电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2018091602007

姓 名 王乐卿

（实验） 课程名称 程序设计与算法基础I

理论教师 罗绪成

实验教师 罗绪成、朱相印

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：王乐卿 学号：2018091602007 指导教师：罗绪成**

**实验地点：基础实验大楼535 实验时间：2018/12/22**

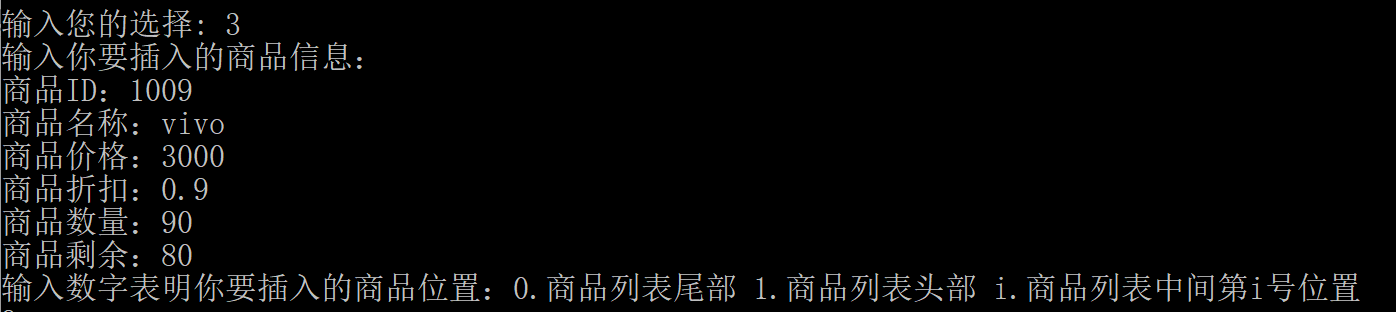
1. **实验名称：超市商品管理系统链表实现**
2. **实验学时：4学时**
3. **实验目的：**
4. 掌握单链表的定义和使用方法
5. 掌握单链表的建立方法
6. 掌握单链表中节点的查找与删除
7. 掌握输出单链表节点的方法
8. 掌握链表节点排序的一种方法
9. 掌握C语言创建菜单的方法
10. 掌握结构体的定义和使用方法
11. **实验原理**
    1. 格式化输入/输出：printf/scanf/putchar/getchar等；
    2. 表达式：赋值、自增、求值等；
    3. 结构体：结构是可能具有不同类型的值（成员）的集合，结构的元素（成员）可能具有不同的类型。每种结构代表一种新的定义域，所以每个结构都为他的成员设置了独立的名字空间；
    4. 指针：本实验主要运用了结构指针；
    5. 指针的高级运用：如动态储存分配：malloc(),空指针NULL的运用，释放储存空间：free()；二重指针等；
    6. 链表：运用链表储存商品信息，并对此作插入、排序、修改、删除、查找、显示等处理。
    7. 数组：本实验主要运用了结构指针数组，数组的每个元素为一个指针，指向一个存有商品信息和指向下一个结构指针的结构体；
    8. 选择语句：主要运用了if语句和switch语句；
    9. 循环语句：可以通过for(;;)和while(1)，来实现无限循环，直至输入退出系统的命令。
    10. 函数：本实验将超市管理系统的不同功能划分为不同的函数，在main函数中调用这些函数达到实现完整功能的目的。并且在函数调用中，分为设置全局变量和main函数中局部变量两种，若非全局变量，则需要在函数调用中将参数指定为指针。
    11. 预处理器：本实验定义了大量的宏，目的是为了增加程序的可读性。
    12. 文件处理：文件指针的运用：FILE ；用于文本文件的模式字符串；文件操作，如文件打开/关闭：fopen()/fclose();块的输入/输出：fwrite() /fread();
12. **实验内容**

用C语言+单链表数据结构实现一个小型的超市商品管理系统，该系统需要具备商品信息录入、商品信息修改、商品信息删除、商品信息查找、商品信息的插入这几个功能。具体实现步骤如下（注：图示为建议显示内容和格式，可自行增加显示内容）:

* 1. 软件界面控制:实现一个数字选项式的启动界面，其中至少包含显示所有商品信息、商品信息插入、商品信息修改、商品信息删除、商品信息查找、商品价格排序、退出系统并保存7个选项。并且这些功能可以循环调用。



