参考代码

其他题目没找到眼前一亮的代码就不放了，基本都这样。

智能车**切忌闭门造车**，大家可以多多**交流讨论。**

但是**不要**互传答案，可以**交流思路。**

第六题

1. 结构、逻辑完整

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

#define MAX_STUDENTS 100

typedef struct {
    char name[21];
    char dormitory[4];
    char departure[8];
    char back[8];
    int departure_day;
    int back_day;      // 返校时间转为整型 (月 * 100 + 日)
} Student;

int compareStudents(const void *a, const void *b) {
    Student *studentA = (Student *)a;
    Student *studentB = (Student *)b;

    // 按寝室升序
    int dormCompare = strcmp(studentA->dormitory, studentB->dormitory);
    if (dormCompare != 0) return dormCompare;
    return strcmp(studentA->name, studentB->name);
    // 比较离校时间升序
    return studentA->departure_day - studentB->departure_day;
}

int compareDeparture(const void *a, const void *b) {
    Student *studentA = (Student *)a;
    Student *studentB = (Student *)b;

    if (studentA->departure_day != studentB->departure_day) {
        return studentA->departure_day - studentB->departure_day;
    }
    return strcmp(studentA->name, studentB->name); // 若离校时间相同，按名字排序
}

int compareBack(const void *a, const void *b) {
    Student *studentA = (Student *)a;
    Student *studentB = (Student *)b;
```

```c
    if (studentA->back_day != studentB->back_day) {
        return studentA->back_day - studentB->back_day;
    }
    return strcmp(studentA->name, studentB->name); // 若返校时间相同，按名字排序
}

int main() {
    int T;
    scanf("%d", &T);
    for (int t = 1; t <= T; t++) {
        int N;
        scanf("%d", &N);

        Student students[MAX_STUDENTS];
        for (int i = 0; i < N; i++) {
            scanf("%s %s %s %s", students[i].name, students[i].dormitory,
                  students[i].departure, students[i].back);

            // 将离校时间和返校时间转换为整型
            int dep_month, dep_day;
            sscanf(students[i].departure, "%d.%d", &dep_month, &dep_day);
            students[i].departure_day = dep_month * 100 + dep_day;

            int back_month, back_day;
            sscanf(students[i].back, "%d.%d", &back_month, &back_day);
            students[i].back_day = back_month * 100 + back_day;
        }

        // 按寝室排序
        qsort(students, N, sizeof(Student), compareStudents);
        printf("Dormitory %d:\n", t);
        for (int i = 0; i < N; i++) {
            printf("%s %s %s %s\n", students[i].name, students[i].dormitory,
                   students[i].departure, students[i].back);
        }

        // 按离校时间排序
        qsort(students, N, sizeof(Student), compareDeparture);
        printf("Departure %d:\n", t);
        for (int i = 0; i < N; i++) {
            printf("%s %s %s %s\n", students[i].name, students[i].dormitory,
                   students[i].departure, students[i].back);
        }

