

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称：实验五WIN32编程**

**实验时间： 2019-5-7，14：00-17：30 实验地点： 南一楼804室070号实验台**

**指导教师： 李专**

**专业班级：计算机科学与技术201705班**

**学 号： U201714726 姓 名： 王明明**

**同组学生： 无 报告日期： 2019年 5 月 16日**

**原创性声明**

  本人郑重声明：本报告的内容由本人独立完成，有关观点、方法、数据和文献等的引用已经在文中指出。除文中已经注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品或成果，不存在剽窃、抄袭行为。

特此声明！

学生签名：王明明

日期：2019.5.16

成绩评定

|  |  |  |
| --- | --- | --- |
| 实验完成质量得分（70分）（实验步骤清晰详细深入，实验记录真实完整等） | 报告撰写质量得分（30分）（报告规范、完整、通顺、详实等） | 总成绩（100分） |
|  |  |  |

指导教师签字：

                    日期：

**目录**

[1 实验目的与要求 3](#_Toc9882641)

[2 实验内容 3](#_Toc9882642)

[3 实验过程 4](#_Toc9882643)

[3.1 任务1 4](#_Toc9882644)

[3.1.1 设计思想 4](#_Toc9882645)

[3.1.2 变量 4](#_Toc9882646)

[3.1.3 源程序 5](#_Toc9882647)

[3.1.4 实验步骤 14](#_Toc9882648)

[3.1.5 实验记录与分析 14](#_Toc9882649)

[4 总结与体会 16](#_Toc9882650)

[参考文献 18](#_Toc9882651)

# 实验目的与要求

1. 熟悉WIN32程序的设计和调试方法；
2. 熟悉宏汇编语言中INVOKE、结构变量、简化段定义等功能；
3. 进一步理解机器语言、汇编语言、高级语言之间以及实方式、保护方式之间的一些关系。

# 实验内容

编写一个基于窗口的WIN32程序，实现网店商品信息管理程序的推荐度计算及商品信息显示的功能（借鉴实验三的一些做法），具体要求如下描述。

功能一：编写一个基于窗口的WIN32程序的菜单框架，具有以下的下拉菜单项：

File Action Help

Exit Recommendation About

List

点菜单File下的Exit选项时结束程序；点菜单Help下的选项About，弹出一个消息框，显示本人信息，类似图5.1所示。点菜单Action下的选项Recommendation、List将分别实现计算推荐度或显示SHOP所有商品信息的功能（详见功能二的描述）。



图3-1 菜单示例

功能二：要求采用结构变量存放商品的相关信息。商品数至少定义5种。

（1）点菜单项Recommendation时，按照实验三的方法计算所有商品的推荐度。用TD32观察计算结果。

（2）点菜单项List时，要求能在窗口中列出SHOP的所有商品的信息。具体显示格式自行定义，可以参照图5.2的样式（不要求用中文）。



图5-2 商品信息显示示意图

# 实验过程

## 任务1

### 设计思想

该基于窗口的程序是基于WIN32的标准框架实现的。该程序中还用到了WIN32的标准框架，其中包含了主程序、窗口主程序、窗口消息、处理程序以及用户处理程序。操作系统首先执行主程序，待主程序获得与本程序有关的基本信息后，在调用窗口主程序。窗口主程序创建窗口后，将说道的消息通过操作系统发送给窗口消息处理程序。窗口消息处理程序判断收到消息的类型，完成相应的功能

菜单栏的实现用到了.rc文件。在主程序winmain中，程序将对菜单资源进行装载。再在窗口过程中对相应的菜单功能进行设计，即可完成相关菜单功能的实现。在实现菜单功能时，需要对子函数进行调用。在本次实验中，需要编写的子程序为计算推荐度和显示商品信息的子函数。

### 变量

ClassName：字节变量，存储窗口类字符串‘TryWinClass’

AppName：字节变量，存储窗口名字符串'Our First Window'

MenuName：字节变量，存储菜单名字符串'MyMenu'

DlgName：字节变量，存储字符串'MyDialog'