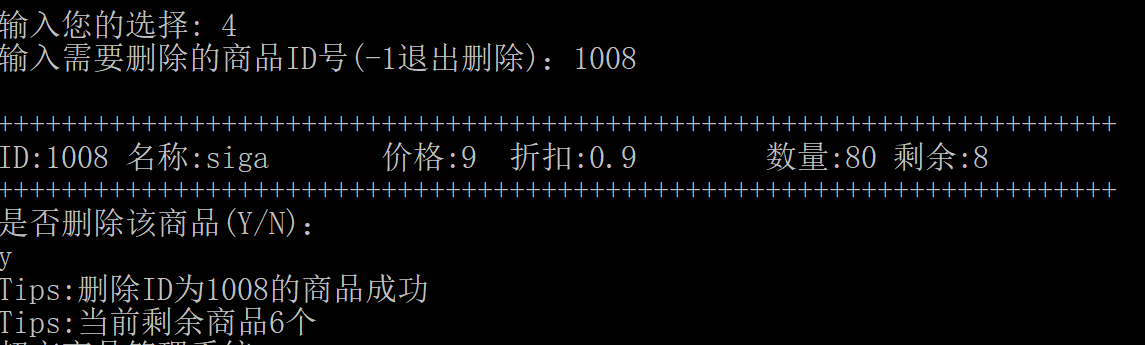
* 1. 商品信息的初始化: 定义链表并初始化。实现从已有的商品信息文件中读入商品信息，并且分配内存保存至链表中。如1）里的图示所示，从文件中读取了7个商品记录。
  2. 商品信息的增加：实现一个函数完成单个商品信息的增加，接受用户的输入的各项信息，然后保存至链表结点。同时实现可以根据用户的输入，将该结点插入到列表的头部，尾部，或者中间i号位置。



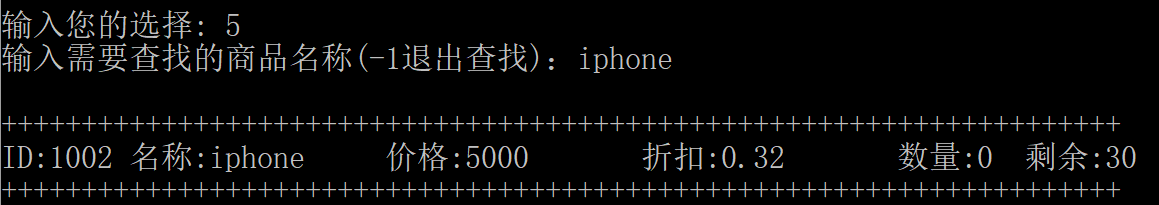
* 1. 商品信息的修改: 实现一个函数完成商品信息的修改功能，实现可以根据商品的名称或者ID修改商品信息。其中用字符串比较的方式来查找待修改商品。



