

## ICS 103, Term 103

### Computer Programming in C

#### HW# 4 Solution

Due date: Friday, August 12, 2011

**Q.1.** Assume that a file contains students' ids, names and their scores. You are required to write a C program to do the following:

- (i) Ask the user to enter the input file name and read it. Read the data from the file and store it in three arrays, IDs, Names, Scores. The IDs should be declared as an array of integers, the Names as a two-dimensional array of characters and the Scores as an array of doubles. Assume that the maximum number of students that can be read from the file is 100 and the maximum length of the name of any student is 80 characters. Assume that names are stored based on first name and family name.
- (ii) Display the following menu to the user and read the entered choice:
  1. Sort data in **ascending** order according to students' IDs and then display it.
  2. Sort data in **ascending** order according to students' names and then display it.
  3. Sort data in **descending** order according to students' scores and then display it.
  4. Ask the user to enter a student ID and display his score
  5. Ask the user to enter a student name and display his score
  6. Exit the program

The program should keep displaying the menu until the user selects to exit from the program.

(iii) Implement each of the first five menu options as a separate function.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 40

void menu(void);
void Display(int IDS[], char NAMES[][80], double SCORES[], int count);
void SortID(int IDS[], char NAMES[][80], double SCORES[], int count);
void SortScores(int IDS[], char NAMES[][80], double SCORES[], int count);
void SortNames(int IDS[], char NAMES[][80], double SCORES[], int count);
void SearchID(int IDS[], double SCORES[], int count, int id);
void SearchName(char NAMES[][80], double SCORES[], int count, char name[]);
```

```

int main() {

    char filename[100];
    FILE *infile;
    int choice, status, count;
    int IDS[SIZE], id;
    double SCORES[SIZE];
    char NAMES[SIZE][80], first[80], last[80], full[80], name[80];

    printf("Enter the file name: ");
    scanf("%s",filename);
    infile = fopen(filename,"r");
    if (infile==NULL) { // to check if input file is opened properly or not
        printf("Sorry, input file %s is not found", filename);
        system("pause");
        exit(1); // terminates the program
    }

    count=0;
    status = fscanf(infile, "%d%s%s%lf", &IDS[count], first, last, &SCORES[count]);
    while (status != EOF){
        strcpy(full, first);
        strcat(full, " ");
        strcat(full, last);
        strcpy(NAMES[count], full);
        count++;
        status = fscanf(infile, "%d%s%s%lf", &IDS[count], first, last, &SCORES[count]);
    }

    do {

        menu();
        scanf("%d",&choice);
        switch (choice){

            case 1: SortID(IDS, NAMES, SCORES, count);
                Display(IDS, NAMES, SCORES, count);
                break;

            case 2: SortNames(IDS, NAMES, SCORES, count);
                Display(IDS, NAMES, SCORES, count);
                break;

            case 3: SortScores(IDS, NAMES, SCORES, count);
                Display(IDS, NAMES, SCORES, count);
                break;

            case 4: printf("Enter the student ID: ");
                scanf("%d", &id);

```

```

        SearchID(IDS, SCORES, count, id);
        break;

    case 5: printf("Enter the student name: ");
            fflush(stdin);
            gets(name);
            SearchName(NAMES, SCORES, count, name);
            break;

    case 6: system("pause"); return 0;

    default: printf("Invalid choice...\n");
            }
} while (choice !=6);

system("pause");
return 0;
}

void menu(void){

printf("1. Sort data in ascending order according to students' IDs and then display it\n");
printf("2. Sort data in ascending order according to students' names and then display it\n");
printf("3. Sort data in descending order according to students' scores and then display it\n");
printf("4. Ask the user to enter a student ID and display his score\n");
printf("5. Ask the user to enter a student name and display his score\n");
printf("6. Exit the program\n");

}

void Display(int IDS[], char NAMES[][80], double SCORES[], int count){

    printf("\nID\tName\tScore\n");
    for (int i=0; i< count; i++)
        printf("%-9d\t%-20s\t%6.2f\n", IDS[i],NAMES[i],SCORES[i]);

    printf("\n");
}

void SortID(int IDS[], char NAMES[][80], double SCORES[], int count){

    int i, pass = 1, swap_occurs, tempi;
    char tempn[80];
    double temps;

    do{
        swap_occurs = 0;

```

```

    for(i = 1; i <= count - pass; i++) {
        if (IDS[i-1] > IDS[i]) {
            tempi=IDS[i-1]; IDS[i-1]=IDS[i]; IDS[i]=tempi;
            temps=SCORES[i-1]; SCORES[i-1]=SCORES[i]; SCORES[i]=temps;
            strcpy(tempn, NAMES[i-1]);
            strcpy(NAMES[i-1], NAMES[i]);
            strcpy(NAMES[i], tempn);
            swap_occurs = 1;
        }
    }
    pass++;
} while (swap_occurs && pass <= count-1);

}

```

```

void SortScores(int IDS[], char NAMES[][80], double SCORES[], int count){

```

```

    int i, pass = 1, swap_occurs, tempi;
    char tempn[80];
    double temps;

    do{
        swap_occurs = 0;
        for(i = 1; i <= count - pass; i++) {
            if (SCORES[i-1] < SCORES[i]) {
                tempi=IDS[i-1]; IDS[i-1]=IDS[i]; IDS[i]=tempi;
                temps=SCORES[i-1]; SCORES[i-1]=SCORES[i]; SCORES[i]=temps;
                strcpy(tempn, NAMES[i-1]);
                strcpy(NAMES[i-1], NAMES[i]);
                strcpy(NAMES[i], tempn);
                swap_occurs = 1;
            }
        }
        pass++;
    } while (swap_occurs && pass <= count-1);

}

```

```

void SortNames(int IDS[], char NAMES[][80], double SCORES[], int count){

```

```

    int i, pass = 1, swap_occurs, tempi;
    char tempn[80];
    double temps;

    do{
        swap_occurs = 0;
        for(i = 1; i <= count - pass; i++) {
            if (strcmp(NAMES[i-1], NAMES[i])>0) {
                tempi=IDS[i-1]; IDS[i-1]=IDS[i]; IDS[i]=tempi;

```

```

        temps=SCORES[i-1]; SCORES[i-1]=SCORES[i]; SCORES[i]=temps;
        strcpy(tempn, NAMES[i-1]);
        strcpy(NAMES[i-1], NAMES[i]);
        strcpy(NAMES[i], tempn);
        swap_occurs = 1;
    }
}
pass++;
} while (swap_occurs && pass <= count-1);
}

```

```

void SearchID(int IDS[], double SCORES[], int count, int id){

```

```

    int i=0, found = 0;

    while (!found && i < count) {
        if (IDS[i] == id)
            found = 1;
        else
            i++;
    }

    if (found)
        printf("The student with ID %d has a score of %6.2f\n\n", id, SCORES[i]);
    else
        printf("The student with ID %d is not found\n\n", id);
}

```

```

void SearchName(char NAMES[][80], double SCORES[], int count, char name[]){

```

```

    int i=0, found = 0;

    while (!found && i < count) {
        if (!strcmp(NAMES[i], name))
            found = 1;
        else
            i++;
    }

    if (found)
        printf("The student %s has a score of %6.2f\n\n", name, SCORES[i]);
    else
        printf("The student %s is not found\n\n", name);
}

```