

北京交通大学

《数据结构（A）》第2章设计型作业

专 业： 计算机科学与技术

班 级：

学生姓名：

学 号：

北京交通大学计算机与信息技术学院

2021 年 10 月 06 日

第 2 章设计型作业^①

1 设计型题目

2.5 某软件公司大约有 30 名员工，每名员工有姓名、工号、职务等属性，每年都有员工离职和入职。把所有员工建立一个线性表，建立离职和入职函数，当有员工离职或入职时，修改线性表，并且显示最新的员工名单。具体要求：

(1) 顺序表存储：实现顺序表的插入、删除、查找、输出等基本操作；调用基本操作完成。

(2) 链表存储：实现链表的插入、删除、查找、输出等基本操作；调用基本操作完成。

2 设计型题目解答

【第 2.5 题解答】

(1) 思路：

正常建立线性表，声明 `struct staff` 和 `struct company`，然后结构体嵌套，`company` 中就可以有其他如 `count` 一类的成员。

代码：

```
01://
02:// Created by Planck Chang on 2021/10/4.
03://
04:#include <stdio.h>
05:#include <stdlib.h>
06:#include <string.h>
07:
08:struct staff{
09:    char name[20];
10:    int id;
```

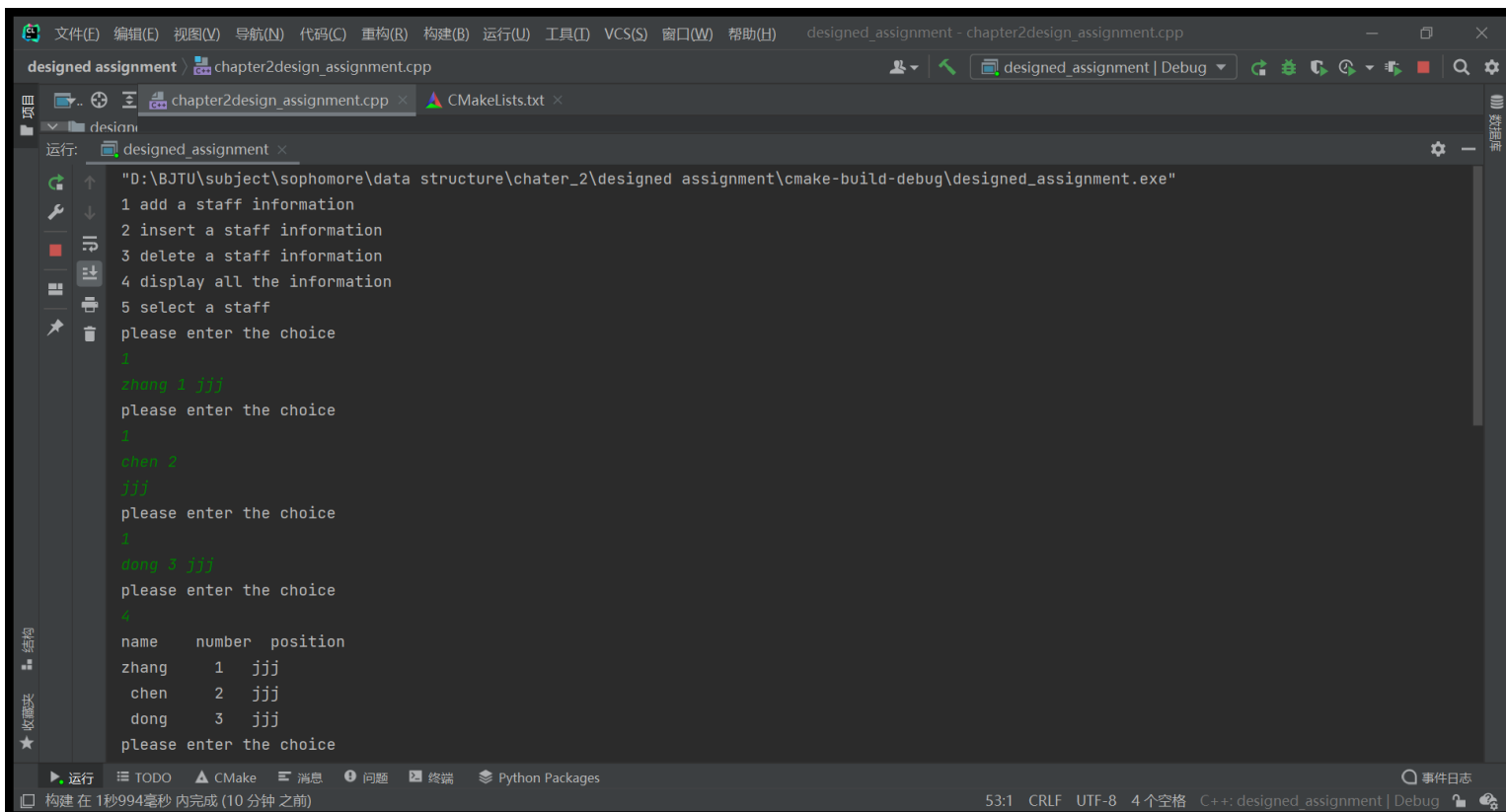
^① 《数据结构 (A)》第 2 章设计型作业只有这一个题目，但其中有多个问题。要求学生最晚提交日期是 2021 年 10 月 9 日。

```
11:  char position[20];
12:};
13:
14:struct company{
15:  staff *member;
16:  int count;
17:};
18:
19:
20:void init_company(company* company){
21:  company->member = (staff*) malloc(35* sizeof(staff));
22:  company->count=-1;
23:}
24:
25:
26:void insert_staff(company *l, int location, staff *e){
27:  if( location<0||location>l->count){
28:    printf("the position inserted error\n");
29:  }
30:  else{
31:    for(int i=l->count; i>=location; i--){
32:      l->member[i+1] = l->member[i];
33:    }
34:    l->member[location] = *e;
35:    l->count++;
36:  }
37:}
38:
39:void locate_elem(company *l, int num){
40:  for(int i=0; i<=l->count; i++){
41:    if(l->member[i].id == num){
42:      printf("%s %d %s\n", l->member[i].name , l->member[i].id,
        l->member[i].position);
43:    }
44:  }
45:}
46:
47:void delete_elem(company*l, int num){
48:  for(int i= 0; i<=l->count; i++){
49:    if(l->member[i].id == num){
50:      for(int j=i; j<l->count; j++){
51:        l->member[j] = l->member[j+1];
52:      }
```

```
53:         l->count--;
54:         break;