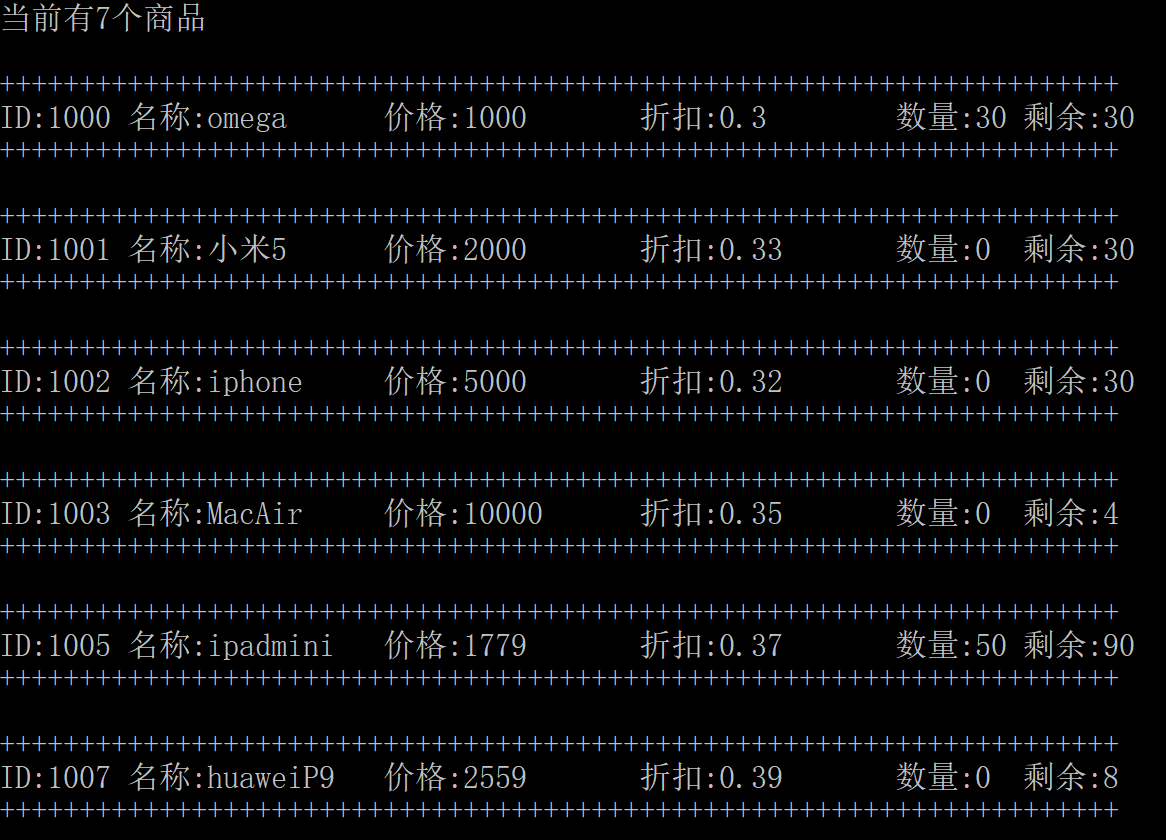
* 1. 商品信息的删除: 实现一个函数，实现根据商品的名称来删除对应的商品信息的功能，商品查找通过字符串比较的方式，查找到后释放对应指针指向的内存区域，完成删除。



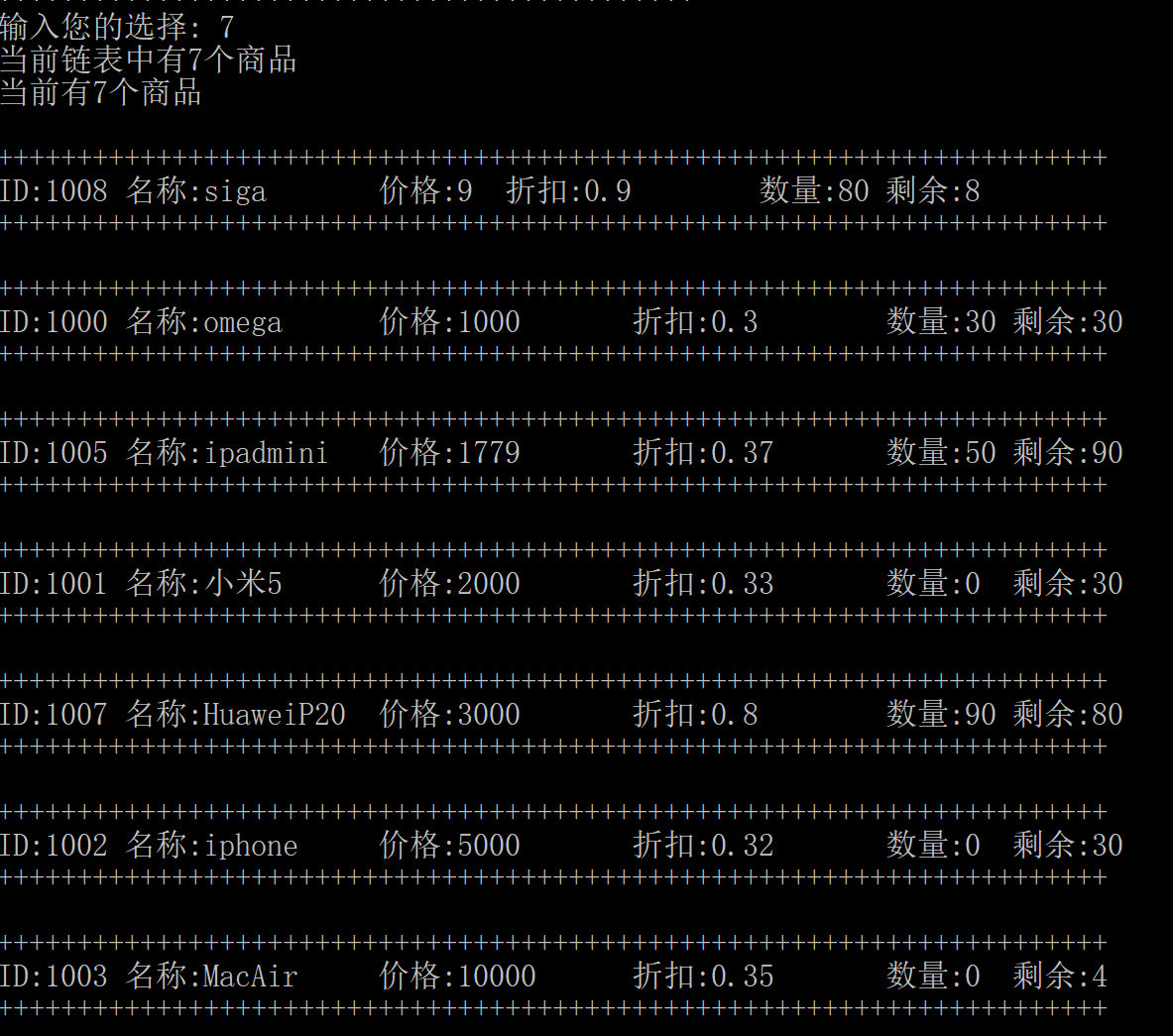
* 1. 商品信息的查找: 实现一个函数，函数的功能是根据输入的商品名称来查找对应的商品信息，商品名称的判断用字符串比较的方式来实现，然后调用格式化显示查找到的商品信息。



