目录

设计总说明 ............................................................1

1.课程设计目的 ........................................................1

2.系统分析与设计 ......................................................2

3.系统结构图 ..........................................................6

4.实现过程 ............................................................7

5.代码分析 ............................................................10

* 头文件和规定 ...........................................................**·································10**
* 主函数..............................................................**·································11**
* 加载界面.............................................................**·································11**
* 介绍界面**································**.............................................................**11**
* 主菜单界面**································**............................................................**.12**
* 菜单选项输入1-9处理**································**........................................................**·12**
* 任务结束选项界面..........................................................**································12**
* 任务结束选项输入1-9或a-c容错处理**·····················**.................................................**·13**
* 程序与功能选择**·································**..........................................................**13**
* 九大程序模块**·····························**...........................................................**14**
* 验证码系统**·································**............................................................**21**
* **代码整合·································**..............................................................**22**

6.系统测试出现的问题和解决的方案 ......................................35

7.系统优缺点 ..........................................................35

8.心得体会 ............................................................35

**设计总说明**

用c语言，编写一个程序，程序的进出都由菜单界面完成。要求通过菜单选项调用相应的函数完成9个功能

**1.课程设计目的：**

系统需求描述

1. 介绍程序，按任意键进入主菜单
2. 实现换页功能，加载动画功能，更改背景色与文字色功能，
3. 子程序的进出都由主菜单界面完成。

若输入1-9，要求通过菜单选项调用相应的函数完成如下程序功能：

1. 程序1.输入两个正整数m和n，输出其最大公约数（**用辗转相除法**）。
2. 程序2.（手动输入数组元素）将一个二维数组如a[2][3]的行和列的元素互换，放到另一个二维数组中如b[3][2]。
3. 程序3. （选择自动或者手动输入数组15个元素,当手动输入每错误5次进行模拟验证码的输入）有15个数，按由大到小顺序存放在一个数组中，输入一个数，要求用**折半查找法**找出该数是数组中的第几个元素的值。如果该数不在数组中，则输出“无此数”。
4. 程序1.输入两个正整数，m和n（如100和200），输出m~n间的全部素数（要求用到**sqrt函数**）。（并且输出了多少个素数）
5. 程序4.用**递归方法**求n!（注意定义n的类型，类型的最大值）。
6. 程序5.定义一个数组，并输入10个学生的成绩，用**冒泡法**按由低到高顺序排列输出10个学生的成绩。（优化冒泡法，并输出冒泡法过程及结果）
7. 程序6.定义一个数组，并输入10个学生的成绩，用**选择法**按由低到高的顺序排列输出10个学生的成绩。（优化选择法，输出选择法过程及结果）
8. 程序9.输入两个整数，用**指针变量作为函数的参数**，对输入的两个整数按大小顺序输出。
9. 程序10.定义一个结构体变量（包括年、月、日）。计算该日在本年中是第几天，注意闰年问题。
10. 函数结束后，执行小型选项菜单；输入a-c或1-9通过小型菜单选项调用相应的函数完成：a.显示菜单b.重新任务c.退出程序 1-9.选择任务
11. 主菜单界面若输入的不是1-9，判断出错误，并要求重新输入，每4次输入错误进行模拟验证码的输入