55:     }
56: }
57:}
58:
59:void display(company *l){
60:     printf("name\tnumber\tposition\n");
61:     for (int i = 0; i <= l->count; i++)
62:         printf("%5s %5d %5s\n", l->member[i].name, l->member[i].id,
        l->member[i].position);
63:}
64:
65:void add_elem(company *l, staff *e) {
66:     l->member[l->count+1] = *e;
67:     l->count++;
68:}
69:
70:int main(){
71:     int choice, location;
72:     staff *tem = (staff*)malloc(sizeof(staff));
73:     company *l = (company*) malloc(sizeof(company));
74:
75:     init_company(l);
76:
77:     printf("1 add a staff information\n");
78:     printf("2 insert a staff information\n");
79:     printf("3 delete a staff information\n");
80:     printf("4 display all the information\n");
81:     printf("5 select a staff\n");
82:     printf("please enter the choice\n");
83:     scanf("%d", &choice);
84:     while(choice!=0){
85:         switch (choice) {
86:             case 1:
87:                 scanf("%s %d %s",&tem->name ,&tem->id, &tem->position);
88:                 add_elem(l, tem);
89:                 break;
90:             case 2:
91:                 scanf("%s %d %s",&tem->name ,&tem->id, &tem->position);
92:                 scanf("%d", &location);
93:                 insert_staff(l, location, tem);
94:                 break;
```

```
95:         case 3:
96:             scanf("%d", &tem->id);
97:             delete_elem(1, tem->id);
98:             break;
99:         case 4:
100:            display(1);
101:            break;
102:        case 5:
103:            scanf("%d",&location);
104:            locate_elem(1, location);
105:            break;
106:    }
107:    printf("please enter the choice\n");
108:
109:    scanf("%d", &choice);
110:}
111:}
```

调试结果:



The screenshot shows a C++ IDE with the following content:

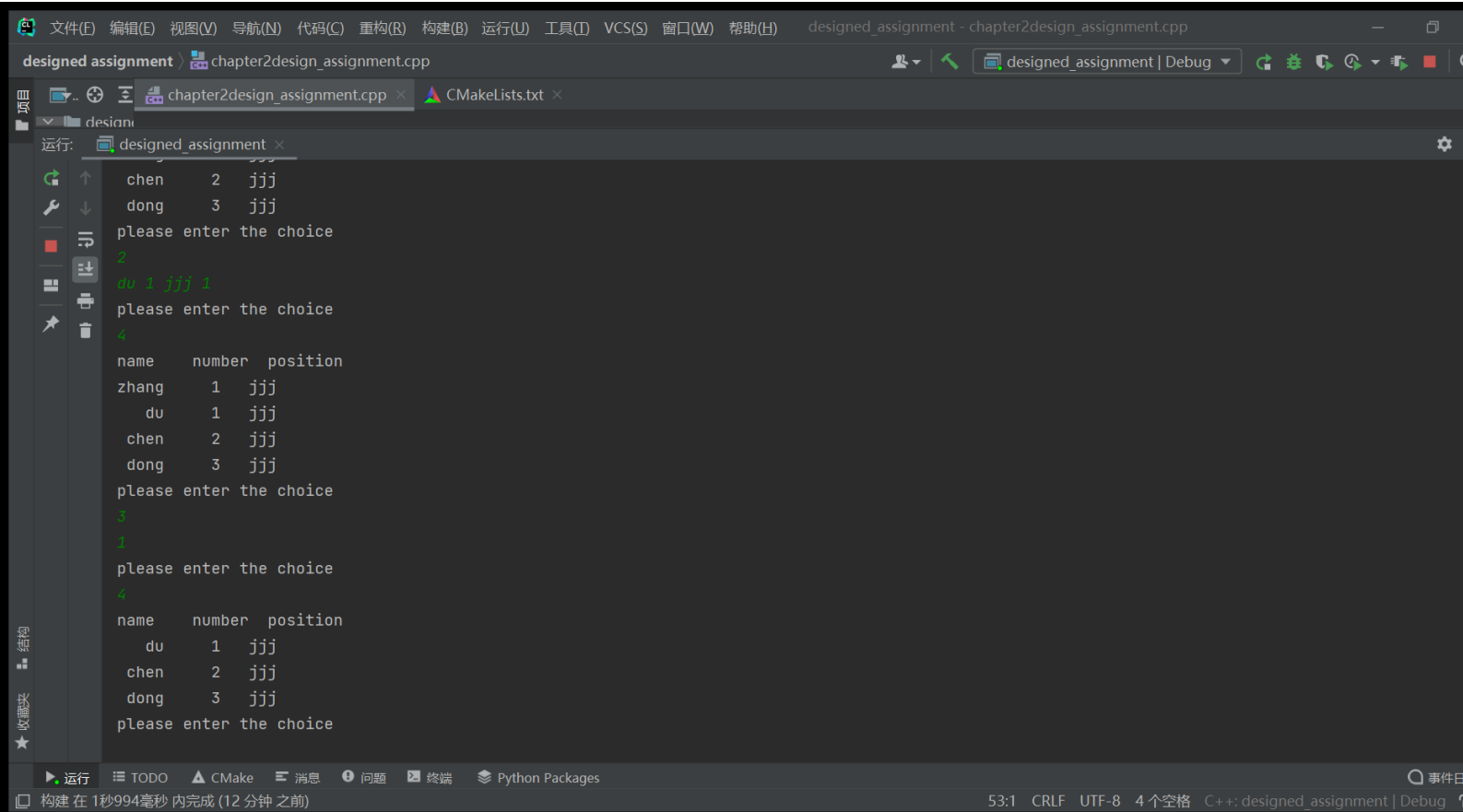
File Explorer: `chapter2design_assignment.cpp`, `CMakeLists.txt`

Run Output: `designed_assignment.exe`

Terminal Output:

```
"D:\BJTU\subject\sophomore\data structure\chater_2\designed assignment\cmake-build-debug\designed_assignment.exe"
1 add a staff information
2 insert a staff information
3 delete a staff information
4 display all the information
5 select a staff
please enter the choice
1
zhang 1 jji
please enter the choice
1
chen 2
jji
please enter the choice
1
dong 3 jji
please enter the choice
4
name    number position
zhang   1    jji
chen    2    jji
dong    3    jji
please enter the choice
```