* 1. 所有商品信息的显示：实现一个函数，该函数的功能是将链表中所有的商品信息以格式化的方式打印输出到屏幕上。如下图所示的例子



* 1. 对商品按价格排序：实现一个函数，可以根据链表中的商品的价格，对商品进行排序，排序算法采用冒泡排序实现，最后将排序后的链表打印至屏幕。



* 1. 退出系统，并保存: 实现一个文件写入函数将所有信息的改动写入到商品信息文件，0然后清理系统运行过程中已分配的内存。

1. **实验器材（设备、元器件）：**

个人电脑一台

1. **实验步骤**
   1. 定义用于表示某种商品的所有信息的结构体，将其定义为全局变量，并且定义***链表结构***用来组织所有的商品信息库中的商品信息。
   2. 定义并实现一个函数: void info\_init(void)。该函数初始化一个链表，然后完成从一个txt文件读入商品链表库并使用相应的商品信息来初始化商品库（即初始化链表），在读入的时候每读到一条商品信息就实时的动态分配内存来把信息放到分配得到的链表结点指针指向的内存单元中，然后把链表节点加入到商品链表中。
   3. 定义并实现函数:void save\_exit(void)。该函数完成将系统运行期间改动过的商品信息库写回到存放商品信息的txt文件中，然后退出超市管理系统。
   4. 定义并实现函数eatline(void)。该函数用于去除多余空格符输入。
   5. 定义并实现函数:void output\_all(void)。以格式化的方式，完成将商品链表中的每项信息打印到标准输出(即屏幕上)。
   6. 定义并实现函数:void info\_change(void)。该函数完成商品信息的修改功能，其中要求用户输入需要修改的某项商品的名称或者ID，然后对名称或者ID进行查找，找到则输出商品现有信息，然后提示修改信息，让继续输入该商品新的的各项信息，并提示修改成功，返回主菜单;如果没有找到则提示对应商品未找到，直接返回主菜单。
   7. 定义并实现函数:void info\_dele(void)。该函数完成删除某条商品库中信息的功能， 通过输入的某项商品的名称或者ID删除对应的信息，如果在商品库中找到对应的商品便显示商品信息，提示是否确认删除该商品，若确认，则删除商品信息(即释放指针所指向的内存，并把该指针赋值为 NULL，同时完成链表的前后重新衔接)，并提示删除成功;如果没有找到该商品要提示没有找到该商品信息，在两种选择下都回到上一步选择界面 ，且在任意过程中用户可选择返回主菜单。
   8. 定义并实现函数 info\_deleteall(void)。该函数实现删除链表中所有内容，释放内存。
   9. 定义并实现函数:void info\_search(void)。该函数完成商品信息的查找功能，然后通过输入某种商品信息的名称或者ID来检索商品信息库，查找到则显示该商品的详细信息，没有查找到则提示没有该商品，然后返回主菜单，且在任意过程中用户可选择返回主菜单。
   10. 定义并实现函数:void info\_insert(void)。该函数完成商品信息的插入，在插入之前动态的分配内存用来存储插入的商品信息，然后把指向该内存的指针加入到链表中，加入链表时需要选择在头部插入或者在尾部插入，或者在链表范围内的i号位置插入;插入成功则显示插入成功提示，并返回主菜单。在插入之前必须考虑整个信息库的限定容量，如果超过上限要给用户以提示 ，还需要判断插入商品是否已经存在，若存在，则显示错误信息，返回主菜单。
   11. 定义并实现函数：info\_sort(void)。该函数实现将链表中的所有商品按照商品价格从低到高排序。
   12. 定义并实现函数 welcome\_menu(voif)。该函数用于显示欢迎界面。
   13. 定义并实现函数 exit\_menu(voif)。该函数用于显示退出界面
   14. 实现程序的入口函数即 main 函数，然后for(;;)实现无限循环完成以上功能的循环调用，直至选择正常存盘退出，其余退出皆不保存，且若输入非法指令，则提示错误信息并退出。。
   15. 编译、调试程序直至达到实验要求
2. **实验结果与分析（含重要数据结果分析或核心代码流程分析）**
   1. 读入信息