**2.系统分析与设计**

系统总体设计

**该程序包括**

**1）界面显示**

**void desGui();//介绍界面**

**void loadGui();//加载页面**

**void menuGui();//菜单界面**

**void optionGui();//任务结束选项界面**

**//对用户输入选项数据处理**

**void get1();//菜单选项输入1-9**

**void menuOptions();//菜单得到数据后的执行函数**

**void get2();//任务结束选项输入1-9或a-c**

**void Suboptions();//任务结束选项得到数据后的执行函数**

**2)菜单中的9个程序**

**void maxComDivisor();//求最大公约数**

**void counterExchang();//行列交换**

**void Bisearch();//折半查找**

**void primeNum();//素数**

**void Factorial();//求n!**

**void bubbleSort();//冒泡排序**

**void selectSort();//选择排序**

**void PointSort();//用指针变量作参数排序**

**void day();//计算某一日在本年中的第几天**

**3）多次错输入的处理**

**void checkCode();//模拟验证码系统**

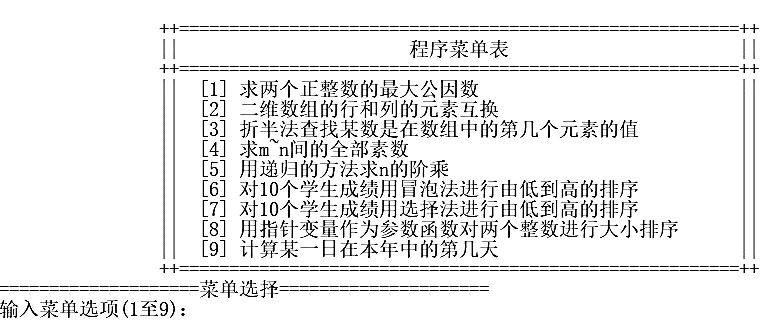
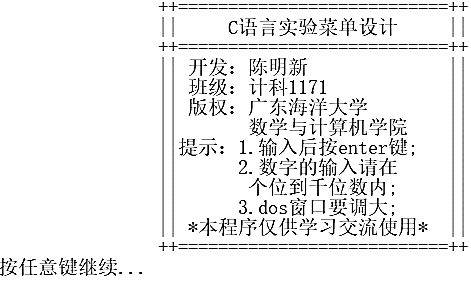
系统详细设计