AboutMsg：字节变量，存储about显示的字符串'I Am CS1705 WMM',0

hInstance：字变量，存储句柄

CommandLine：字变量，存储命令行参数

### 源程序

1.demo.asm源文件

.386

.model flat,stdcall

option casemap:none

WinMain proto :DWORD,:DWORD,:DWORD,:DWORD

WndProc proto :DWORD,:DWORD,:DWORD,:DWORD

Display proto :DWORD

Tjd proto :DWORD

include menuID.INC

include D:\masm32\INCLUDE\windows.inc

include D:\masm32\INCLUDE\user32.inc

include D:\masm32\INCLUDE\kernel32.inc

include D:\masm32\INCLUDE\gdi32.inc

include D:\masm32\INCLUDE\shell32.inc

includelib D:\masm32\LIB\user32.lib

includelib D:\masm32\LIB\kernel32.lib

includelib D:\masm32\LIB\gdi32.lib

includelib D:\masm32\LIB\shell32.lib

goods struct

gname db 10 dup(0)

discount db 0

inprice db 0

outprice db 0

innum db 0

outnum db 0

recommendation db 0

goods ends

.data

ClassName db 'TryWinClass',0

AppName db 'Our First Window',0

MenuName db 'MyMenu',0

DlgName db 'MyDialog',0

AboutMsg db 'I Am CS1705 WMM',0

hInstance dd 0

CommandLine dd 0

buf goods <>

goods <'Pen',10,35,56,70,25,'0'>

goods <'Book',9,12,30,25,5,'0'>

goods <'Pencil',9,1,4,50,6,'0'>

goods <'Ruler',8,2,5,50,10,'0'>

goods <'Bumf',10,3,1,100,1,'0'>

msg\_gname db 'GName',0

msg\_discount db 'Discount',0

msg\_inprice db 'In\_Price',0

msg\_outprice db 'Out\_price',0

msg\_innum db 'In\_Num',0

msg\_outnum db 'Out\_Num',0

msg\_recommendation db 'Recommendation',0

discount db 2,1,1,1,2, '10','9','9','8','10'

inprice db 2,2,1,1,1, '35','12','1','2','3'

outprice db 2,2,1,1,1, '56','30','4','5','1'

innum db 2,2,2,2,3, '70','25','50','50','100'

outnum db 2,1,1,2,1, '25','5','6','10','1'

;;recommendation db 2,0,0,0, '0'

menuItem db 0 ;当前菜单状态, 1=处于list, 0=Clear

.code

Start: invoke GetModuleHandle,NULL

mov hInstance,eax

invoke GetCommandLine

mov CommandLine,eax

invoke WinMain,hInstance,NULL,CommandLine,SW\_SHOWDEFAULT

invoke ExitProcess,eax

;;

WinMain proc hInst:DWORD,hPrevInst:DWORD,CmdLine:DWORD,CmdShow:DWORD

LOCAL wc:WNDCLASSEX

LOCAL msg:MSG

LOCAL hWnd:HWND

invoke RtlZeroMemory,addr wc,sizeof wc

mov wc.cbSize,SIZEOF WNDCLASSEX

mov wc.style, CS\_HREDRAW or CS\_VREDRAW

mov wc.lpfnWndProc, offset WndProc

mov wc.cbClsExtra,NULL

mov wc.cbWndExtra,NULL

push hInst

pop wc.hInstance

mov wc.hbrBackground,COLOR\_WINDOW+1

mov wc.lpszMenuName, offset MenuName

mov wc.lpszClassName,offset ClassName

invoke LoadIcon,NULL,IDI\_APPLICATION

mov wc.hIcon,eax

mov wc.hIconSm,0

invoke LoadCursor,NULL,IDC\_ARROW

mov wc.hCursor,eax

invoke RegisterClassEx, addr wc

INVOKE CreateWindowEx,NULL,addr ClassName,addr AppName,\

WS\_OVERLAPPEDWINDOW,CW\_USEDEFAULT,\

CW\_USEDEFAULT,CW\_USEDEFAULT,CW\_USEDEFAULT,NULL,NULL,\

hInst,NULL

mov hWnd,eax

INVOKE ShowWindow,hWnd,SW\_SHOWNORMAL

INVOKE UpdateWindow,hWnd

;;