Bottom Bar: 53:1 CRLF UTF-8 4 个空格 C++: designed_assignment | Debug



```
designed assignment - chapter2design_assignment.cpp
chapter2design_assignment.cpp x CMakeLists.txt x
desian
designed_assignment x
运行:
chen 2 jjj
dong 3 jjj
please enter the choice
1
du 1 jjj
please enter the choice
4
name number position
zhang 1 jjj
du 1 jjj
chen 2 jjj
dong 3 jjj
please enter the choice
3
1
please enter the choice
4
name number position
du 1 jjj
chen 2 jjj
dong 3 jjj
please enter the choice
```

(2) 思路:

正常建立单链表。注意，不要多个指针指向同一个临时标量的地址，要多解引用赋值到不会被覆盖的地址。

代码:

```
01:#include <iostream>
02:#include <string.h>
03:#include <stdlib.h>
04:
05:struct staff{
06:    char name[10];
07:    int id;
08:    char position[10];
09:    staff *next;
10:};
11:
12:staff *init_link(){
13:    staff *head= (staff*) malloc(sizeof(staff));
14:    head->next=NULL;
15:    return head;
```

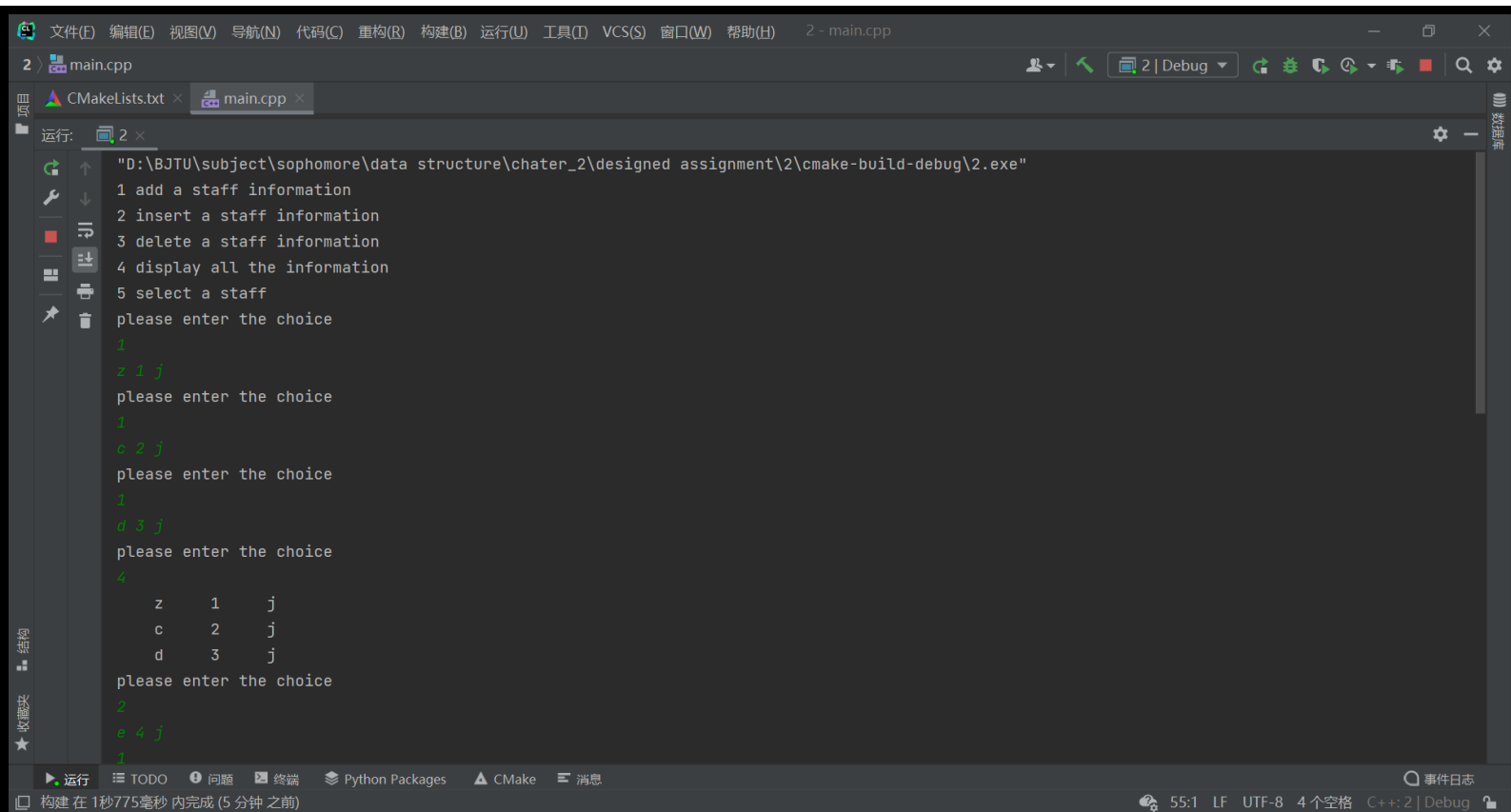
```
16:}
17:
18:void add_node(staff*l, staff add){
19:    staff*p=l;
20:    staff*q=(staff*) malloc(sizeof(staff));
21:    while(p->next!=NULL)
22:        p= p->next;
23:    *q=add;
24:    p->next=q;
25:}
26:
27:void select_node(staff*l, int id){
28:    staff*p= l;
29:    p=p->next;
30:    while (p->id!=id){
31:        p= p->next;
32:    }
33:    printf("%5s %5d %5s\n", p->name, p->id, p->position );
34:}
35:
36:void insert_node(staff*l, int location, staff new_node){
37:    staff*p=l->next;
38:    staff*q =(staff*) malloc(sizeof(staff));
39:    *q = new_node;
40:
41:    for(int i=0; i<location-1; i++)
42:        p=p->next;
43:    q->next=p->next;
44:    p->next=q;
45:}
46:
47:void display_node(staff*l){
48:    staff*p=l;
49:    p=p->next;
50:    while(p->next!=NULL){
51:        printf("%5s %5d %5s\n", p->name, p->id, p->position );
52:        p=p->next;
53:    }
54:    printf("%5s %5d %5s\n", p->name, p->id, p->position );
55:
56:}
57:
58:void delete_node(staff*l, int id){
```