1. **定义并使用了上述函数，声明和定义**
2. **同时，设计输入选项的变量。**

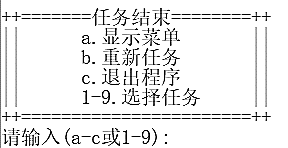
**char opt; //控制选择的变量**

1. **分块设计**

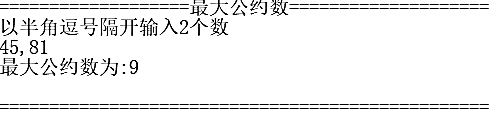
**介绍界面设计,主菜单设计**



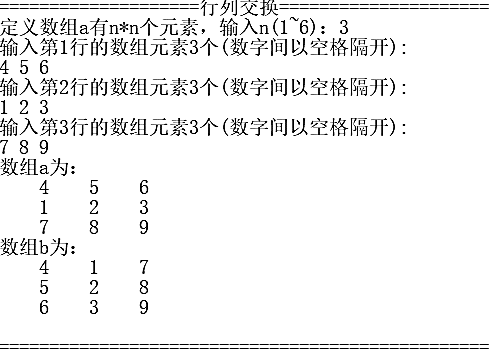
②.任务结束小型选项菜单设计



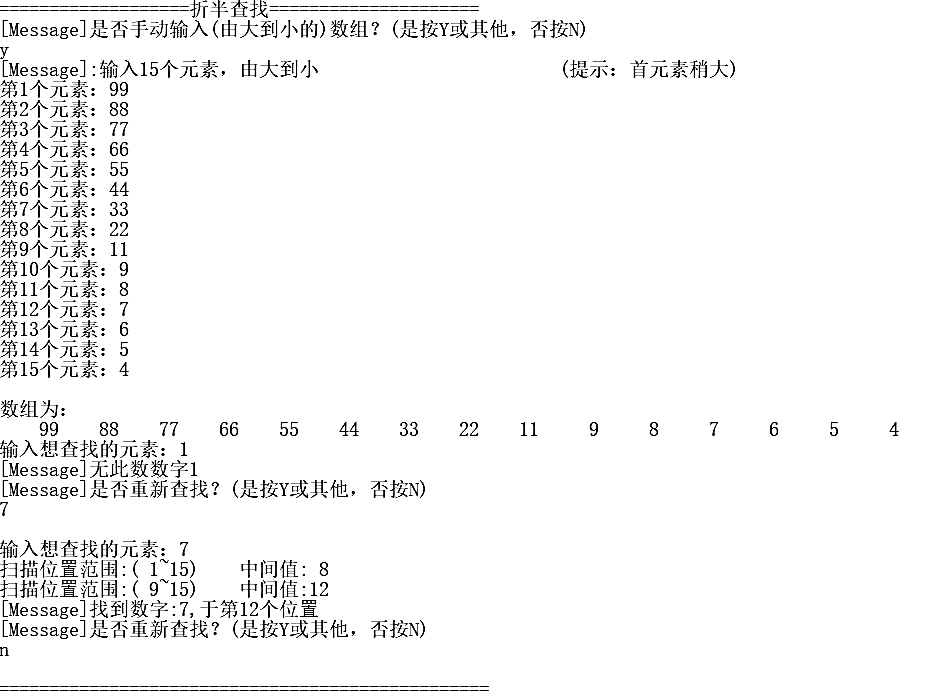
③. 求最大公约数界面



