```
                         /*函数及变量声明部分*/
#include<iostream>
#include<stdlib.h>
#include<fstream>
#include<wchar.h>
#include<string.h>

/*VS 编译器环境*/
using namespace std;
#define _NUMBER 10//定义结构体成员的最大长度
#define BUFFER_MAX 520//最大缓冲区长度
#define BUFFER_MIN 40//最小缓冲区长度

void Create_information_list(List &);//创建员工信息表
void Insert_of_list(List & );//员工信息新增
void information_output(const List L);//员工信息输出
int inferation_List_length(List &);//统计多少条记录
void serach_by_ID(const List &);//以 ID 为关键字进行搜索
void serach_by_name(List &);//以姓名为关键字进行搜索

//void Clear_ID_data(List &, char[]);//以 ID 为关键字清空该人的信息    该函数作为保留功能
void Clear_all_data(List &);//清空表中所有信息
void Destroy_People_of_ID(List &);//以 ID 为关键字删除该人的信息
void Destroy_List_People(List    *);
void change_data_ID(List &);//以 ID 为关键字更改该人的信息
void selection_sort_by_ID(List &);//以 ID 为进行选择排序信息表
void sieve_method_by_sex(List &);//以性别为关键字进行筛选
void sieve_method_by_duty(List &);//以职务为关键字进行筛选
int Menu();//菜单
void Output_of_file(List &);//输出到文件
void Input_of_file(List &);//从文件读取到内存


typedef struct List
{
      long ID;//编号
      char name[_NUMBER];//姓名
      char sex[_NUMBER];//性别
      int age;//年龄
      int data;//保留操作
      char duty[_NUMBER];//职务
      char title[_NUMBER];//职称
      char politics_status[_NUMBER];//政治面貌
```

```
    char academic[_NUMBER];//最高学历
    char category[_NUMBER];//人员类别
    struct List *pnext;//指针域
}List;




  /*                          函数定义部分                              */

void Create_information_list(List &head)//初始化员工信息管理表
{
    List *temp;
    temp = &head;
    printf("\t***********请输入表头结点员工的信息***********\n");
    printf("请输入第一个员工的编号、姓名、性别(以换行/空格/Tab 分隔）\n");
    std::cin >> temp->ID >> temp->name >> temp->sex;
    //std::cout<<L->ID<<"    "<<L->name<<"    "<<L->sex<<std::endl;
    printf("请输入第一个员工的年龄、职务、职称(以换行/空格/Tab 分隔）\n");
    std::cin >> temp->age >> temp->duty >> temp->title;
    printf("请输入第一个员工的政治面貌、最高学历、人员类别(以换行/空格/Tab 分隔）\n");
    std::cin >> temp->politics_status >> temp->academic >> temp->category;
    temp->pnext = NULL;
    printf("第一个员工数据输入完成\n");
}

void Insert_of_list(List &L)//新增员工信息
{
    int n;
    printf("请输入要新增员工记录的个数\n");
    cin >> n;
    List *ppm = &L;
    List *temp;
    for (int i = 1; i <= n; i++)
    {
        temp = new List;//循环申请节点
        printf("请输入第%d 个员工的编号、姓名、性别(以换行/空格/Tab 分隔）\n", i);
        std::cin >> temp->ID >> temp->name >> temp->sex;
        //std::cout<<L->ID<<"    "<<L->name<<"    "<<L->sex<<std::endl;
        printf("请输入第%d 个员工的年龄、职务、职称(以换行/空格/Tab 分隔）\n", i);
        std::cin >> temp->age >> temp->duty >> temp->title;
        printf("请输入第%d 个员工的政治面貌、最高学历、人员类别(以换行/空格/Tab 分
隔）\n", i);
        std::cin >> temp->politics_status >> temp->academic >> temp->category;
        printf("第%d 个员工数据输入完成\n", i);
        ppm->pnext = temp;
        ppm = temp;
    }
    ppm->pnext = NULL;
}

void information_output(const List L)
{
    const List *temp = &L;
    /*                  输出员工数据                  */
    printf("人员编号   姓名   性别   年龄   职务   职称   政治面貌   最高学历   人员类型
\n ");
    for (temp; temp; temp = temp->pnext)
    {
        std::cout << temp->ID << "    " << temp->name << "    " << temp->sex;
        std::cout << "    " << temp->age << "    " << temp->duty << "    " << temp->title;
        std::cout << "    " << temp->politics_status << "    " << temp->academic << "    " <<
temp->category << std::endl;
    }
    temp = &L;//重回
}
int inferation_List_length(List &L)//统计个数
{
    List *temp;
    temp = &L;
    int count = 0;//计数器
    for(temp;temp;temp=temp->pnext)
    {
        count++;
        if (temp->pnext == NULL)//如果 pnext 域为空，则退出循环
        {
            break;
        }
    }
    return count;
}

void serach_by_ID(const List & L )//以 ID 为关键字进行搜索
{
    long number;
    printf("请输入要搜索员工的 ID 编号\n");
    cin >> number;
    const List *ptemp = &L;//初始化临时指针
    for (; ptemp;)//计数器
    {
        if (ptemp->ID == number)
        {   /*          输出信息                */
            printf("find the  number of %ld.\n", number);
            std::cout << ptemp->ID << "  " << ptemp->name << "  " << ptemp->sex;
            std::cout << "  " << ptemp->age << "  " << ptemp->duty << "  " << ptemp->title;
            std::cout << "  " << ptemp->politics_status << "  " << ptemp->academic << "  "
