第八次上机

PB17081531 沈鹏飞

第一题

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

#define N 10

typedef struct {
    int year;
    int month;
    int day;
} Date;


typedef struct BOOK{
    int id;
    char name[20];
    char author[20];
    char press[20];
    double price;
    Date date;
} BOOK;

BOOK books[N];

void swap(BOOK *book1, BOOK * book2)
{
    int temp_id = book1->id;
    book1->id = book2->id;
    book2->id = temp_id;

    char temp[20];

    strcpy(book1->name, temp);
    strcpy(book2->name,book1->name);
    strcpy(temp,book2->name);

    strcpy(book1->author, temp);
    strcpy(book2->author, book1->author);
    strcpy(temp, book2->author);
    strcpy(book1->press, temp);
    strcpy(book2->press, book1->press);
    strcpy(temp, book2->press);
    strcpy(book1->name, temp);


    double temp_price;

    temp_price = book1->price;
    book1->price = book2->price;
    book2->price = temp_price;

    int temp_int;


    temp_id = book1->date.year;
    book1->date.year = book2->date.year;
    book2->date.year = temp_id;

    temp_id = book1->date.month;
    book1->date.month = book2->date.month;
    book2->date.month = temp_id;
    temp_id = book1->date.day;
    book1->date.day = book2->date.day;
    book2->date.day = temp_id;
}

void sort(BOOK *book_arr)
{
    for (int i = 0; i < N; i++)
    {
        int max = book_arr[i].id;
        int index = i;
        for (int j = i + 1; j < N; j++)
        {
            if (book_arr[j].id > max)
            {
                max = book_arr[j].id;
                index = j;
            }
        }
        swap(book_arr + i, book_arr + index);
    }
}

int main()
{
    for (int i = 0; i < N; i++)
    {
        books[i].id = rand() % 100;
    }

    sort(books);

    for (int i = 0; i < N; i++)
    {
        printf("%5d", books[i].id);
    }
    system("pause");
}
```

第二题

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

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

stu* insert_head(stu* head)
{
    stu* new_node = (stu*)malloc(sizeof(stu));
    new_node->next = head;

    printf("The name:");
    scanf("%s", &new_node->name);

    printf("The id:");
    scanf("%d", &new_node->num);

    printf("The sex:");
    scanf("%d", &new_node->sex);

    printf("The age:");
    scanf("%d", &new_node->age);

    printf("The grade:");
    scanf("%lf", &new_node->grade);

    return new_node;
}

void print(stu * head)
{

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

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

        printf("\n");
        head = head->next;
    }
    printf("\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);
    return temp_head;
}

int main()
{
    stu * head = NULL;

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

        case 1:
            head = insert_head(head);
            break;
        case 2:
            print(head);
            break;
        case 3:
            head=delete_one(head);
        default:
            break;
        }
        printf("1.Insert a student\n2.Print all the information\n3.Delete someone with their age\n");
    }

}
```

第三题

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

typedef struct Book {
    int id;
    char name[20];
    char author[20];
    char press[20];
    double price;
    struct Book * next;
}BOOK;



BOOK* insert_head(BOOK* head)
{
    BOOK* new_node = (BOOK*)malloc(sizeof(BOOK));
    new_node->next = head;

    printf("The name:");
    scanf("%s", &new_node->name);

    printf("The id:");
    scanf("%d", &new_node->id);

    printf("The author:");
    scanf("%s", &new_node->author);

    printf("The press:");
    scanf("%s", &new_node->press);

    printf("The price:");
    scanf("%lf", &new_node->price);

    return new_node;
}

void print(BOOK * head)
{
    while (head)
    {
        printf("%10d%15s", head->id, head->name);

        printf("%10s", head->author);

        printf("%8.1lf%10s", head->price, head->press);

        printf("\n");
        head = head->next;
    }
    printf("\n");
}

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

    BOOK* temp_head = head;

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

    if (head->next==NULL)
    {
        if (head->id==temp_id)
        {
            free(head);
            head = NULL;
        }
        return head;
    }

    while (head->next)
    {
        if (head->id==temp_id&&temp_head==head)
        {
            head = head->next;
            free(temp_head);
            temp_head = head;
            continue;
        }
        if ((head->next->id)==temp_id)
        {
            BOOK * temp = head->next;
            head->next = head->next->next;
            free(temp);
        }
        else
        {
            head = head->next;
        }
    }
    return temp_head;

}

int search(BOOK* head,int given_id)
{
    while ( head)
    {
        if (head->id==given_id)
        {
            return 1;
        }
        else
        {
            head = head->next;
        }
    }
    return 0;
}

int main()
{
    BOOK * head = NULL;

    int choice;
    printf("1.Insert a BOOK\n2.Print all the information\n3.Delete someone with their id\n4.Search for an item\n");
    while (scanf("%d",&choice))
    {
        switch (choice)
        {

        case 1:
            head = insert_head(head);
            break;
        case 2:
            print(head);
            break;
        case 3:
            head=delete_one(head);
            break;
        case 4:
            printf("The id wanted:");
            scanf("%d", &choice);

            if (search(head,choice))
            {
                printf("Found!\n");
            }
            else
            {
                printf("Not found!\n");
            }

        default:
            break;
        }
        printf("1.Insert a BOOK\n2.Print all the information\n3.Delete someone with their id\n4.Search for an item\n");
    }

}
```