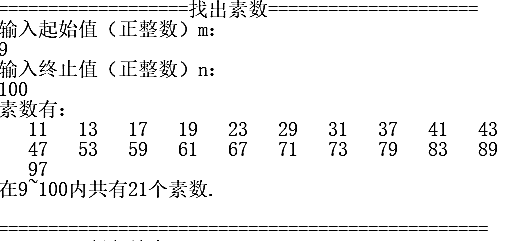
④.行列交换



⑤.折半查找



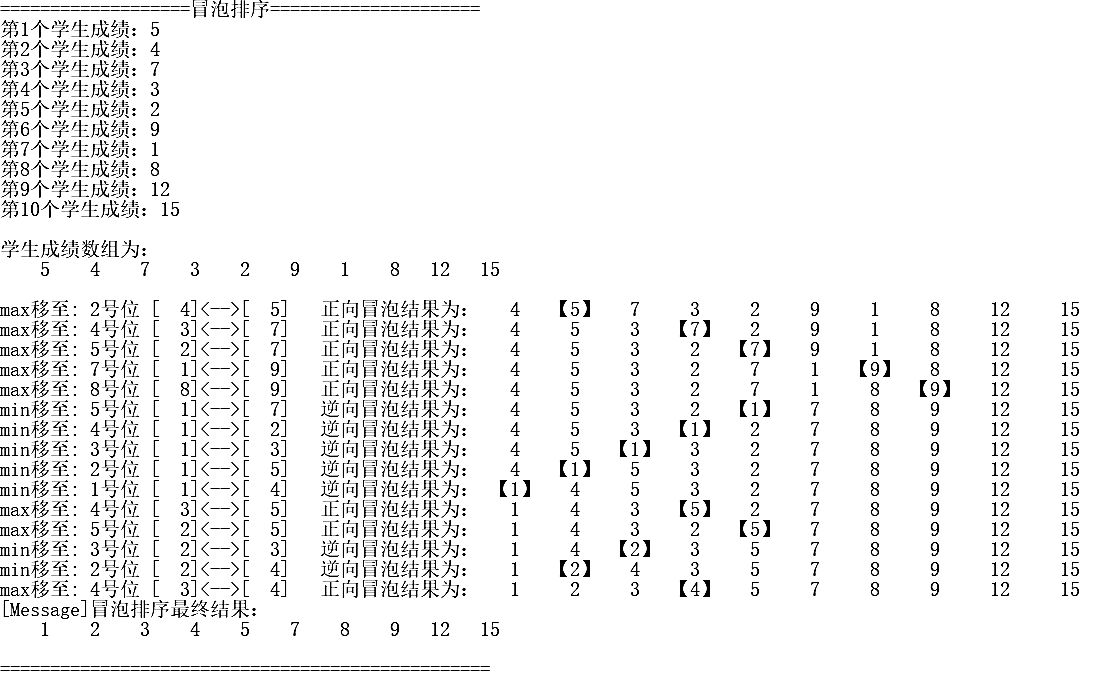
⑥. 挖素数



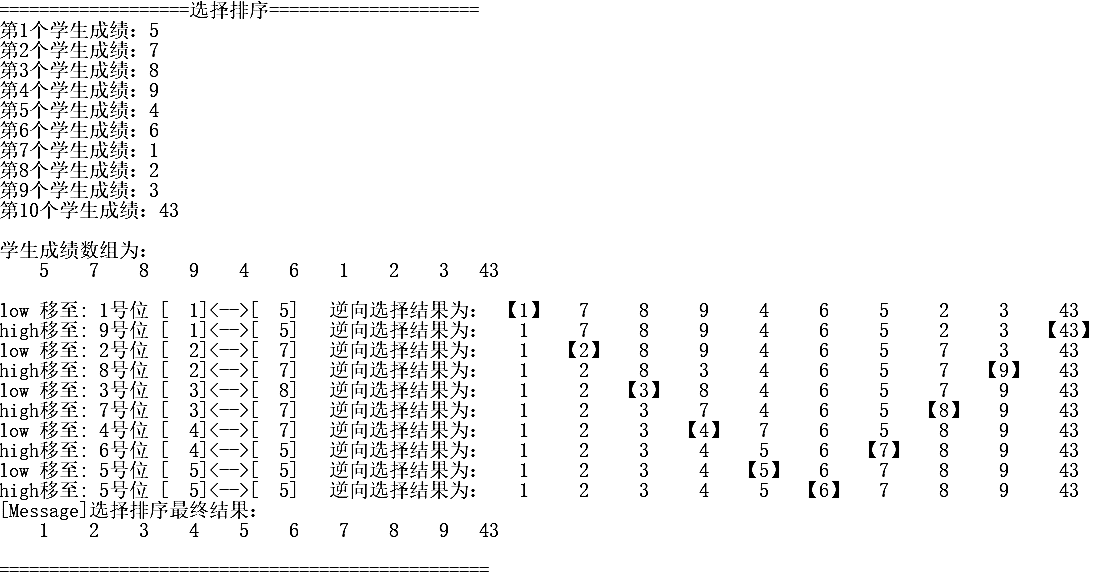
.n的阶乘



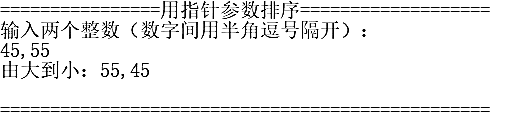
.冒泡排序



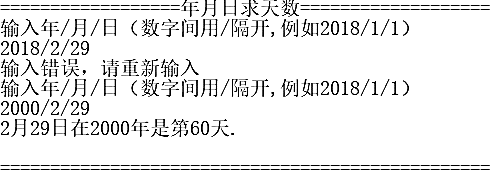
.选择排序



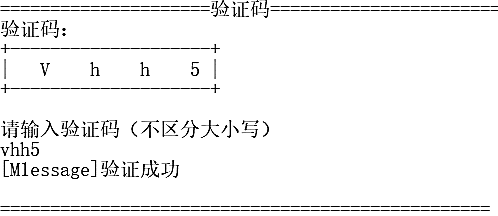
.用指针对两个整数排序