|  |
| --- |
| void info\_init(void) /\*读入信息\*/  {  FILE \*fp;  fp=fopen("Goodsinfo.txt", "rb");  struct list \*p;  if (fp== NULL)  {  perror("读入信息文件打开失败：");  exit(EXIT\_FAILURE);  }  for (;;)  {    new\_item = malloc(sizeof(struct list));  if (new\_item == NULL)  {  perror("读入信息分配空间失败：");  exit(EXIT\_FAILURE);  }  if (fread(new\_item, sizeof(struct list), 1, fp) == 1)  {  p = fgoods;  if (p == NULL) {  new\_item->next = NULL;  fgoods = new\_item;  }  else  {  for (p = fgoods; p->next != NULL; p = p->next)  ;  new\_item->next = p->next;  p->next = new\_item;  }  }  else  break;  }  fclose(fp);  } |

* 1. 储存并退出

|  |
| --- |
| void save\_exit(void) /\*储存并推出\*/  {  FILE \*fp;  if (fgoods == NULL)  {  printf("列表空空如也");  exit(EXIT\_FAILURE);  }  fp = fopen("Goodsinfo.txt", "wb");  if (fp == NULL)  {  printf("保存信息文件打开失败");  exit(EXIT\_FAILURE);  }  while (fgoods != NULL)  {  if (fwrite(fgoods, sizeof(struct list), 1, fp) == 1)  fgoods = fgoods->next;  }  fclose(fp);  exit(EXIT\_SUCCESS);  } |

* 1. 输出所有商品信息

|  |
| --- |
| void output\_all(void) /\*打印所有\*/  {  int n = 0;  struct list \*p;  for (p = fgoods; p != NULL; p = p->next)  {  printf("\nID:%-10s名称:%-10s价格%-10d折扣:%-10.2f数量：%-10d剩余:%-10d", p->goods\_id,p->goods\_name,p->goods\_price,p->goods\_discount,p->goods\_amount,p->goods\_remain);  n++;  }  printf("\n打印完毕!当前共有%d个商品\n\n",n);  } |

