Simple grade management system

course work report of C

Class Num: 2017215120

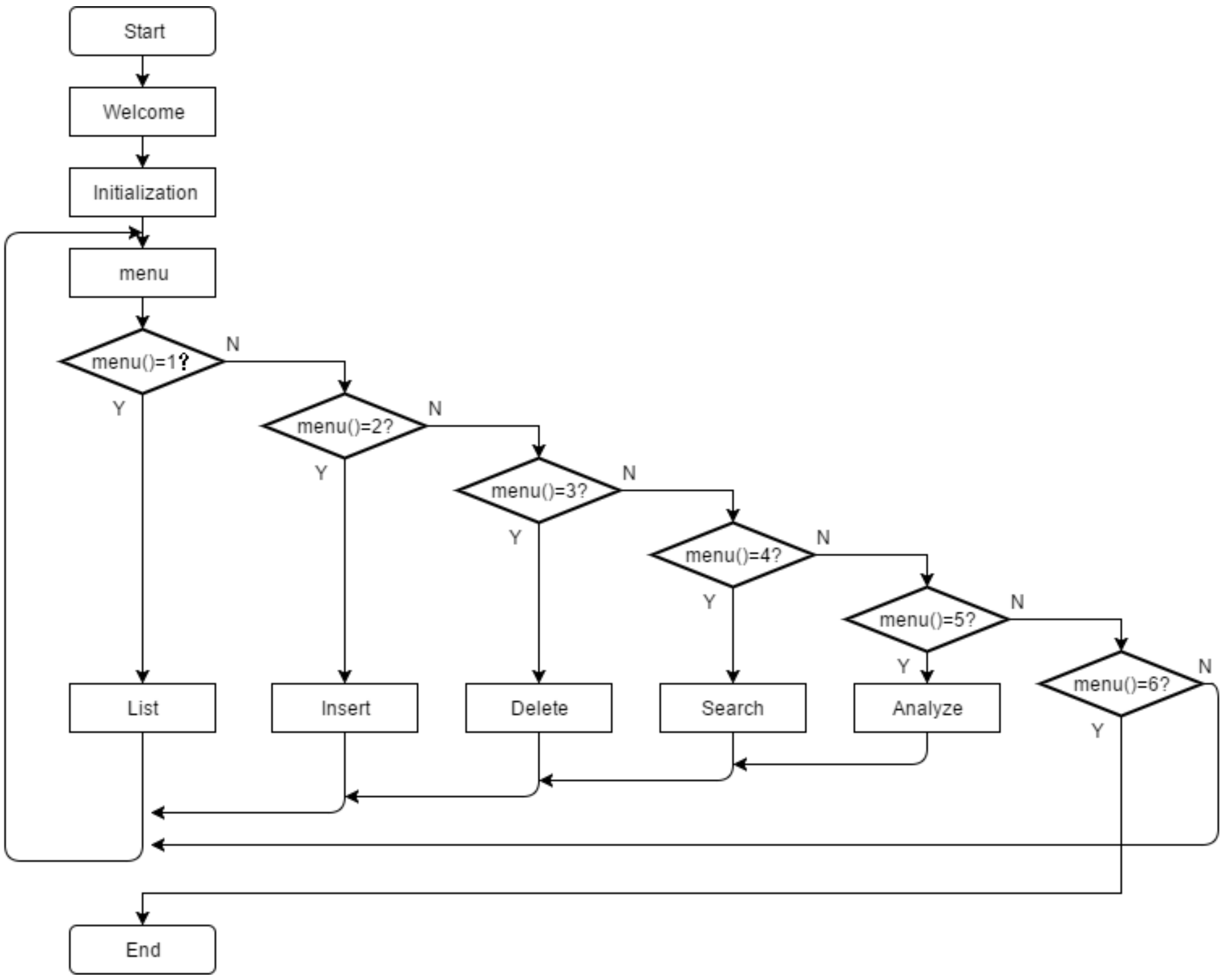
Name: LIU Zekuan

Student Num: 2017213176

2017 autumn

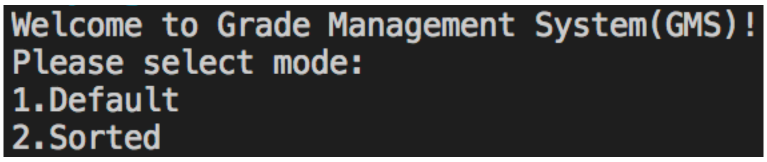
Content

1.Flowchart of main function



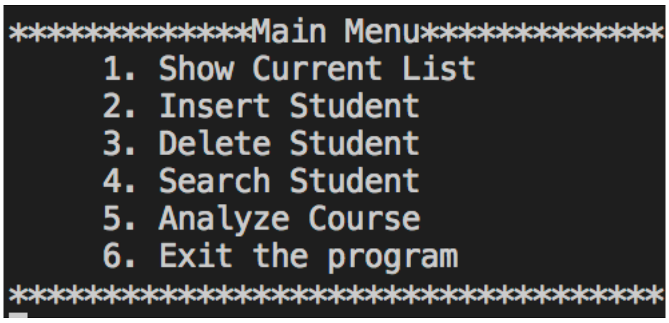
2.Screenshots of program execution

2.1 Welcome



```
Welcome to Grade Management System(GMS)!
Please select mode:
1.Default
2.Sorted
```

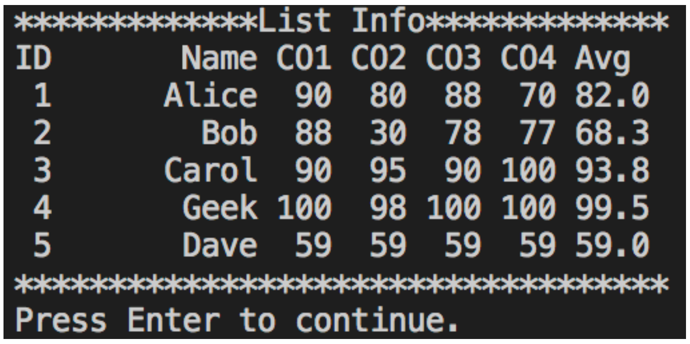
2.2 Main Menu



```
*************Main Menu*************
     1. Show Current List
     2. Insert Student
