

步骤 1: 显示并选择主菜单

此功能主要是向用户展示系统的操作选项,包括“Input”(录入信息)、“Output”(输出信息)、“Order”(排序)和“Quit”(退出)等。通过在控制台输出固定格式的菜单内容,让用户清楚了解系统可执行的操作。

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void inputAndSortStudentInfo(struct Student students[], int count) {
4      inputStudentInfo(students, count);
5      for (int i = 0; i < count - 1; i++) {
6          for (int j = 0; j < count - i - 1; j++) {
7              if (students[j].score > students[j + 1].score) {
8                  struct Student temp = students[j];
9                  students[j] = students[j + 1];
10                 students[j + 1] = temp;
11             }
12         }
13     }
14
15     displayStudentInfo(students, count);
16 }
```

步骤 2：录入并显示学生信息

该步骤允许用户输入学生的详细信息，包括学号（要求为 5 位自然数且高位不为 0）、班级、姓名以及高数、大学物理和英语成绩（成绩要求为 2 位整数和 1 位小数）。在代码中，首先定义了 Student 结构体来存储学生信息，结构体成员包括指向字符串的指针 id（学号）、clas（班级）、name（姓名）以及三个成绩成员 score1、score2、score3 和总成绩 score。在录入信息时，为每个学生的学号、班级和姓名动态分配内存空间，然后使用 scanf 函数依次获取用户输入的各项信息，并计算总成绩。

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 void deleteStudentInfo(struct Student students[], int *count, char *idOrName) {
5     int i;
6     for (i = 0; i < *count; i++) {
7         if (strcmp(students[i].id, idOrName) == 0 || strcmp(students[i].name, idOrName) == 0) {
8             break;
9         }
10    }
11
12    if (i == *count) {
13        // 未找到学生
14        printf("未找到该学生信息\n");
15        return;
16    }
17    for (int j = i; j < *count - 1; j++) {
18        students[j] = students[j + 1];
19    }
20    (*count)--;
21 }
```

步骤 3: 显示选择主菜单 + 录入并显示学生信息 (包含排序)

首先再次向用户展示主菜单, 提供操作选项。然后获取用户输入并根据输入进行相应处理, 当用户输入 “i” 时, 进入下一步骤的信息录入和排序流程; 输入 “m” 或 “q” 时, 输出相应提示信息; 输入其他字符时, 提示错误。

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 struct Student {
5     char *id;
6     char *clas;
7     char *name;
8     double score1;
9     double score2;
10    double score3;
11    double score;
12 };
13 void modifyStudentInfo(struct Student students[], int count, struct Student modifiedStudent) {
14     int i;
15     for (i = 0; i < count; i++) {
16         if (strcmp(students[i].id, modifiedStudent.id) == 0) {
17             break;
18         }
19     }
20
21     if (i == count) {
22         printf("未找到该学生信息\n");
23         return;
24     }
25     students[i] = modifiedStudent;
26 }
27 void displayModifiedStudentInfo(struct Student students[], int count) {
28     for (int i = 0; i < count; i++) {
29         if (i > 0 && strcmp(students[i].clas, students[i - 1].clas) == 0) {
30             printf(" ");
31         } else {
32             printf("%s ", students[i].clas);
33         }
34         printf("%s %s %.11f %.11f %.11f", students[i].id, students[i].name, students[i].score1, students[i].score2,
35             students[i].score3);
36         if (strcmp(students[i].id, modifiedStudent.id) == 0) {
37             printf(" modified\n");
38         } else {
39             printf("\n");
40         }
41     }
42 }
```

步骤 4: 插入并显示学生信息

假设系统中已有部分学生信息，此功能允许用户插入新的学生信息。新插入的学生信息需要插入到合适的位置，以保持学生信息数组按照班级从小到大、同一班级内总成绩从大到小的顺序排列。在代码中，首先通过循环找到新学生信息应插入的位置，比较新学生的班级与已有学生的班级，如果班级较小或者班级相同但总成绩较大，则找到插入位置。然后，将插入位置后的学生信息依次向后移动一位，为新学生腾出空间，最后将新学生信息插入到合适位置，并更新学生信息数组的元素个数。

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  void insertStudentInfo(struct Student students[], int *count, struct Student newStudent) {
5      int i;
6      for (i = 0; i < *count; i++) {
7          if (strcmp(newStudent.clas, students[i].clas) < 0 ||
8              (strcmp(newStudent.clas, students[i].clas) == 0 && newStudent.score > students[i].score)) {
9              break;
10         }
11     }
12     for (int j = *count; j > i; j--) {
13         students[j] = students[j - 1];
14     }
15     students[i] = newStudent;
16     (*count)++;
17 }
```