* 1. 添加一条商品信息

|  |
| --- |
| void info\_insert(void) /\*插入某个信息\*/  {  struct list\* new\_good,\*pre=NULL,\*p;  int choice,i;  new\_good = malloc(sizeof(struct list));  printf("输入你要增加的商品信息：\n");  printf("\n商品ID：");  scanf("%s", &new\_good->goods\_id);  for (p = fgoods;p!=NULL; p = p->next)  {  if (strcmp(new\_good->goods\_id, p->goods\_id) == 0)  {  printf("\n该商品ID已存在！请重新输入：");  scanf("%s", &new\_good->goods\_id);  p = fgoods;  }  }  printf("\n商品名称：");  scanf("%s", &new\_good->goods\_name);  for (p = fgoods; p != NULL; p = p->next)  {  if (strcmp(new\_good->goods\_name, p->goods\_name) == 0)  {  printf("\n该商品名称已存在！请重新输入：");  scanf("%s", &new\_good->goods\_name);  p = fgoods;  }  }  printf("\n商品价格：");  scanf("%d", &new\_good->goods\_price);  printf("\n商品折扣：");  scanf("%f", &new\_good->goods\_discount);  printf("\n商品数量：");  scanf("%d", &new\_good->goods\_amount);  printf("\n商品剩余：");  scanf("%d", &new\_good->goods\_remain);  for (;;)  {  if (new\_good->goods\_remain > new\_good->goods\_amount)  {  printf("错误！剩余数大于商品总数！请重新输入：");  scanf("%d", &new\_good->goods\_remain);  }  else  break;  }  printf("\n输入数字表明你要插入的位置：\n1：插入表头\n0：插入表尾\ni：插入第i号位\n您的选择：");  scanf("%d", &choice);  switch (choice)  {  case 1: {  new\_good->next = fgoods;  fgoods = new\_good;  printf("\n插入表头成功！返回主菜单中...\n");  }break;  case 0: {  p = fgoods;  if (p == NULL){  new\_good->next = NULL;  fgoods = new\_good; }  else  {  for (p = fgoods; p->next != NULL; p = p->next)  ;  new\_good->next = NULL;  p->next = new\_good;  }  printf("插入表尾成功！返回主菜单中...\n");  }break;  default: {  p = fgoods;  for (i = 1; i < choice&&p->next!=NULL; i++)  {  pre = p;  p = p->next;  }  if (p != NULL&&pre!=NULL)  {  pre->next = new\_good;  new\_good->next = p;  printf("插入第%d号位成功，返回主菜单中...\n", choice);    }  else  {  printf("插入位置超出范围，返回主菜单中...\n");  return;  }    }break;  }  } |

* 1. 删除一条商品信息

|  |
| --- |
| void info\_dele(void) /\*删除某个信息\*/  {  int choice;  char code[30],sure;  struct list \*p,\*pre;  p = fgoods;  pre = NULL;  printf("根据商品ID删除：输入1\n根据商品名称删除：输入2\n退出删除：输入-1\n您的选择：");  scanf("%d", &choice);  if (choice == -1)  return;  else if (choice == 1) {  printf("\n输入需要删除的商品ID\n退出查找：输入-1：\n");  scanf("%s", code);  if (strcmp(code, "-1") == 0)  return;  for (pre=NULL; p!=NULL; pre=p,p = p->next)  {  if (strcmp(p->goods\_id, code) == 0)  break;  }  if (p == NULL)  {  printf("未找到该ID商品，返回主菜单中...\n");  return;  }  }  else if (choice == 2) {  printf("输入需要删除的商品名称:(退出删除：输入-1：)\n");  scanf("%s", code);  if (strcmp(code, "-1") == 0)  return;  for (pre=NULL; p!=NULL; pre=p,p= p->next)  {  if (strcmp(p->goods\_name, code) == 0)  break;  }  if (p == NULL)  {  printf("未找到该ID商品，返回主菜单中...\n");  return;  }  }  printf("ID:%-10s名称:%-10s价格%-10d折扣:%-10f数量：%-10d剩余:%-10d\n", p->goods\_id, p->goods\_name, p->goods\_price, p->goods\_discount, p->goods\_amount, p->goods\_remain);  printf("确认删除？是：Y否：N\n您的选择：");  eatline();  scanf("%c", &sure);  sure = toupper(sure);  switch (sure)  {  case 'Y': {  if (pre == NULL)  fgoods = fgoods->next;  else  pre->next = p->next;  free(p);  printf("\n删除成功！返回主菜单中...\n");  return;  }break;  case 'N': {  printf("\n取消删除成功！返回主菜单中...\n");  return;  }break;  }  } |

* 1. 删除所有商品信息

|  |
| --- |
| void info\_deleteall(void) /\*删除所有内容\*/  {  char sure;  struct list \*p;  printf("确认清空列表？是：Y否：N\n您的选择：");  scanf("%c", &sure);  sure = toupper(sure);  switch (sure)  {  case 'Y': {  while(fgoods!=NULL)  {  p = fgoods;  fgoods = fgoods->next;  free(p);    }  printf("\n删除成功！返回主菜单中...\n");  }break;  case 'N': {  printf("\n取消删除成功！返回主菜单中...\n");  return;  }break;  }  } |