     3. Delete Student
     4. Search Student
     5. Analyze Course
     6. Exit the program
***********************************
```

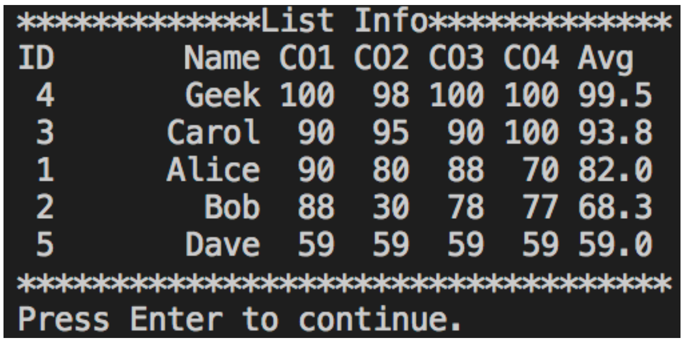
2.3 Show Current List

2.3.1 Default



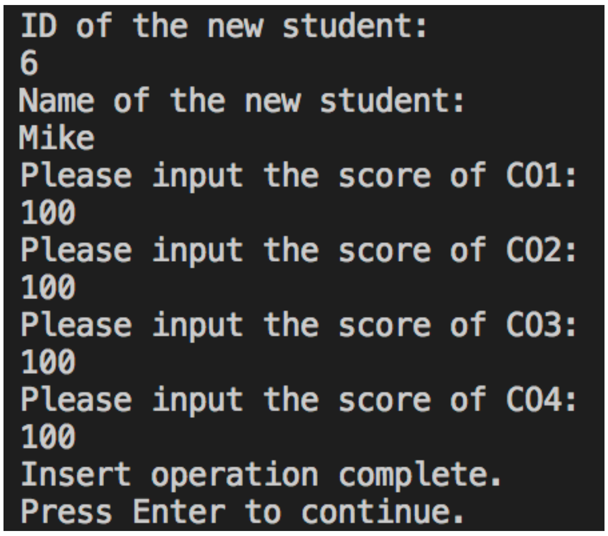
```
*************List Info*************
ID        Name CO1 CO2 CO3 CO4 Avg
 1       Alice  90  80  88  70 82.0
 2         Bob  88  30  78  77 68.3
 3       Carol  90  95  90 100 93.8
 4        Geek 100  98 100 100 99.5
 5        Dave  59  59  59  59 59.0
***********************************
Press Enter to continue.
```