步骤 5: 删除并显示学生信息

此功能用于删除系统中已有的学生信息。用户输入待删除学生的学号或姓名后，系统在学生信息数组中查找该学生。如果找到，将该学生信息从数组中删除，并保持数组的有序性；如果未找到，则输出提示信息“未找到该学生信息”。在代码中，通过循环遍历数组，比较输入的学号或姓名与每个学生的对应信息，找到后将其后的学生信息依次向前移动一位，覆盖要删除的学生信息，并将数组元素个数减 1。

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct Student {
4      char *id;
5      char *clas;
6      char *name;
7      double score1;
8      double score2;
9      double score3;
10     double score;
11 };
12 void inputStudentInfo(struct Student students[], int count) {
13     for (int i = 0; i < count; i++) {
14         students[i].id = (char *)malloc(sizeof(char) * 10);
15         students[i].clas = (char *)malloc(sizeof(char) * 10);
16         students[i].name = (char *)malloc(sizeof(char) * 10);
17
18         printf("请输入学号 (5位自然数, 高位不为0) : ");
19         scanf("%s", students[i].id);
20         printf("请输入班级: ");
21         scanf("%s", students[i].clas);
22         printf("请输入姓名: ");
23         scanf("%s", students[i].name);
24         printf("请输入高数成绩 (2位整数, 1位小数) : ");
25         scanf("%lf", &students[i].score1);
26         printf("请输入大学物理成绩 (2位整数, 1位小数) : ");
27         scanf("%lf", &students[i].score2);
28         printf("请输入英语成绩 (2位整数, 1位小数) : ");
29         scanf("%lf", &students[i].score3);
30
31         students[i].score = students[i].score1 + students[i].score2 + students[i].score3;
32     }
33 }
34 void displayStudentInfo(struct Student students[], int count) {
35     for (int i = 0; i < count; i++) {
36         printf("%s %s %s %.1lf %.1lf %.1lf %.1lf\n", students[i].id, students[i].clas, students[i].name,
37             students[i].score1, students[i].score2, students[i].score3, students[i].score);
38     }
39 }
```

步骤 6: 修改并显示学生信息

当用户需要修改某个学生的信息时,使用此功能。用户输入要修改的学生的全部信息(包括学号、班级、姓名和三门课程成绩),系统通过学号在学生信息数组中查找该学生。

如果找到,将其信息替换为新输入的信息;如果未找到,则输出提示信息“未找到该学生信息”。在代码中,通过循环比较输入的学号与数组中每个学生的学号,找到匹配的学生后进行信息更新。

```
1  #include <stdio.h>
2  void displayMenu() {
3      printf("          1.Input\n");
4      printf("          2.Output\n");
5      printf("          3.Order\n");
6      printf("          4.Quit\n");
7  }
8  void handleMenuChoice() {
9      char choice;
10     scanf(" %c", &choice);
11     switch (choice){
12         case 'i':
13             printf("You are trying to Input info\n");
14             break;
15         case 'o':
16             printf("You are trying to Output info\n");
17             break;
18         case 'm':
19             printf("You are trying to Make things ordered\n");
20             break;
21         case 'q':
22             printf("You are about to Quit\n");
23             break;
24         default:
25             printf("Wrong input\n");
26     }
27 }
```

总结

学生成绩管理系统通过六个步骤实现了全面的功能，包括显示主菜单供用户选择操作，如录入、输出、排序、插入、删除、修改学生信息等；录入时对学号、成绩等格式有要求并计算总成绩，可按需排序；插入时新信息按规则有序插入，删除依据学号或姓名准确执行且保持数组有序，修改能更新指定学生信息，各步骤紧密配合，在数据处理过程中注重与用户交互，提供清晰反馈，确保学生信息管理的高效性、准确性与便捷性。