* 1. 查找一条商品记录

|  |
| --- |
| void info\_search(void) /\*查找某个信息\*/  {  int choice;  char code[30];  struct list \*p;  printf("根据商品ID查找：输入1\n根据商品名称查找：输入2\n退出查找：输入-1\n您的选择：");  scanf("%d", &choice);  if (choice == -1)  return;  else if (choice == 1) {  printf("\n输入需要查找的商品ID:(退出查找：输入-1：)");  scanf("%s", code);  if (strcmp(code, "-1") == 0)  return;  for (p = fgoods; p != NULL; p = p->next)  {  if (strcmp(p->goods\_id, code) == 0)  break;  }  if (p == NULL)  {  printf("\n未找到该ID商品，返回主菜单中...\n");  return;  }  }  else if (choice == 2) {  printf("输入需要查找的商品名称:(退出查找：输入-1：)");  scanf("%s", code);  if (strcmp(code, "-1") == 0)  return;  for (p = fgoods; p != NULL; p = p->next)  {  if (strcmp(p->goods\_name, code) == 0)  break;  }  if (p == NULL)  {  printf("\n未找到该ID商品，返回主菜单中...\n");  return;  }  }  printf("\nID:%-10s名称:%-10s价格%-10d折扣:%-10f数量：%-10d剩余:%-10d\n", p->goods\_id, p->goods\_name, p->goods\_price, p->goods\_discount, p->goods\_amount, p->goods\_remain);  } |

* 1. 修改一条商品信息

|  |
| --- |
| void info\_change(void) /\*修改某个商品\*/  {  int choice;  char code[30];  struct list \*p;  printf("\n根据商品ID修改：输入1\n根据商品名称修改：输入2\n退出修改：输入-1\n您的选择：");  scanf("%d", &choice);  if (choice == -1)  return;  else if (choice == 1) {  printf("\n输入需要修改的商品ID\n退出修改：输入-1：\n");  scanf("%s", code);  if(strcmp(code,"-1")==0)  return ;  for (p = fgoods; p!=NULL; p = p->next)  {  if (strcmp(p->goods\_id, code) == 0)  break;  }  if (p == NULL)  {  printf("未找到该ID商品，返回主菜单中...\n");  return;  }  }  else if (choice == 2) {  printf("输入需要修改的商品名称:(退出查找：输入-1：)");  scanf("%s", code);  if (strcmp(code, "-1") == 0)  return;  for (p = fgoods; p!=NULL; p = p->next)  {  if (strcmp(p->goods\_name, code) == 0)  break;  }  if (p == NULL)  {  printf("\n未找到该ID商品，返回主菜单中...\n");  return;  }    }  printf("\nID:%-10s名称:%-10s价格%-10d折扣:%-10f数量：%-10d剩余:%-10d", p->goods\_id, p->goods\_name, p->goods\_price, p->goods\_discount, p->goods\_amount, p->goods\_remain);  printf("\n输入新的产品信息：\n");  printf("商品ID：");  scanf("%s", &p->goods\_id);  printf("\n商品名称：");  scanf("%s", &p->goods\_name);  printf("\n商品价格：");  scanf("%d", &p->goods\_price);  printf("\n商品折扣：");  scanf("%f", &p->goods\_discount);  printf("\n商品数量：");  scanf("%d", &p->goods\_amount);  printf("\n商品剩余：");  scanf("%d", &p->goods\_remain);  for (;;)  {  if (p->goods\_remain > p->goods\_amount)  {  printf("\n错误！剩余数大于商品总数！请重新输入：");  scanf("%d", &p->goods\_remain);  }  else  break;  }  printf("\n修改成功！修改后商品信息如下：\n");  printf("ID:%-10s名称:%-10s价格%-10d折扣:%-10f数量：%-10d剩余:%-10d\n", p->goods\_id, p->goods\_name, p->goods\_price, p->goods\_discount, p->goods\_amount, p->goods\_remain);  } |