MsgLoop: INVOKE GetMessage,addr msg,NULL,0,0

cmp EAX,0

je ExitLoop

INVOKE TranslateMessage,addr msg

INVOKE DispatchMessage,addr msg

jmp MsgLoop

ExitLoop: mov eax,msg.wParam

ret

WinMain endp

WndProc proc hWnd:DWORD,uMsg:DWORD,wParam:DWORD,lParam:DWORD

LOCAL hdc:HDC

LOCAL ps:PAINTSTRUCT

.IF uMsg == WM\_DESTROY

invoke PostQuitMessage,NULL

.ELSEIF uMsg == WM\_KEYDOWN

.IF wParam == VK\_F1

;;your code

.ENDIF

.ELSEIF uMsg == WM\_COMMAND

.IF wParam == IDM\_FILE\_EXIT

invoke SendMessage,hWnd,WM\_CLOSE,0,0

.ELSEIF wParam == IDM\_FILE\_LIST

mov menuItem, 1

invoke InvalidateRect,hWnd,0,1 ;擦除整个客户区

invoke UpdateWindow, hWnd

.ELSEIF wParam == IDM\_FILE\_CLEAR

mov menuItem, 0

invoke InvalidateRect,hWnd,0,1 ;擦除整个客户区

invoke UpdateWindow, hWnd

.ELSEIF wParam == IDM\_FILE\_RECOMMENDATION

invoke Tjd,5

.ELSEIF wParam == IDM\_HELP\_ABOUT

invoke MessageBox,hWnd,addr AboutMsg,addr AppName,0

.ENDIF

.ELSEIF uMsg == WM\_PAINT

invoke BeginPaint,hWnd, addr ps

mov hdc,eax

.IF menuItem == 1

invoke Display,hdc

.ENDIF

invoke EndPaint,hWnd,addr ps

.ELSE

invoke DefWindowProc,hWnd,uMsg,wParam,lParam

ret

.ENDIF

xor eax,eax

ret

WndProc endp

Display proc hdc:HDC

XX equ 10

YY equ 10

XX\_GAP equ 100

YY\_GAP equ 30

invoke TextOut,hdc,XX+0\*XX\_GAP,YY+0\*YY\_GAP,offset msg\_gname,5

invoke TextOut,hdc,XX+1\*XX\_GAP,YY+0\*YY\_GAP,offset msg\_discount,8

invoke TextOut,hdc,XX+2\*XX\_GAP,YY+0\*YY\_GAP,offset msg\_inprice,8

invoke TextOut,hdc,XX+3\*XX\_GAP,YY+0\*YY\_GAP,offset msg\_outprice,9

invoke TextOut,hdc,XX+4\*XX\_GAP,YY+0\*YY\_GAP,offset msg\_innum,6

invoke TextOut,hdc,XX+5\*XX\_GAP,YY+0\*YY\_GAP,offset msg\_outnum,7

invoke TextOut,hdc,XX+6\*XX\_GAP,YY+0\*YY\_GAP,offset msg\_recommendation,14

;;

invoke TextOut,hdc,XX+0\*XX\_GAP,YY+1\*YY\_GAP,offset buf[1\*16].gname,3

invoke TextOut,hdc,XX+1\*XX\_GAP,YY+1\*YY\_GAP,offset discount+5, discount

invoke TextOut,hdc,XX+2\*XX\_GAP,YY+1\*YY\_GAP,offset inprice+5, inprice

invoke TextOut,hdc,XX+3\*XX\_GAP,YY+1\*YY\_GAP,offset outprice+5, outprice

invoke TextOut,hdc,XX+4\*XX\_GAP,YY+1\*YY\_GAP,offset innum+5, innum

invoke TextOut,hdc,XX+5\*XX\_GAP,YY+1\*YY\_GAP,offset outnum+5, outnum

invoke TextOut,hdc,XX+6\*XX\_GAP,YY+1\*YY\_GAP,offset buf[1\*16].recommendation,1