.计算某一日在本年中的第几天



.验证码界面



加载动画



**3.系统结构图：**

程序菜单系统一览

程序菜单系统

介绍

程序主菜单表

求两个正整数的最大公因数

二维数组的行和列的元素互换

折半法查找某数是在数组中的第几个元素的值

求m~n间的全部素数

对10个学生成绩用冒泡法进行由低到高的排序

计算某一日在本年中的第几天

用指针变量作为参数函数对两个整数进行大小排序

对10个学生成绩用选择法进行由低到高的排序

任务结束小型选项菜单

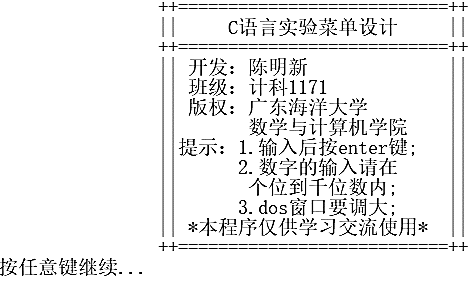
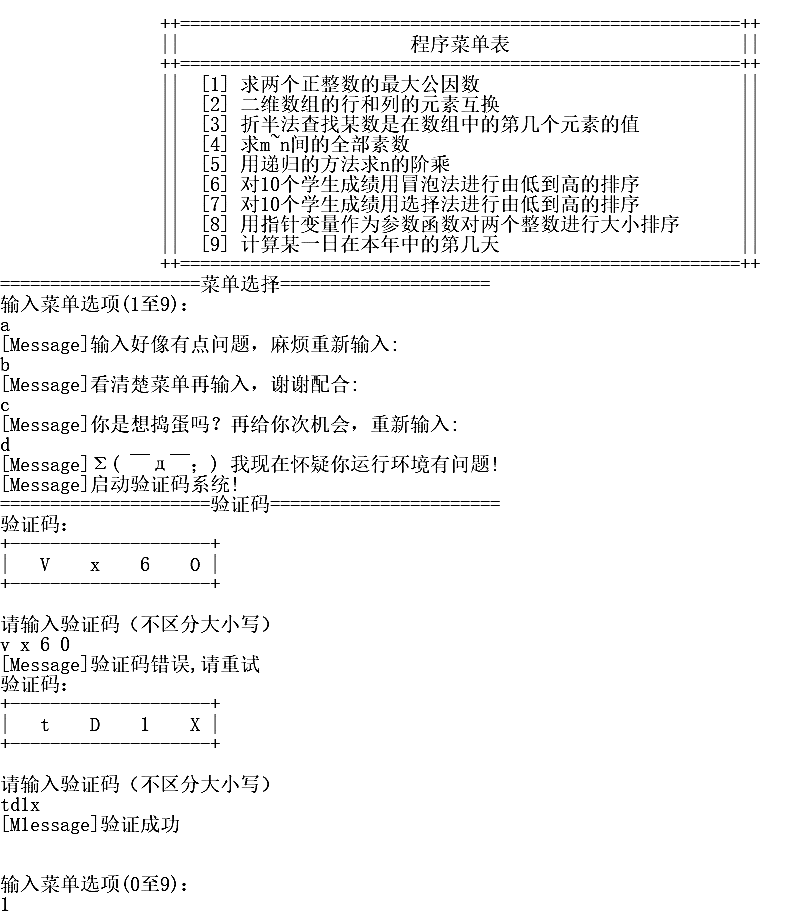