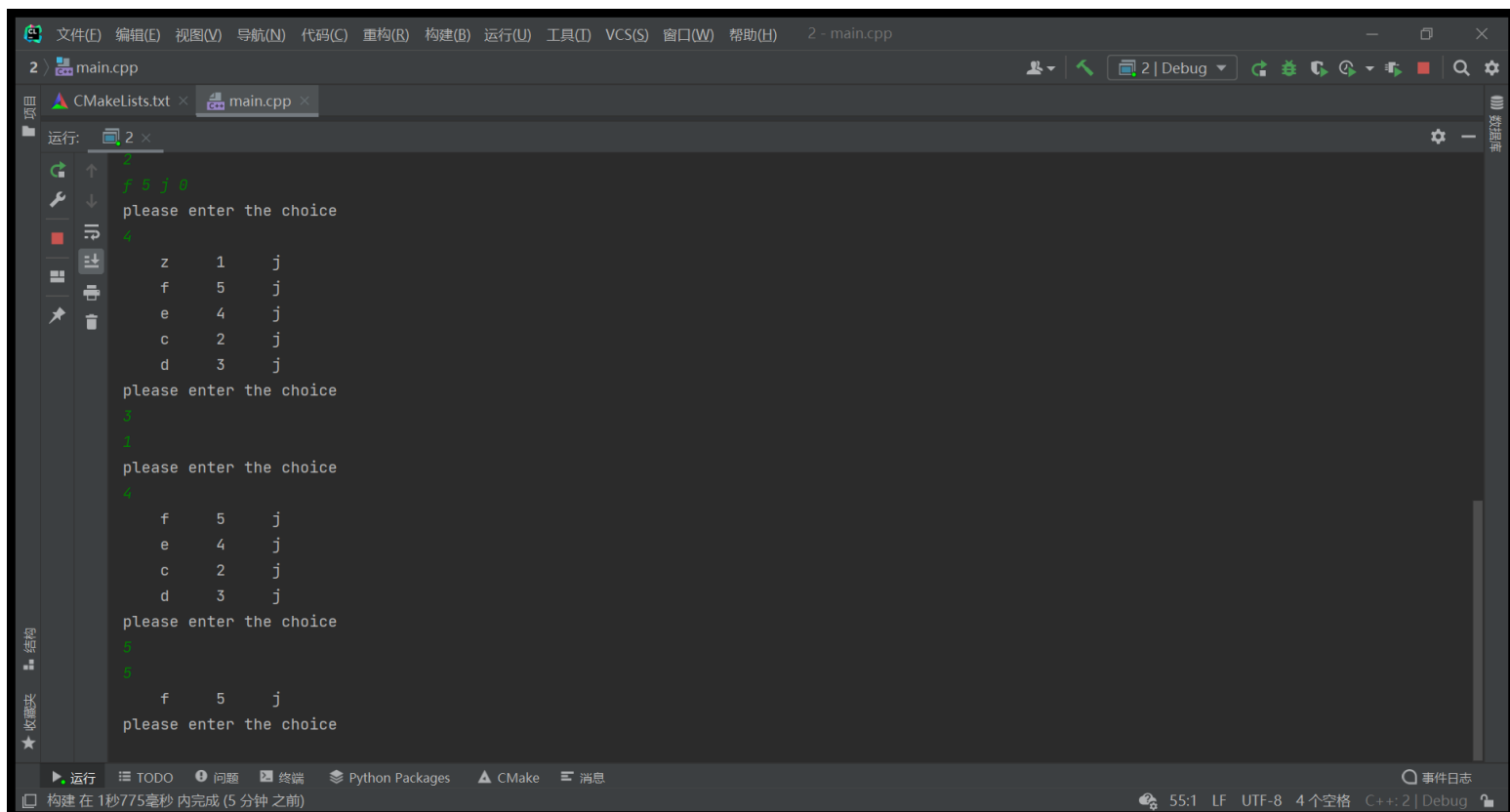
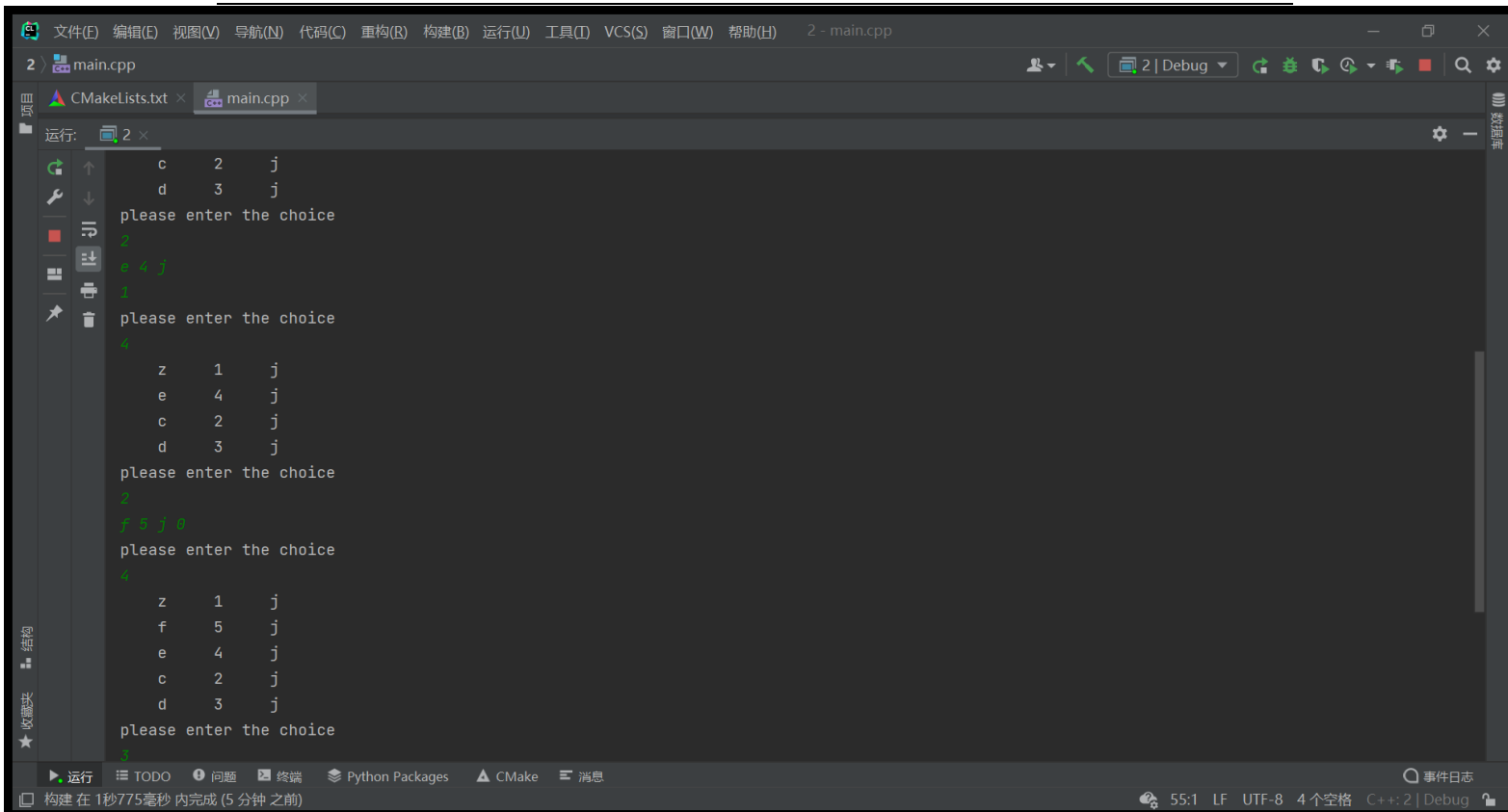
```
59:   staff*p= l->next;
60:   staff*q=l;
61:   while(p->next!=NULL){
62:       if(p->id==id){
63:           q->next=p->next;
64:           free(p);
65:           break;
66:       }
67:       else{
68:           q=q->next;
69:           p=p->next;
70:       }
71:   }
72:}
73:
74:
75:
76:
77:
78:int main(){
79:   int choice, location;
80:   staff tem;
81:   tem.next=NULL;
82:   staff*l= init_link();
83:
84:   printf("1 add a staff information\n");
85:   printf("2 insert a staff information\n");
86:   printf("3 delete a staff information\n");
87:   printf("4 display all the information\n");
88:   printf("5 select a staff\n");
89:   printf("please enter the choice\n");
90:   scanf("%d", &choice);
91:   while(choice!=0){
92:       switch (choice) {
93:           case 1:
94:               scanf("%s %d %s",&tem.name ,&tem.id, &tem.position);
95:               add_node(l, tem);
96:               break;
97:           case 2:
98:               scanf("%s %d %s",&tem.name ,&tem.id, &tem.position);
99:               scanf("%d", &location);