<< ptemp->category << std::endl;
        }
        else
        {
            printf("Not find the data number of %ld.\n", number);
        }
        ptemp = ptemp->pnext;//指针移动
    }
}

void serach_by_name(List & L)//以姓名为关键字进行搜索
{
    char str[100];
    printf("请输入要搜索的员工的姓名\n");
    cin >> str;
    List *ptemp = &L;//初始化临时指针
    for (; ptemp;)//计数器
    {

        if (!strcmp(ptemp->name, str))
        {
            printf("The people named  %s was found in the list\n", ptemp->name);
            std::cout << ptemp->ID << "  " << ptemp->name << "  " << ptemp->sex;
            std::cout << "  " << ptemp->age << "  " << ptemp->duty << "  " << ptemp->title;
            std::cout << "  " << ptemp->politics_status << "  " << ptemp->academic << "  "
<< ptemp->category << std::endl;
        }
        else
        {
            printf("The people named  %s was not found in the list\n", ptemp->name);
        }
        ptemp = ptemp->pnext;//指针移动
    }
}
```

```
void Clear_all_data(List &L)//引用类型
{
    List *temp;
    temp = &L;
    while (temp)
    {   /*     数据请空操作       */
        temp->ID = 0;
        temp->name[0] = '\0';
        temp->age = 0;
        temp->sex[0] = '\0';
        temp->duty[0] = '\0';
        temp->title[0] = '\0';
        temp->politics_status[0] = '\0';
        temp->academic[0] = '\0';
        temp->category[0] = '\0';
        temp = temp->pnext;//指针移动
    }
}
void Destroy_List_People(List   *L)
{
    printf("销毁线性表");
    for (L; L; L = L->pnext)
    {
        delete L;//释放链表  C++风格
    }
    printf("成功\n");
}

void Destroy_People_of_ID(List &L )
{
    long delete_number;
    printf("请输入要进行删除记录员工的员工 ID\n");
    cin >> delete_number;
    List *ptemp = &L;//初始化临时指针
    List *temp;
    List *bbq;
    for (; ptemp; ptemp = ptemp->pnext)
    {

        if (ptemp->ID == delete_number)
        {
            temp = ptemp->pnext;//结点移动
            ptemp->pnext = temp;
            delete temp;//删除借点
            printf("delete the ID of %ld's information \n", delete_number);
            break;
        }
        else
        {
            printf(" Not delete the ID of %ld \n", delete_number);
        }
    }
}

void change_data_ID(List &L)
{
    long id;
    printf("请输入要进行删除记录员工的员工 ID\n");
    cin >> id;
    List *ptemp = &L;//初始化临时指针
    for (; ptemp; ptemp = ptemp->pnext)//计数器
    {
        if (ptemp->ID == id)
        {
            printf("The ID of %s was found,you can change the informations of %s \n", ptemp->name, ptemp->name);
            printf("请输入 ID 为%ld 的员工的编号、姓名、性别\n", id);
            std::cin >> ptemp->ID >> ptemp->name >> ptemp->sex;
            //std::cout<<L->ID<<"    "<<L->name<<"    "<<L->sex<<std::endl;
            printf("请输入 ID 为%ld 个员工的年龄、职务、职称\n", id);
            std::cin >> ptemp->age >> ptemp->duty >> ptemp->title;
            printf("请输入 ID 为%ld 个员工的政治面貌、最高学历、人员类别\n", id);
            std::cin >> ptemp->politics_status >> ptemp->academic >> ptemp->category;
            printf("ID 为%ld 的员工的数据更改完成\n", id);
            break;//跳出循环
        }
        else
        {
            printf(" Not delete the ID of %ld \n", id);
        }
    }
}
void selection_sort_by_ID(List &L)
{
    List *temp=&L;
    List *min_data = &L;
    int    temp_of_data;
```

```
	while (temp)
	{
		if (min_data->ID >= temp->ID)
		{
			temp_of_data = min_data->ID;
			min_data->ID = temp->ID;
			temp->ID = temp_of_data;
		}
		temp = temp->pnext;
	}
	cout << "以 ID 为关键字排序完成" << endl;
	//delete temp, min_data;
}

void sieve_method_by_sex(List &L )//以性别为关键字进行筛选
{
	char sex_array[100];
	printf("请以性别关键字进行筛选\n");
	cin >> sex_array;
	List *temp = &L;
	for (temp; temp; temp = temp->pnext)
	{
		if (!strcmp(temp->sex, sex_array))
		{
			printf("The	name of %s	is %s\n", temp->name, temp->sex);
		}
		else
			printf("未找到\n");
	}
}

void sieve_method_by_duty(List &L)//以职务为关键字进行筛选
{
	char duty_array[100];
	printf("请以职务关键字进行筛选\n");
	cin >> duty_array;
	List *temp = &L;
	L.pnext = NULL;
	for (temp; temp; temp = temp->pnext)
	{
		if (!strcmp(temp->sex, duty_array))
		{
			printf("The	name of %s	is %s\n", temp->name, temp->duty);
		}
		else
			printf("未找到\n");
	}
}
int Menu()
{
	int end_number;//菜单标识
	printf("\n                    DIY 员工信息管理                    \n");
