第十次上机

PB17081531 沈鹏飞

```
#include<stdio.h>
#include<Windows.h>

typedef struct Student
{
    int id;
    char name[20];
    double course1;
    double course2;
    double course3;
    double average;
} Student;




Student stu_arr[5];

void calc_ave()
{
    for (int i = 0; i < 5; i++)
    {
        stu_arr[i].average = 1 / 3.0*(stu_arr[i].course1 + stu_arr[i].course2 +
stu_arr[i].course3);
    }
}

void swap(Student *student1, Student * student2)
{
    int temp_id = student1->id;
    student1->id = student2->id;
    student2->id = temp_id;

    char temp[20];

    strcpy(temp, student1->name);
    strcpy(student1->name, student2->name);
    strcpy(student2->name, temp);

    double temp_course;

    temp_course = student1->course1;
    student1->course1 = student2->course1;
    student2->course1 = temp_course;

    temp_course = student1->course2;
    student1->course2 = student2->course2;
    student2->course2 = temp_course;
```

```
    temp_course = student1->course3;
    student1->course3 = student2->course3;
    student2->course3 = temp_course;

    temp_course = student1->average;
    student1->average = student2->average;
    student2->average = temp_course;
}
//
void sort(Student *student_arr)
{
    calc_ave();
    for (int i = 0; i < 5; i++)
    {
        int max = student_arr[i].average;
        int index = i;
        for (int j = i + 1; j < 5; j++)
        {
            if (student_arr[j].average > max)
            {
                max = student_arr[j].average;
                index = j;
            }
        }
        swap(student_arr + i, student_arr + index);
    }
}


void print(FILE * ptr)
{
    for (int i = 0; i < 5; i++)
    {
        fprintf(ptr, "%d\n%s\n%f\n%f\n%f\n", stu_arr[i].id, stu_arr[i].name,
stu_arr[i].course1, stu_arr[i].course2, stu_arr[i].course3);
    }
}

int main()
{
    FILE *fp1, *fp2;

    if ((fp1 = fopen("C:\\Users\\10069\\Desktop\\stud_dat.txt", "r")) == NULL)
    {
        printf("File not opened!\n");
        exit(0);
    }

    if ((fp2 = fopen("C:\\Users\\10069\\Desktop\\stud_sort.txt", "w")) == NULL)
    {
        printf("File not opened!\n");
        exit(0);
    }

    int i = 0;
    while (fscanf(fp1, "%d%s%lf%lf%lf", &(stu_arr[i].id), stu_arr[i].name, &
(stu_arr[i].course1), &(stu_arr[i].course2), &(stu_arr[i].course3)) != EOF)
    {
```

```
        i++;
    }


    sort(stu_arr);

    print(fp2);

    fclose(fp1);
    fclose(fp2);

    system("pause");

}
```

```
#include<stdio.h>
#include<Windows.h>

typedef struct Student
{
    int id;
    char name[20];
    double course1;
    double course2;
    double course3;
    double average;
} Student;




Student stu_arr[5];

void calc_ave()
{
    for (int i = 0; i < 5; i++)
    {
        stu_arr[i].average = 1 / 3.0*(stu_arr[i].course1 + stu_arr[i].course2 +
stu_arr[i].course3);
    }
}

void swap(Student *student1, Student * student2)
{
    int temp_id = student1->id;
    student1->id = student2->id;
    student2->id = temp_id;

    char temp[20];

    strcpy(temp, student1->name);
    strcpy(student1->name, student2->name);
    strcpy(student2->name, temp);

    double temp_course;
```

```
    temp_course = student1->course1;
    student1->course1 = student2->course1;
    student2->course1 = temp_course;

    temp_course = student1->course2;
    student1->course2 = student2->course2;
    student2->course2 = temp_course;

    temp_course = student1->course3;
    student1->course3 = student2->course3;
    student2->course3 = temp_course;

    temp_course = student1->average;
    student1->average = student2->average;
    student2->average = temp_course;
}
//
void sort(Student *student_arr)
{
    calc_ave();
    for (int i = 0; i < 5; i++)
    {
        int max = student_arr[i].average;
        int index = i;
        for (int j = i + 1; j < 5; j++)
        {
            if (student_arr[j].average > max)
            {
                max = student_arr[j].average;
                index = j;
            }
        }
        swap(student_arr + i, student_arr + index);
    }
}


void print(FILE * ptr)
{
    for (int i = 0; i < 5; i++)
    {
        fprintf(ptr, "%d\n%s\n%f\n%f\n%f\n", stu_arr[i].id, stu_arr[i].name,
stu_arr[i].course1, stu_arr[i].course2, stu_arr[i].course3);