2.3.2 Sorted



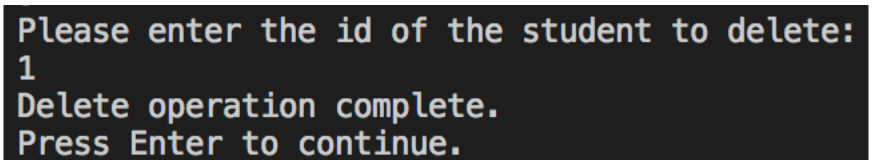
```
*************List Info*************
ID        Name CO1 CO2 CO3 CO4 Avg
 4        Geek 100  98 100 100 99.5
 3       Carol  90  95  90 100 93.8
 1       Alice  90  80  88  70 82.0
 2         Bob  88  30  78  77 68.3
 5        Dave  59  59  59  59 59.0
***********************************
Press Enter to continue.
```

2.4 Insert Student



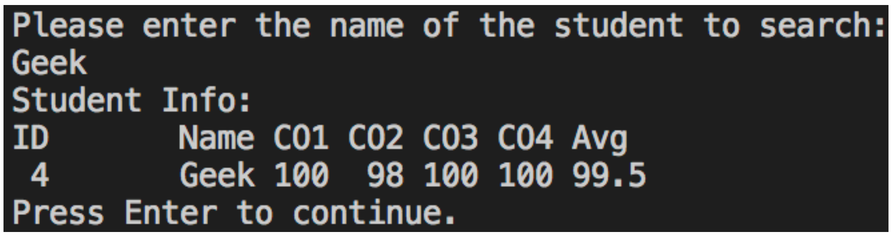
```
ID of the new student:
6
Name of the new student:
Mike
Please input the score of CO1:
100
Please input the score of CO2:
100
Please input the score of CO3:
100
Please input the score of CO4:
100
Insert operation complete.
Press Enter to continue.
```

2.5 Delete Student



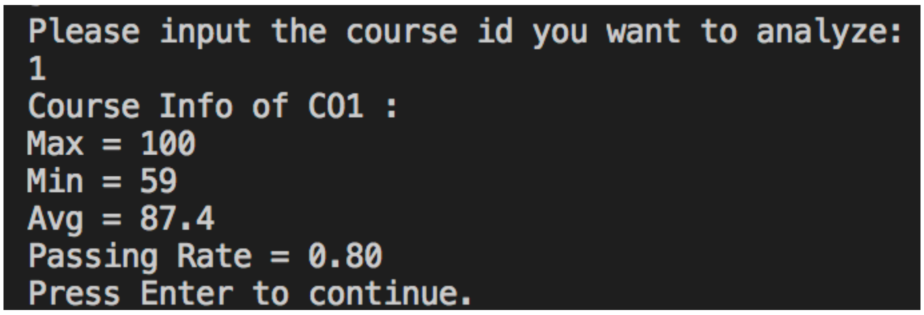
```
Please enter the id of the student to delete:
1
Delete operation complete.
Press Enter to continue.
```

2.6 Search Student



```
Please enter the name of the student to search:
Geek
Student Info:
ID       Name CO1 CO2 CO3 CO4 Avg
 4       Geek 100  98 100 100 99.5
Press Enter to continue.
```

2.7 Analyze Course



```
Please input the course id you want to analyze:
1
Course Info of CO1 :
Max = 100
Min = 59
Avg = 87.4
Passing Rate = 0.80
Press Enter to continue.
```