;;

invoke TextOut,hdc,XX+0\*XX\_GAP,YY+2\*YY\_GAP,offset buf[2\*16].gname,4

invoke TextOut,hdc,XX+1\*XX\_GAP,YY+2\*YY\_GAP,offset discount+7, discount+1

invoke TextOut,hdc,XX+2\*XX\_GAP,YY+2\*YY\_GAP,offset inprice+7, inprice+1

invoke TextOut,hdc,XX+3\*XX\_GAP,YY+2\*YY\_GAP,offset outprice+7, outprice+1

invoke TextOut,hdc,XX+4\*XX\_GAP,YY+2\*YY\_GAP,offset innum+7, innum+1

invoke TextOut,hdc,XX+5\*XX\_GAP,YY+2\*YY\_GAP,offset outnum+7, outnum+1

invoke TextOut,hdc,XX+6\*XX\_GAP,YY+2\*YY\_GAP,offset buf[2\*16].recommendation,1

;;

invoke TextOut,hdc,XX+0\*XX\_GAP,YY+3\*YY\_GAP,offset buf[3\*16].gname,6

invoke TextOut,hdc,XX+1\*XX\_GAP,YY+3\*YY\_GAP,offset discount+8, discount+2

invoke TextOut,hdc,XX+2\*XX\_GAP,YY+3\*YY\_GAP,offset inprice+9, inprice+2

invoke TextOut,hdc,XX+3\*XX\_GAP,YY+3\*YY\_GAP,offset outprice+9, outprice+2

invoke TextOut,hdc,XX+4\*XX\_GAP,YY+3\*YY\_GAP,offset innum+9, innum+2

invoke TextOut,hdc,XX+5\*XX\_GAP,YY+3\*YY\_GAP,offset outnum+8, outnum+2

invoke TextOut,hdc,XX+6\*XX\_GAP,YY+3\*YY\_GAP,offset buf[3\*16].recommendation,1

;;

invoke TextOut,hdc,XX+0\*XX\_GAP,YY+4\*YY\_GAP,offset buf[4\*16].gname,5

invoke TextOut,hdc,XX+1\*XX\_GAP,YY+4\*YY\_GAP,offset discount+9, discount+3

invoke TextOut,hdc,XX+2\*XX\_GAP,YY+4\*YY\_GAP,offset inprice+10, inprice+3

invoke TextOut,hdc,XX+3\*XX\_GAP,YY+4\*YY\_GAP,offset outprice+10, outprice+3

invoke TextOut,hdc,XX+4\*XX\_GAP,YY+4\*YY\_GAP,offset innum+11, innum+3

invoke TextOut,hdc,XX+5\*XX\_GAP,YY+4\*YY\_GAP,offset outnum+9, outnum+3

invoke TextOut,hdc,XX+6\*XX\_GAP,YY+4\*YY\_GAP,offset buf[4\*16].recommendation,1

;;

invoke TextOut,hdc,XX+0\*XX\_GAP,YY+5\*YY\_GAP,offset buf[5\*16].gname,4

invoke TextOut,hdc,XX+1\*XX\_GAP,YY+5\*YY\_GAP,offset discount+10, discount+4

invoke TextOut,hdc,XX+2\*XX\_GAP,YY+5\*YY\_GAP,offset inprice+11, inprice+4

invoke TextOut,hdc,XX+3\*XX\_GAP,YY+5\*YY\_GAP,offset outprice+11, outprice+4

invoke TextOut,hdc,XX+4\*XX\_GAP,YY+5\*YY\_GAP,offset innum+13, innum+4

invoke TextOut,hdc,XX+5\*XX\_GAP,YY+5\*YY\_GAP,offset outnum+11, outnum+4

invoke TextOut,hdc,XX+6\*XX\_GAP,YY+5\*YY\_GAP,offset buf[5\*16].recommendation,1

ret

Display endp

Tjd proc goonum:DWORD

push eax

push ebx

push ecx

push edx

push edi

push esi

push ebp

MOV bx,word ptr goonum

mov ebp,0

LEA EBP,buf

add Ebp,16

LOPA1:

MOV AL,128

MOV AH,0

CWDE

MOV EDI,EAX

MOV AL,ds:[ebp+10]