        // 按返校时间排序
        qsort(students, N, sizeof(Student), compareBack);
        printf("Back %d:\n", t);
        for (int i = 0; i < N; i++) {
            printf("%s %s %s %s\n", students[i].name, students[i].dormitory,
                   students[i].departure, students[i].back);
        }
    }
    return 0;
}
```

2. 接近题解，运用了结构体指针

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

//日期存在10.10此类，用浮点数表示不利于统一
typedef struct {
    int day;
    int month;
} Date;

typedef struct {
    char name[21];
    char dorm[4];
    Date departure;
    Date back;
} Student;

// 按寝室
int compare_dorm(const void *a, const void *b) {
    Student *studentA = (Student *)a;
    Student *studentB = (Student *)b;

    int dorm_comparison = strcmp(studentA->dorm, studentB->dorm);
    if (dorm_comparison!= 0) return dorm_comparison;

    return strcmp(studentA->name, studentB->name);
}

// 按离校时间
int compare_departure(const void *a, const void *b) {
    Student *studentA = (Student *)a;
    Student *studentB = (Student *)b;

    if (studentA->departure.month!= studentB->departure.month)
        return studentA->departure.month - studentB->departure.month;
    if (studentA->departure.day!= studentB->departure.day)
        return studentA->departure.day - studentB->departure.day;

    return strcmp(studentA->name, studentB->name);
}

// 按返校时间
int compare_back(const void *a, const void *b) {
    Student *studentA = (Student *)a;
    Student *studentB = (Student *)b;

    if (studentA->back.month!= studentB->back.month)
        return studentA->back.month - studentB->back.month;
    if (studentA->back.day!= studentB->back.day)
        return studentA->back.day - studentB->back.day;

    return strcmp(studentA->name, studentB->name);
}
```

```
int main() {
    int T;
    scanf("%d", &T);

    for (int a = 0; a < T; a++) {
        int N;
        scanf("%d", &N);

        Student students[N];
        for (int i = 0; i < N; i++) {
            scanf("%s %s", students[i].name, students[i].dorm);
            scanf("%d.%d", &students[i].departure.month,
&students[i].departure.day);
            scanf("%d.%d", &students[i].back.month, &students[i].back.day);
        }

        qsort(students, N, sizeof(Student), compare_dorm);
        printf("Dormitory %d:\n", a + 1);
        for (int i = 0; i < N; i++) {
            printf("%s %s %d.%d %d.%d\n", students[i].name, students[i].dorm,
students[i].departure.month, students[i].departure.day, students[i].back.month,
students[i].back.day);
        }

        qsort(students, N, sizeof(Student), compare_departure);
        printf("Departure %d:\n", a + 1);
        for (int i = 0; i < N; i++) {
            printf("%s %s %d.%d %d.%d\n", students[i].name, students[i].dorm,
students[i].departure.month, students[i].departure.day, students[i].back.month,
students[i].back.day);
        }

        qsort(students, N, sizeof(Student), compare_back);
        printf("Back %d:\n", a + 1);
        for (int i = 0; i < N; i++) {
            printf("%s %s %d.%d %d.%d\n", students[i].name, students[i].dorm,
students[i].departure.month, students[i].departure.day, students[i].back.month,
students[i].back.day);
        }



    }

    return 0;
}
```

第七题

1. 这个同学的非常奇妙

```
#include<stdio.h>
#include<string.h>
int main(){
    int h,i,j,k,n,m;
    char a[16][20]={"Main\n","    ","data\n","    ","image\n","data\n","
","data1 101\n","    ","data2 20.1\n","    ","data3 1.23\n","image\n","
","ASC\n","->  "};
    for(h=i=0,m=1,j=k=5;scanf("%d",&n);){
        m=(m==13?1:m);
        for(;i<j;i++){
            printf("%s",a[i==m?15:i]);
        }
        i=h;
        if(n==0)break;
        if(n==1){
            m=(m-2<i?j-2:m-2);
        }
        h=i;
        if(n==2){
            m=(m+2>=j?i+1:m+2);
        }
        j=k;
        if(n==3&&m==1&&j==5){
            h=i=5;j=k=12;m=6;
        }
        if(n==3&&m==3&&j==5){
            h=i=12;j=k=15;m=13;
        }
        if(n==4&&j!=5){
            h=i=0;j=k=5;m=1;
        }
    }
    return 0;
}
```

2. 用结构体同学

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

typedef struct MenuItem{
    char *name;
    struct MenuItem *subMenu;
    int subMenusize;
}MenuItem;

MenuItem datamenu[] ={
    {"data1 101",NULL,0},
    {"data2 20.1",NULL,0},
    {"data3 1.23",NULL,0}
};
MenuItem imagemenu[] ={
    {"ASC",NULL,0}
};
MenuItem mainmenu[] ={
    {"data",datamenu,3},
    {"image",imagemenu,1}
};
```

