : Nina Lawson
Input Student ID        : 2512935800
Input Student GPA       : 3.50
```

Result

```

Sorting...

=====
Here's your student that have been sorted by GPA then by NAME : 
<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>

```

NO	NAME	ID	GPA
1	Adrian Richard	2598127592	3.00
2	Fujioka Kanae	2512305923	3.25
3	Andrea Weaso	2502938230	3.50
4	Nina Lawson	2512935800	3.50
5	Akasuki Chisaki	2510923850	3.87
6	Hasegawa Akira	2502190480	3.87
7	Joko Anwar	2412935700	4.00
8	Kawasaki Honda	2510235123	4.00
9	Rich Meggie	2514981275	4.00
10	Suzuki Kyoko	2501983105	4.00

```

-----
Process exited after 312.8 seconds with return value 0

```

Explanation

Program ini berjalan dari section Headline kemudian akan memberikan pertanyaan berapa banyak siswa yang ingin di-inputkan ke dalam file temporary/struct. Setelah memberikan sebuah angka untuk membuat sebuah looping agar kita dapat menginput sebanyak yang kita inginkan, program akan meminta data nama siswa, NIM/student ID, dan GPA dari tiap siswa.

Setelah kita memasukkan seluruh data itu tadi didalam looping. Kita memanggil sebuah fungsi dengan parameter dari banyaknya siswa yang ingin kita inputkan, yaitu variable x. Kemudian pada bagian fungsi sort() disitu saya menggunakan for nest untuk melakukan perbandingan dari 2 buah array yaitu array dari struct i dan struct j. Dikarenakan kita diminta untuk menggunakan *selection sort* pada program kali ini. Maksud dari struct i dan j adalah kita akan bergerak dari tiap-tiap elemen yang ada di dalam array i yang kemudian di compare dengan array j dimana j telah diberikan inisialisasi i+1 agar comparing bisa berjalan secara bersamaan pada elemen yang berbeda.

Setelah itu karena diminta untuk mengurutkan berdasarkan GPA maka, kita menggunakan if dimana jika array j < dari array i maka akan diberlakukan swap dengan cara memberikan nama struct yang berbeda lagi sehingga untuk program ini dibutuhkan 3 nama struct yaitu maha[20], gpaSort, dan nameSort. Setelah diberlakukan perbandingan program akan check apakah dari GPA tadi ada yang memiliki nilai sama? Jika ya, maka akan diurutkan berdasarkan nama. Dari logika ini maka dapat kita simpulkan kita membutuhkan sebuah if nest untuk melakukan logika ini. Pada bagian pengurutan nama kita memerlukan fungsi strcmp() untuk membandingkan tiap character di dalam array yang ada. Namun, perlu diingat bahwa pengurutan ini bersifat sensitif dimana huruf besar akan didahulukan dibandingkan huruf kecil. Contoh: Sapi dengan SBudi maka diurutkan SBudi terlebih dahulu kemudian Sapi.

Setelah memanggil fungsi `srot()` saya printkan sebuah "footer" agar program terlihat lebih menarik dan rapi dimana setelah footer saya membutuhkan sebuah looping lagi agar dapat melakukan print dari tiap data yang sudah terdata di dalam struct atau file temporary itu tadi. Karena pada dasarnya struct adalah file secara temporer.