3. Source code

```
#include<stdio.h>
#include<string.h>
#define NAMELEN 10
#define COURSES 4

typedef struct stu_info{
    char stu_name[NAMELEN];
    int id, score[COURSES];
    float avgScore;
    struct stu_info *next;
} STU_INFO;

char mode=0;
STU_INFO *headPtr=NULL;

int main(){
    int welcome();
    int initialization();
    int menu();
    int list();
    int insert();
    int delete();
    int search();
    int analyze();

    welcome();

    initialization();

    do{
        switch(menu()){
            case 1:
                list();
                break;
            case 2:
                insert();
                break;
            case 3:
                delete();
                break;
            case 4:
                search();
                break;
            case 5:
                analyze();
                break;
            case 6:
                goto exit;
        }
    }while(1);

    exit:

    return 0;
}

int welcome(){
    printf("Welcome to Grade Management System(GMS)!\n");
    do{
        printf("Please select mode:\n1.Default\n2.Sorted\n");
        scanf("%c",&mode);
        getchar();
        if(mode=='1'||mode=='2'){
            mode-='1';
            break;
        }
        printf("Input Error!\n");
    }while(1);

    if(mode){
        printf("Sorted Mode:\n");
    }else{
        printf("Default Mode:\n");
    }
    return 0;
}

int initialization(){
    FILE* fp = fopen("a.in","r");
    if(fp){
        int idt=0,scoret[COURSES]={0};
        char namet[NAMELEN];
        float avgt=0;
        STU_INFO *lastPtr=NULL,*currentPtr=NULL;

        for(int i=0;i<5;i++){
            fscanf(fp,"%d %s %d %d %d %d %f",&idt,namet,&scoret[0],&scoret[1],&scoret[2],&scoret[3],&avgt);
            currentPtr=(STU_INFO*)malloc(sizeof(STU_INFO));
            if(currentPtr!=NULL){
                strcpy(currentPtr->stu_name,namet);
                currentPtr->id=idt;
                for(int i=0;i<4;i++){
                    currentPtr->score[i]=scoret[i];
                }
                currentPtr->avgScore=avgt;
                currentPtr->next=NULL;
                if(headPtr==NULL){
                    headPtr=currentPtr;
                    lastPtr=currentPtr;
                }else{
                    lastPtr->next=currentPtr;
                    lastPtr=currentPtr;
                }
            }
        }
        fclose(fp);
    }else{
        printf("File Open Error!\n");
    }
    return 0;
}

int menu(){
    int option=0;
    char i=0;
    printf("************Main Menu************\n");
    printf("     1. Show Current List\n");
    printf("     2. Insert Student\n");
    printf("     3. Delete Student\n");
    printf("     4. Search Student\n");
    printf("     5. Analyze Course\n");
    printf("     6. Exit the program\n");
    printf("************************************\n");
    scanf("%c",&i);
    option=(int)(i-'0');
    while(option<1||option>6){
        printf("Input Error!\nPlease retype:\n");
        scanf("%d",&option);
    }
    return option;
}

int sort(){
    STU_INFO *p=headPtr,*q=headPtr,*t=malloc(sizeof(STU_INFO));
    for(p=headPtr;p->next!=NULL;p=p->next){
        for(q=headPtr;q->next!=NULL;q=q->next){
            if(q->avgScore<q->next->avgScore){
                strcpy(t->stu_name,q->next->stu_name);
                t->id=q->next->id;
                for(int i=0;i<4;i++){
                    t->score[i]=q->next->score[i];
                }
                t->avgScore=q->next->avgScore;

                strcpy(q->next->stu_name,q->stu_name);
                q->next->id=q->id;
                for(int i=0;i<4;i++){
                    q->next->score[i]=q->score[i];
                }
                q->next->avgScore=q->avgScore;

                strcpy(q->stu_name,t->stu_name);
                q->id=t->id;
                for(int i=0;i<4;i++){
                    q->score[i]=t->score[i];
                }
                q->avgScore=t->avgScore;
            }
        }
    }

    return 0;
}
int list(){
    if(mode){
        sort();
    }
    printf("*************List Info*************\n");
    printf("ID       Name CO1 CO2 CO3 CO4 Avg \n");
    STU_INFO *currentPtr=headPtr;
    while(currentPtr!=NULL){
        printf("%2d %10s %3d %3d %3d %3d %-.1f\n",currentPtr->id,currentPtr->stu_name,currentPtr-
>score[0],currentPtr->score[1],currentPtr->score[2],currentPtr->score[3],currentPtr->avgScore);
        currentPtr=currentPtr->next;
    }
    printf("************************************\n");
    printf("Press Enter to continue.\n");