MOV AH,0

CWDE

MOV ESI,EAX

MOV al,ds:[ebp+12]

CWDE

IMUL EAX,ESI

CDQ

MOV ECX,10

IDIV ECX

push eax

mov edx,0

MOV Al,ds:[ebp+11]

CWDE

IMUL EAX,EDI

CDQ

pop ecx

idiv ecx

MOV ESI,EAX

MOV Al,ds:[ebp+13]

CWDE

IMUL EAX,2

MOV ECX,EAX

push eax

mov edx,0

MOV Al,ds:[ebp+14]

CWDE

IMUL EAX,EDI

CDQ

pop ecx

IDIV ECX

ADD eax,esi ;计算推荐度

cmp ax,100

jg ranka

cmp ax,50

jg rankb

cmp ax,10

jg rankc

mov DS:[EBP+15],BYTE PTR 'F'

jmp LOPA2

ranka: mov DS:[EBP+15],BYTE PTR 'A'

jmp LOPA2

rankb: mov DS:[EBP+15],byte ptr 'B'

jmp LOPA2

rankc: mov DS:[EBP+15],byte ptr 'C'

LOPA2: ADD EBP,16

DEC bx

JNZ LOPA1

pop ebp

pop esi

pop edi

pop edx

pop ecx

pop ebx

pop eax

RET

Tjd endp

end Start

2.menu.rc源文件

#define IDM\_FILE\_EXIT 10001

#define IDM\_FILE\_LIST 10002

#define IDM\_FILE\_CLEAR 10003

#define IDM\_FILE\_RECOMMENDATION 10004

#define IDM\_HELP\_ABOUT 10101

MyMenu MENU

BEGIN

POPUP "&File"

BEGIN

MENUITEM "E&xit", IDM\_FILE\_EXIT

END

POPUP "&Action"

BEGIN

MENUITEM "&Recommendation", IDM\_FILE\_RECOMMENDATION

MENUITEM "C&lear", IDM\_FILE\_CLEAR

MENUITEM "L&ist", IDM\_FILE\_LIST

END

POPUP "&Help"

BEGIN

MENUITEM "&About",IDM\_HELP\_ABOUT

END

END

3.menuID.inc文件

IDM\_FILE\_EXIT equ 10001

IDM\_FILE\_LIST equ 10002

IDM\_FILE\_CLEAR equ 10003

IDM\_FILE\_RECOMMENDATION equ 10004

IDM\_HELP\_ABOUT equ 10101

### 实验步骤

1. 按要求编写源程序。
2. 在cmd中编译和连接并运行程序。
3. 在菜单中点击Help->About，观察并记录结果。
4. 在菜单中点击Action->List，观察并记录结果，然后点击Action->Recommendation后再点击Action->List，观察并记录结果。

### 实验记录与分析

1. 解压MASM32压缩包，在解压后的文件目录下，可以看到在BIN目录下有汇编（ML.exe/RC.exe）和连接（LINK.exe）等程序。
2. 对例子中的demo文件进行编译和连接，编译和连接成功如图3-3所示。