    }
}

int main()
{
    FILE *fp1;

    if ((fp1 = fopen("C:\\Users\\10069\\Desktop\\stud_sort.txt", "r")) == NULL)
    {
        printf("File not opened!\n");
        exit(0);
    }

    int i = 0;
```

```
    while (fscanf(fp1, "%d%s%lf%lf%lf", &(stu_arr[i].id), stu_arr[i].name, &
(stu_arr[i].course1), &(stu_arr[i].course2), &(stu_arr[i].course3)) != EOF)
    {
        i++;
    }

    print(stdout);

    fclose(fp1);

    system("pause");

}
```

```
#include<stdio.h>
#include<stdlib.h>

typedef struct stu
{
    int num;
    char name[16];
    char sex;
    int age;
    double grade;
    struct stu * next;
}stu;

char buffer[10];

stu* insert_head(stu* head, FILE * fp, FILE * ofp)
{
    stu* new_node = (stu*)malloc(sizeof(stu));
    new_node->next = head;

    if(ofp) fprintf(ofp , "The id:");
    fscanf(fp , "%d", &new_node->num);

    if(ofp) fprintf(ofp , "The name:");
    fscanf(fp , "%s", &new_node->name);

    if(ofp) fprintf(ofp , "The sex:");
    if (fp==stdin)
    {
        getchar();
    }
    fscanf(fp , "%s", buffer);
    new_node->sex = buffer[0];

    if (new_node->sex == 'W')
    {
        new_node->sex = 0;
    }

    if(ofp) fprintf(ofp , "The grade:");
    fscanf(fp , "%lf", &new_node->grade);
```

```
    if(ofp) fprintf(ofp , "The age:");
    fscanf(fp , "%d", &new_node->age);

    return new_node;
}

void print(stu * head,FILE * fp)
{

    while (head)
    {
        fprintf(fp , "%10d%15s", head->num, head->name);
        if (head->sex == 0)
        {
            fprintf(fp , "    W    ");
        }
        else
        {
            fprintf(fp , "    M    ");
        }

        fprintf(fp , "%8.1lf%5d", head->grade, head->age);

        fprintf(fp , "\n");
        head = head->next;
    }
    fprintf(fp , "\n");
}

stu* delete_one(stu* head)
{
    int temp_age = 0;
    printf("The age that you want to delete:");
    scanf("%d", &temp_age);

    stu* temp_head = head;

    if (head == NULL)
    {
        return head;
    }

    if (head->next == NULL)
    {
        if (head->age == temp_age)
        {
            free(head);
            head = NULL;
        }
        return head;
    }

    while (head->next)
    {
        if (head->age == temp_age && temp_head == head)
        {
            head = head->next;
```

```
            free(temp_head);
            temp_head = head;
            continue;
        }
        if ((head->next->age) == temp_age)
        {
            stu * temp = head->next;
            head->next = head->next->next;
            free(temp);
        }
        else
        {
            head = head->next;
        }
    }
    print(temp_head,stdout);
    return temp_head;
}

stu * read_from_file(FILE * fp)
{
    stu * head = NULL;

    while (!feof(fp))
    {
        head = insert_head(head, fp, NULL);
    }

    stu * ptr = head;
    head = head->next;
    free(ptr);

    return head;
}

int main()
{
    stu * head = NULL;

    FILE * fpw, *fpr;

    int choice;
    printf("1.Insert a student\n2.Print all the information\n3.Delete someone
with their age\n4.Save to file\n5.Build from data\n");
    while (scanf("%d", &choice))
    {
        switch (choice)
        {

        case 1:
            head = insert_head(head, stdin, stdout);
            break;
        case 2:
            print(head,stdout);
            break;
        case 3:
            head = delete_one(head);
            break;
```

```
        case 4:
            if ((fpw = fopen("C:\\Users\\10069\\Desktop\\stud_list.txt", "w"))
== NULL)
            {
                printf("File not opened!\n");
            }
            print(head, fpw);
            fflush(fpw);
            break;
        case 5:
            while (head)
            {
                stu * temp = head;
                head = head->next;
                free(temp);
            }
            if ((fpr = fopen("C:\\Users\\10069\\Desktop\\stud_list.txt", "r"))
== NULL)
            {
                printf("No data yet\n");
            }
            head = read_from_file(fpr);
            break;
        default:
            break;
        }
        printf("1.Insert a student\n2.Print all the information\n3.Delete
someone with their age\n4.Save to file\n5.Build from data\n");
    }


}
```