* 1. 链表按价格排序

|  |
| --- |
| void info\_sort(void) /\*对商品价格进行排序\*/  {  struct list \*p, \*first, \*second,\*temp;;  int num = 0,i,j,n=0;  p = fgoods;  while (p != NULL)  {  num++;  p = p->next;  }  switch (num)  {  case 0:  {  printf("无商品\n");  return;  }break;  default:  {  for (i = 1; i < num; i++)  {  first = fgoods;  second = fgoods->next;  for (j = 0; j < num - i; j++)  {  if (first->goods\_price > second->goods\_price)  {  temp = malloc(sizeof(struct list));  \*temp = \*first;  strcpy(first->goods\_id, second->goods\_id);  strcpy(first->goods\_name, second->goods\_name);  first->goods\_amount = second->goods\_amount;  first->goods\_discount = second->goods\_discount;  first->goods\_price = second->goods\_price;  first->goods\_remain = second->goods\_remain;  strcpy(second->goods\_id, temp->goods\_id);  strcpy(second->goods\_name, temp->goods\_name);  second->goods\_amount = temp->goods\_amount;  second->goods\_discount = temp->goods\_discount;  second->goods\_price = temp->goods\_price;  second->goods\_remain = temp->goods\_remain;  free(temp);  }  first = first->next;  second = second->next;  }  }  printf("排序完毕！\n");  }break;  }  for (p = fgoods; p != NULL; p = p->next)  {  printf("ID:%-10s名称:%-10s价格%-10d折扣:%-10f数量：%-10d剩余:%-10d\n", p->goods\_id, p->goods\_name, p->goods\_price, p->goods\_discount, p->goods\_amount, p->goods\_remain);  n++;  }  printf("当前共有%d个商品！\n", n);  } |

* 1. 主界面实现

|  |
| --- |
| void welcome\_menu(void) /\*欢迎菜单\*/  {  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("\*\*\*\*\*\* 欢迎使用超市商品信息储存系统 \*\*\*\*\*\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 功能菜单 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf(" 1：显示商品信息 \n");  printf(" 2：商品信息修改 \n");  printf(" 3：商品信息录入 \n");  printf(" 4：商品信息删除 \n");  printf(" 5：商品信息查找 \n");  printf(" 6：存盘退出系统 \n");  printf(" 7：商品信息排序 \n");  printf(" 8：删除所有内容 \n");  }  void exit\_menu(void) /\*退出菜单\*/  {  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\* 正在退出 \*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\* 感谢您的使用 \*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  } |

1. **总结及心得体会：**
2. Switch函数对分支结构的程序编写有很好的适用性，且可以使得代码思路更为清晰。
3. 利用结构指针数组和动态分配内存的技术可以比直接使用结构数组更加节约内存使用，并且使得实验要求中的删除等操作更为的简单，实现起来更加容易。
4. 应当注意缓存区内的空字符等其他输入，在本实验中我使用了eatline()函数，在输入字符前使用。
5. 通过for(;;)或while(1)循环和switch预计可以达到系统功能的反复调用。
6. 编写代码过程中应当尽量避免小错误的发生，往往一个小小的错误可能需要debug半天，以下是容易碰到的错误点：其一，赋值与判等符号容易搞错，或者说容易忘记加一个等号；其二，特殊情况需要额外考虑，如指针为空，这一点在插入一个信息是额外需要注意；其三，拼写错误，比如printf写成pirntf，打字快了就可能出错，然后中英文的逗号分号也要注意；其四，指针的重数，函数的调用，全局变量的值都应该小心谨慎。
7. 有关printf，在开始输入“\n”感觉比在语句末尾的冒号后面加“\n”更加美观，否则会在调试界面出现换行，很难看。
8. 各种括号、语句的缩进应当注意，否则会显得很乱，过一段时间就看不懂了，故而在程序编写过程中应当注意缩进风格的异同。
9. 注释的使用、变量名、函数名的选择都很重要。这既决定了你的代码能否被别人所理解，也决定了你写代码的速度，拥有显然直观的名字可以在很大程度上减少代码的正确性。
10. 应当重视并熟练掌握指针的运用，指针感觉就像是C语言的灵魂，所有的操作大多都基于指针，没有指针寸步难行。指针的存在对各种功能的实现和简化有不可或缺的作用。
11. **对本实验过程及方法、手段的改进建议：**
12. 最好给定固定的测试数据和格式，若是学生自己来输入数据的话，未必能测试出一些比较隐蔽的bug。
13. 实验还是过于死板了这个，应当更加注重于发挥学生的创造力和创新能力。
14. 实验所包含的知识点还不够全面，主要是一些固定知识点的反复使用，感觉除了巩固了一下之外并没有什么更大的帮助，应当加入更多功能使得程序更加复杂化，全面化。

**报告评分：**

**指导教师签字：**