100:              insert_node(l, location, tem);
101:              break;
```

```
102:         case 3:
103:             scanf("%d", &tem.id);
104:             delete_node(1, tem.id);
105:             break;
106:         case 4:
107:             display_node(1);
108:             break;
109:         case 5:
110:             scanf("%d",&location);
111:             select_node(1, location);
112:             break;
113:     }
114:     printf("please enter the choice\n");
115:
116:     scanf("%d", &choice);
117: }
118: }
```

运行情况:



```
文件(F) 编辑(E) 视图(V) 导航(N) 代码(C) 重构(R) 构建(B) 运行(U) 工具(T) VCS(S) 窗口(W) 帮助(H) 2 - main.cpp
2 main.cpp
CMakeLists.txt main.cpp
运行: 2 x
"D:\BJTU\subject\sophomore\data structure\chater_2\designed assignment\2\cmake-build-debug\2.exe"
1 add a staff information
2 insert a staff information
3 delete a staff information
4 display all the information
5 select a staff
please enter the choice
1
2 1 j
please enter the choice
2
3 2 j
please enter the choice
3
4 3 j
please enter the choice
4
z 1 j
c 2 j
d 3 j
please enter the choice
5
z 1 j
c 2 j
d 3 j
please enter the choice
2
3 4 j
1
构建在 1秒775毫秒 内完成 (5 分钟 之前)
```



运行正常。