```c
void printMenu(MenuItem *menu, int size, int cursor);

int cursorPosition = 0;
MenuItem *currentmenu = mainmenu;
int currentsize = 2;

int main(){
    int a;
    while(1){
        printMenu(currentmenu,currentsize,cursorPosition);
        scanf("%d",&a);
        switch(a){

            case 0:
                return 0;
            case 1:
                cursorPosition = (cursorPosition-1+currentsize)%currentsize;
                break;
            case 2:
                cursorPosition = (cursorPosition+1+currentsize)%currentsize;
                break;
            case 3:
                if(currentmenu[cursorPosition].subMenu != NULL){
                    currentsize = currentmenu[cursorPosition].subMenusize;
                    currentmenu = currentmenu[cursorPosition].subMenu;
                    cursorPosition = 0;
                }
                break;
            case 4:
                if(currentmenu != mainmenu){
                    currentmenu = mainmenu;
                    currentsize = 2;
                    cursorPosition = 0;
                }
                break;
        }

    }
}

void printMenu(MenuItem *menu, int size, int cursor) {
        if(currentmenu == mainmenu)
            printf("Main\n");
        if(currentmenu == datamenu)
            printf("data\n");
    if(currentmenu == imagemenu){
        printf("image\n");
        printf("    %s\n", menu[0].name);
    }
    else{
        for (int i = 0; i < size; i++) {
            if (i == cursor) {
                printf("->  %s\n", menu[i].name);
            } else {
                printf("    %s\n", menu[i].name);
            }
        }
    }
}
```

3. 代码规范较好

但是没有非法输入，也没有说非法输入要执行什么操作，自觉加入了。

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

// 定义菜单结构
typedef struct {
    int level;
    int cursor;
    int index1;
    int index2;
} MenuState;

// 打印菜单函数
void printMenu(const MenuState* state) {
    if (state->level == 0) {
        const char* items[] = { "  data", "  image" };
        printf("Main\n");
        for (int i = 0; i < 2; i++) {
            if (i == state->cursor) {
                printf("->%s\n", items[i]);
            }
            else {
                printf("  %s\n", items[i]);
            }
        }
    }
    else if (state->level == 1) {
        const char* items[] = { "  data1 101", "  data2 20.1", "  data3 1.23" };
        printf("data\n");
        for (int i = 0; i < 3; i++) {
            if (i == state->cursor) {
                printf("->%s\n", items[i]);
            }
            else {
                printf("  %s\n", items[i]);
            }
        }
    }
    else if (state->level == 2) {
        printf("image\n");
        printf("    ASC\n");
    }
}

int main() {
    // 初始化菜单状态
    MenuState state = { 0, 0, 0, 0 };
    int command;
    int commands[100]; // 假设最多100个命令
    int commandCount = 0;
    printMenu(&state);
    // 读取命令序列
    while (scanf("%d", &command) == 1 && command != 0 && commandCount < 100) {
        commands[commandCount++] = command;
    }

    // 处理每个命令
    for (int i = 0; i < commandCount; i++) {
        command = commands[i];

        switch (command) {
        case 1: // 向上移动
            if (state.level == 0) {
                state.cursor = (state.cursor + 1) % 2;
            }
            else {
                state.cursor = (state.cursor + 2) % 3;
            }
            break;
        case 2: // 向下移动
            if (state.level == 0) {
                state.cursor = (state.cursor + 1) % 2;
            }
            else {
                state.cursor = (state.cursor + 1) % 3;
            }
            break;
        case 3: // 确定
            if (state.level == 0) {
                if (state.cursor == 0) {
                    state.level = 1;
                    state.cursor = 0;
                }
                else if (state.cursor == 1) {
                    state.level = 2;
                    state.cursor = 0;
                }
            }
            break;
        case 4: // 返回
            if (state.level > 0) {
                state.level = 0;
                state.cursor = state.index1;
            }
            break;
        case 0: // 结束
            break;
        default:
            printf("Unknown command: %d\n", command);
            break;
        }
        printMenu(&state);
    }

    return 0;
}
```