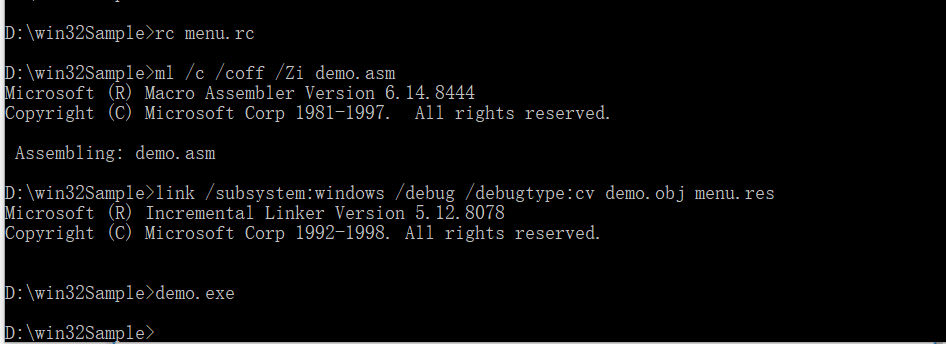


图3-3实验记录

1. TD32与TD16相同之处在于其功能、显示内容基本相同，不同之处在于TD32直接显示32位寄存器的内容，且在保护模式下TD32 中不能查看与该程序段无关的代码段、数据段和堆栈段的内容，且其显示的偏移地址均为32位。16位段程序一般用16位寄存器来进行数据的处理，代码段、数据段以及堆栈段的偏移地址均为16位。而在WIN32程序中的各种偏移地址均为32位，这就为创建更大的程序、存储更多的数据与编写更长的代码段创造了可能。除此以外，WIN32程序还可包括.rc文件，这样一来就可以将代码和菜单、对话框。图标、位图等资源相组合，从而实现各种强大的功能。
2. 在菜单中选择LIST，成功显示商品信息，如图3-4所示。