	printf("
************************************************************************\n");
	printf("            *                 1. 创建员工信息管理表
*\n");
	printf("            *                 2. 新增员工信息
*\n");
	printf("            *                 3. 员工信息输出
*\n");
	printf("            *                 4. 统计员工信息条数
*\n");
	printf("            *                 5. 按 ID 搜索
*\n");
	printf("            *                 6. 按姓名搜索
*\n");
	printf("            *                 7. 清空表中员工信息
*\n");
	printf("            *                 8. 按 ID 删除
*\n");
	printf("            *                 9. 按 ID 更改
*\n");
	printf("            *                 10. 按 ID 排序
*\n");
	printf("            *                 11. 按姓名筛选
*\n");
	printf("            *                 12. 按职务筛选
*\n");
	printf("            *                 13. 保存员工信息
*\n");
	printf("            *                 14. 读取员工信息
*\n");
	printf("            *                 0. 退出
*\n");
	printf("
************************************************************************\n");
	printf("请输入你的选择(0-14):");
	scanf("%d", &end_number);//读入用户输入
```

```
    return    end_number;
}

void Output_of_file(List & head)//输出到文件
{
    List *use_head = &head;//操作链表指针
    ofstream fout;
   char filename[40]="c:\\information_manger_system.txt";//文件名
   printf("请输入要存取的文件路径以及文件名(要使用转义字符)\n");
   cin >> filename;
       fout.open(filename, ios::out|ios::in|ios::app);//打开文件及其模式
        /*    ios::out 是写模式，ios::in 是读模式    */
       fout << "人员编号（ID）" << " " << "姓名" << " " << "性别" << " "\
       << "年龄" << " " << "职务" << " " << "职称" << " "\
       << "政治面貌" << " " << "最高学历" << " " << "人员类别" << endl;
   if (fout.bad())
   {
       printf("文件不存在或打开失败\n");
       fout << "Writing to a basic_ofstream object..." << endl;
       fout.close();
       abort();//退出函数
   }

   else
       for (use_head; use_head; use_head = use_head->pnext)
       {         /*          输出链表中数据的信息          */
           fout << use_head->ID << " " << use_head->name << " " << use_head->sex << " "\
               << use_head->age << " " << use_head->duty << " " << use_head->title << " "\
               << use_head->politics_status << " " << use_head->academic << " "\
               << use_head->category << endl;
       }
   fout.close();//关闭文件
}
void Input_of_file(List &head)//从文件读取到内存
{
    char str[BUFFER_MAX];//按行读取最大缓冲区
    List *use_head = &head;//操作链表指针
    ifstream fin;//文件读指针
    char filename[BUFFER_MIN] = "c:\\information_manger_system.txt";//文件名
    printf("请输入要读取的文件路径以及文件名(要使用转义字符)\n");
    cin >> filename;
    fin.open(filename, ios::out | ios::in);//读取文件及其模式
    int count = 0;
    /*while (fin>>str)//按字节读取
    {
        cout << str<<" ";
        count++;
        if (count == 9)
        {
            printf("\n");
            count = 0;
        }

    }*/
    while (fin.getline(str, BUFFER_MAX))//按行读取
    {
        cout << str << endl;
    }
    fin.close();

}

/*                                                              主测试头部分
*/

void use_Menu(List *& head)
{

    //head->pnext = NULL;可有可无
    int count = 0;//用于接收统计信息的条数
    int chose;
    while (1)
    {
        chose = Menu();//显示菜单，并读取用户输入
        switch (chose)//根据选择进行调用函数
        {
        case 1:Create_information_list(*head); break;
        case 2:Insert_of_list(*head);
            break;
        case 3: information_output(*head);
            break;
        case 4:count=inferation_List_length(*head);
            cout << "表中共有" << count << "条记录信息"<< endl;
            break;
        case 5:serach_by_ID(*head);
            break;
```

```
		case 6:serach_by_name(*head);
			break;
		case 7:Clear_all_data(*head);
			break;
		case 8:Destroy_People_of_ID(*head);
			break;
		case 9:change_data_ID(*head);
			break;
		case 10:selection_sort_by_ID(*head);
			break;
		case 11: sieve_method_by_sex(*head);
			break;
		case 12: sieve_method_by_duty(*head);
			break;
		case 13:Output_of_file(*head);
			break;
		case 14:Input_of_file(*head);
			break;
		case 0:exit(0);
			printf("End of the Employee information management system !");//关闭员工信息管理系统
			break;
		default:printf("Your input was errored!"); break;
		}
	}
}
int main()
{
	List	*head=new List ;//为头结点申请堆中的内存
	use_Menu(head);//调用函数
//	system("pause");
return 0;
}
```