    getchar();getchar();
    return 0;
}

int insert(){
    int idt=0,scoret[COURSES]={0};
    char namet[NAMELEN];
    float avgt=0;
    do{
        if(idt!=0){
            printf("Input Error!\n");
        }
        printf("ID of the new student:\n");
        scanf("%d",&idt);
    }while(idt<1||idt>99);
    printf("Name of the new student:\n");
    scanf("%s",namet);
    do{
        if(scoret[0]!=0){
            printf("Input Error!\n");
        }
        printf("Please input the score of CO1:\n");
        scanf("%d",&scoret[0]);
    }while(scoret[0]<0||scoret[0]>100);
    do{
        if(scoret[1]!=0){
            printf("Input Error!\n");
        }
        printf("Please input the score of CO2:\n");
        scanf("%d",&scoret[1]);
        }while(scoret[1]<0||scoret[1]>100);
    do{
        if(scoret[2]!=0){
            printf("Input Error!\n");
        }
        printf("Please input the score of CO3:\n");
        scanf("%d",&scoret[2]);
    }while(scoret[2]<0||scoret[2]>100);
    do{
        if(scoret[3]!=0){
            printf("Input Error!\n");
        }
        printf("Please input the score of CO4:\n");
        scanf("%d",&scoret[3]);
    }while(scoret[3]<0||scoret[3]>100);
    avgt=(float)(scoret[0]+scoret[1]+scoret[2]+scoret[3])/4;
    STU_INFO *currentPtr=NULL,*Ptrt=NULL,*newPtr=NULL;
    for(currentPtr=headPtr;currentPtr!=NULL&&currentPtr->id<idt;currentPtr=currentPtr->next)
        Ptrt=currentPtr;
    newPtr=malloc(sizeof(STU_INFO));
    newPtr->id=idt;
    strcpy(newPtr->stu_name,namet);
    newPtr->avgScore=avgt;
    for(int i=0;i<4;i++){
        newPtr->score[i]=scoret[i];
    }
    newPtr->next=currentPtr;
    if(Ptrt!=NULL){
        Ptrt->next=newPtr;
    }else{
        headPtr=newPtr;
    }
    printf("Insert operation complete.\n");
    printf("Press Enter to continue.\n");
    getchar();getchar();
    return 0;
}

int delete(){
    int idt=0,flag=0;
    do{
        printf("Please enter the id of the student to delete:\n");
        scanf("%d",&idt);
        for(STU_INFO *currentPtr=headPtr,*Ptrt=headPtr;currentPtr!=NULL;currentPtr=currentPtr->next){
            if(currentPtr->id==idt){
                if(currentPtr==headPtr){
                    headPtr=currentPtr->next;
                    Ptrt=headPtr;
                    flag++;
                }else{
                    Ptrt->next=currentPtr->next;
                    flag++;
                }
                free(currentPtr);
                currentPtr=Ptrt;
            }
            Ptrt=currentPtr;
        }
        if(flag==0){
            printf("Id not found\n");
        }
    }while(flag==0);
    printf("Delete operation complete.\n");
    printf("Press Enter to continue.\n");
    getchar();getchar();
    return 0;
}

int search(){
    char namet[NAMELEN];
    do{
        printf("Please enter the name of the student to search:\n");
        scanf("%s",namet);
        for(STU_INFO *currentPtr=headPtr,*Ptrt=headPtr;currentPtr!=NULL;currentPtr=currentPtr->next){
            if(strcmp(currentPtr->stu_name,namet)==0){
                printf("Student Info:\n");
                printf("ID       Name CO1 CO2 CO3 CO4 Avg \n");
                printf("%2d %10s %3d %3d %3d %3d %-.1f\n",currentPtr->id,currentPtr->stu_name,currentPtr-
>score[0],currentPtr->score[1],currentPtr->score[2],currentPtr->score[3],currentPtr->avgScore);
                goto out;
            }
        }
        printf("Id not found\n");
    }while(1);
    out:
    printf("Press Enter to continue.\n");
    getchar();getchar();
    return 0;
}

int analyze(){
    int course=0;
    int max=0,min=100;
    int pass=0,cnt=0,sum=0;
    do{
        printf("Please input the course id you want to analyze:\n");
        scanf("%d",&course);
        if(course<4&&course>0){
            break;
        }
        printf("Input Error!\n");
    }while(1);
    course--;
    for(STU_INFO *currentPtr=headPtr;currentPtr!=NULL;currentPtr=currentPtr->next){
        sum+=currentPtr->score[course];
        if(max<currentPtr->score[course]){
            max=currentPtr->score[course];
        }
        if(min>currentPtr->score[course]){
            min=currentPtr->score[course];
        }
        if(currentPtr->score[course]>60){
            pass++;
        }
        cnt++;
    }
    printf("Course Info of CO%d :\n",course+1);
    printf("Max = %d\n",max);
    printf("Min = %d\n",min);
    printf("Avg = %.1f\n",(float)sum/cnt);
    printf("Passing Rate = %.2f\n",(float)pass/cnt);
    printf("Press Enter to continue.\n");
    getchar();getchar();
    return 0;
}
```