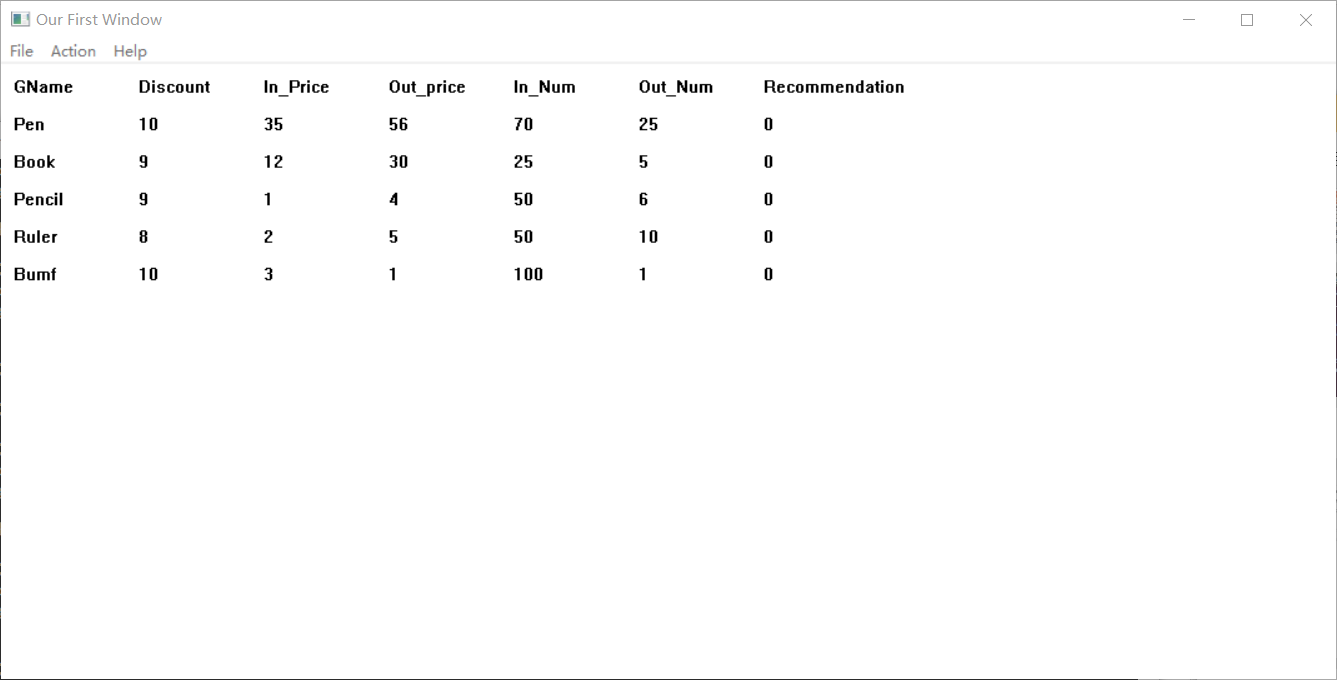


图3-4 实验记录

1. 在菜单中选择RECOMMENDATION，计算商品推荐度，稍等后再次点击LIST，成功显示计算推荐度后的商品信息，如图3-5所示。

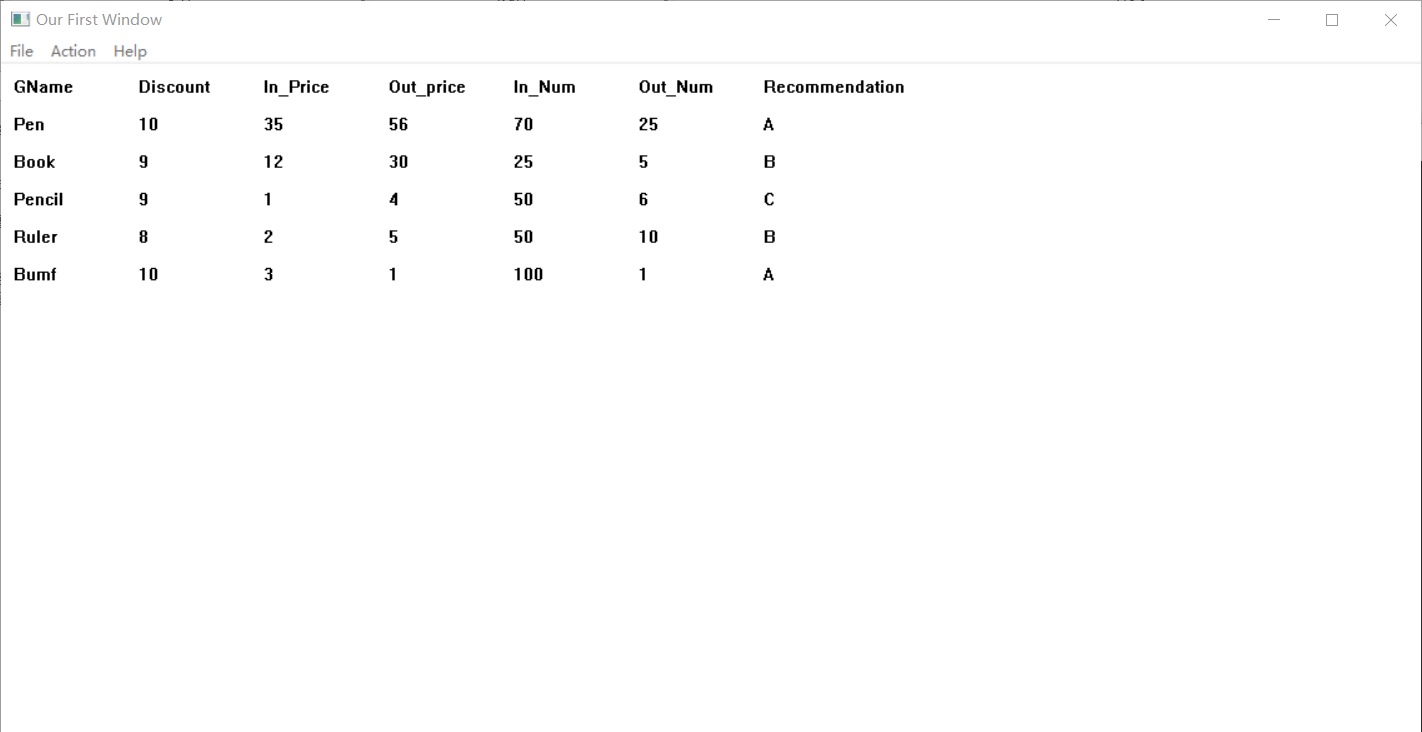


图3-5 实验记录

1. 在菜单中选择About，成功显示含有实验者信息的对话框，如图3-6所示。

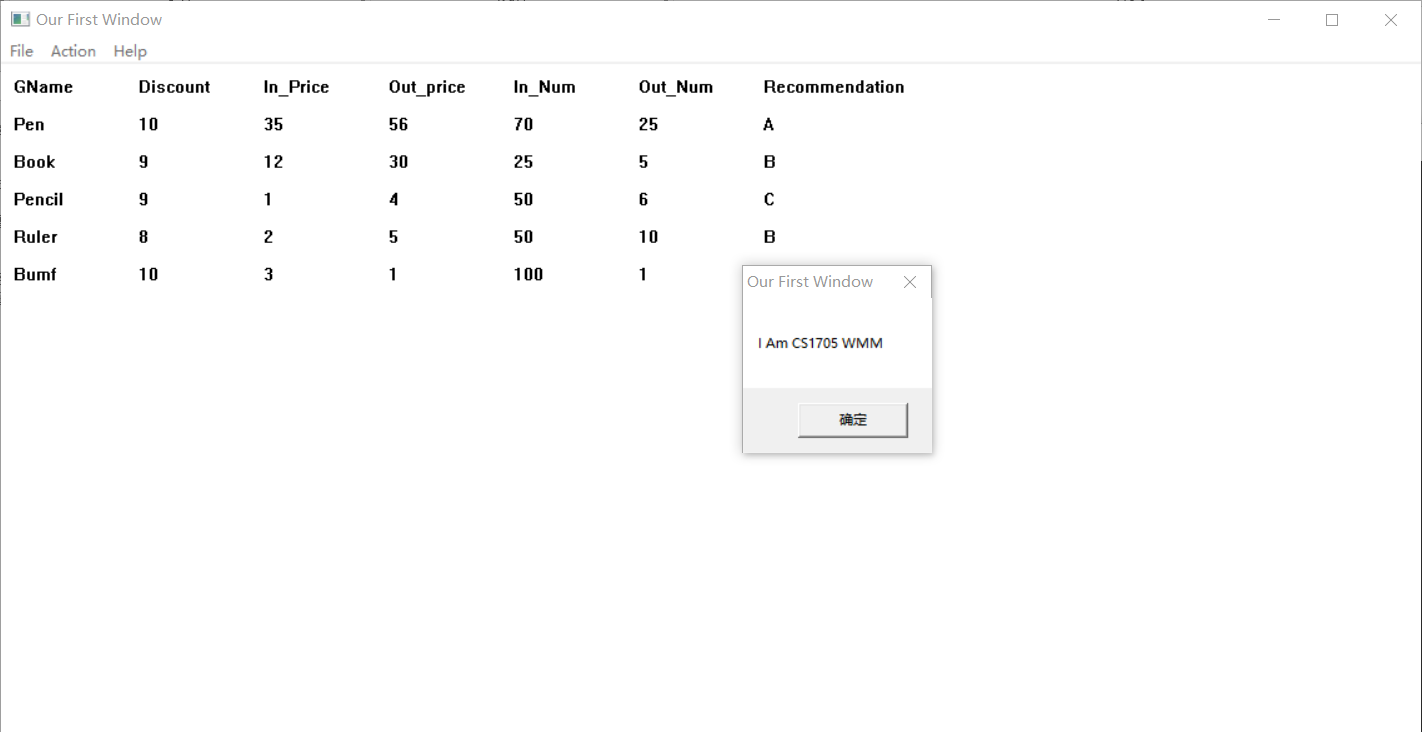


图3-6 实验记录

1. 在DOS系统中，输出是依靠9号功能调用实现的。将需要输出的语句首址传入寄存器DX，再进行9号调用即可输出。而在WIN32中，输出是使用的TEXTOUT函数，其声明如下：

BOOL TextOut(

HDC hdc,//设备描述句柄

int nXStart,//字符串的开始位置x坐标

int nTStart,//字符串的开始位置y坐标

LPCTSTR lpString,//字符串

int cbString//字符串中的字符个数

）；

# 总结与体会

在本次实验中，我成功地在老师给的示例程序的基础上，对该程序进行修改，完成了本次实验任务，对WIN32窗口程序的流程有了初步的掌握。整体来说本次实验内容较少，win32编程也有现成的模板进行参考，总地来说比较简单。通过本次实验，体会到了WIN32编程与DOS下在编译、连接、调试上的差异，以及WIN32编程框架结构中四个模块之间的联系。本次实验最大的不同点就是编写出的程序能实现一个窗口式的交互界面，更加直观。同时对简化段的定义、32位段保护模式下寄存器的使用有了了解。

# 参考文献

[1] 作者.书名.版本(第×版).译者.出版地:出版者,出版年：起页-止页

[序号] URL: 网络地址，如URL:http://www.cbs.dtu.dk/services