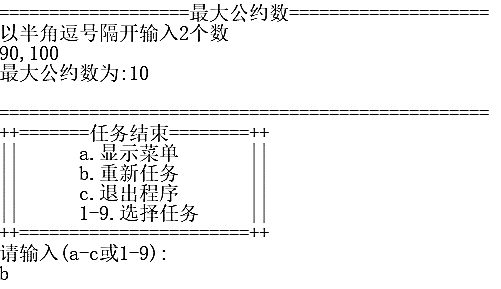
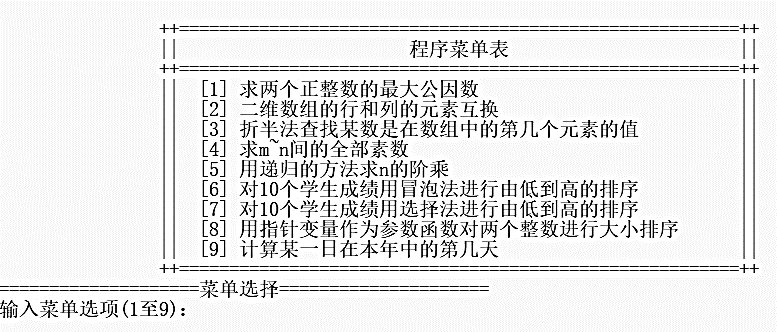
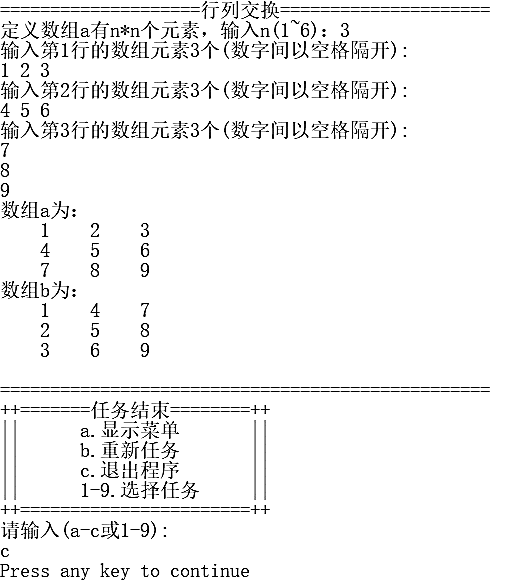
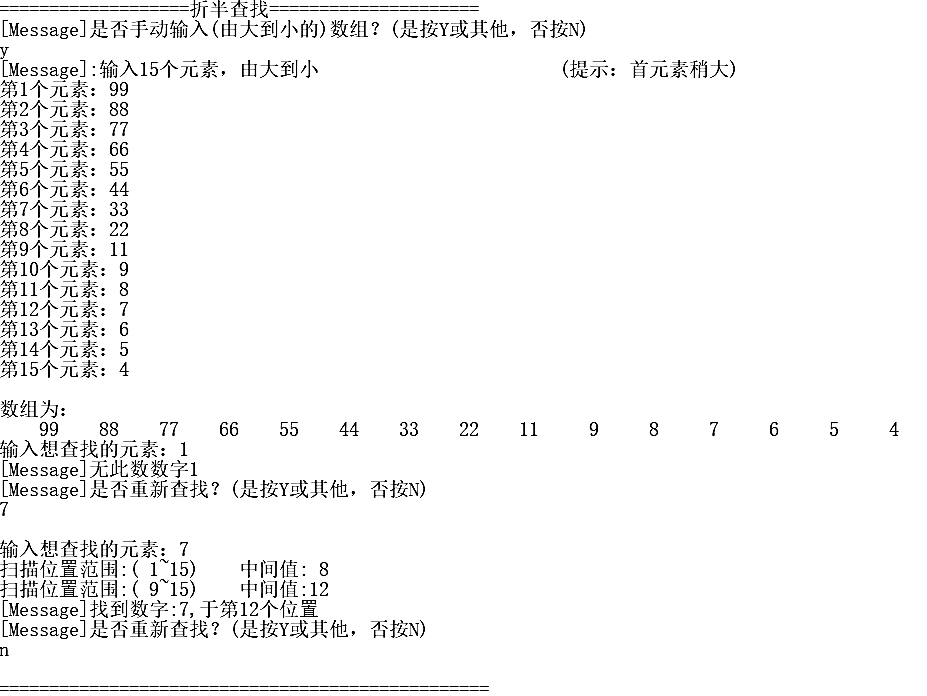
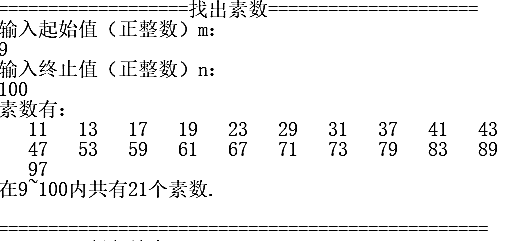
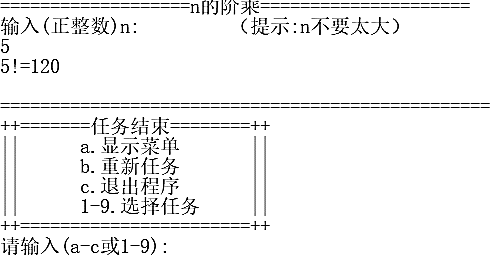
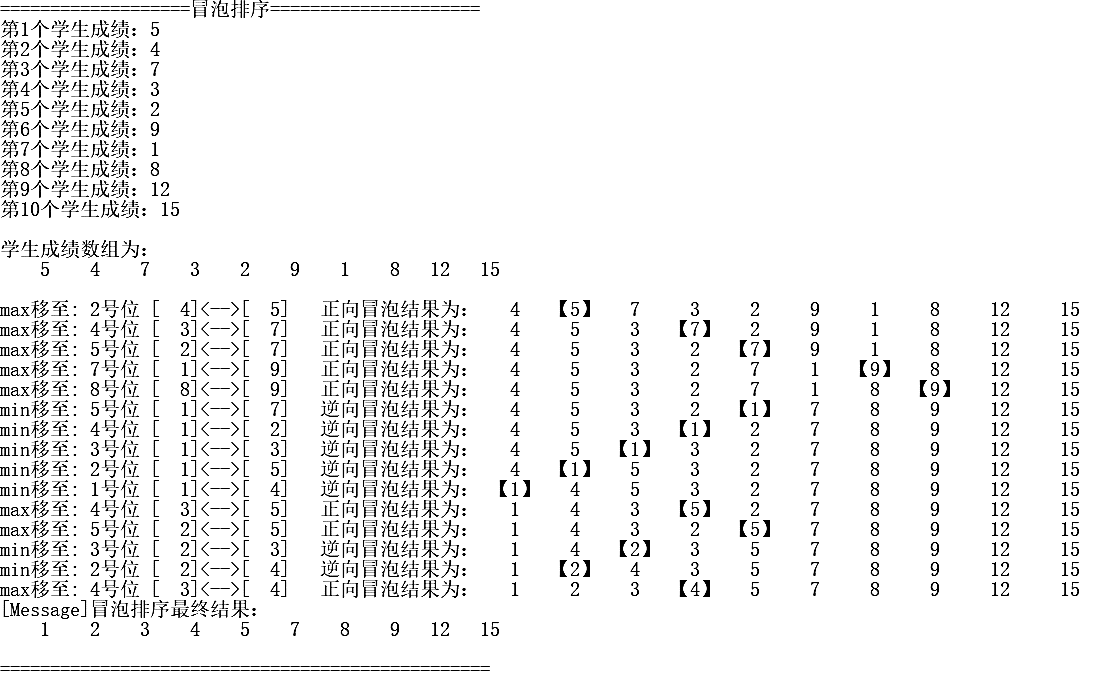
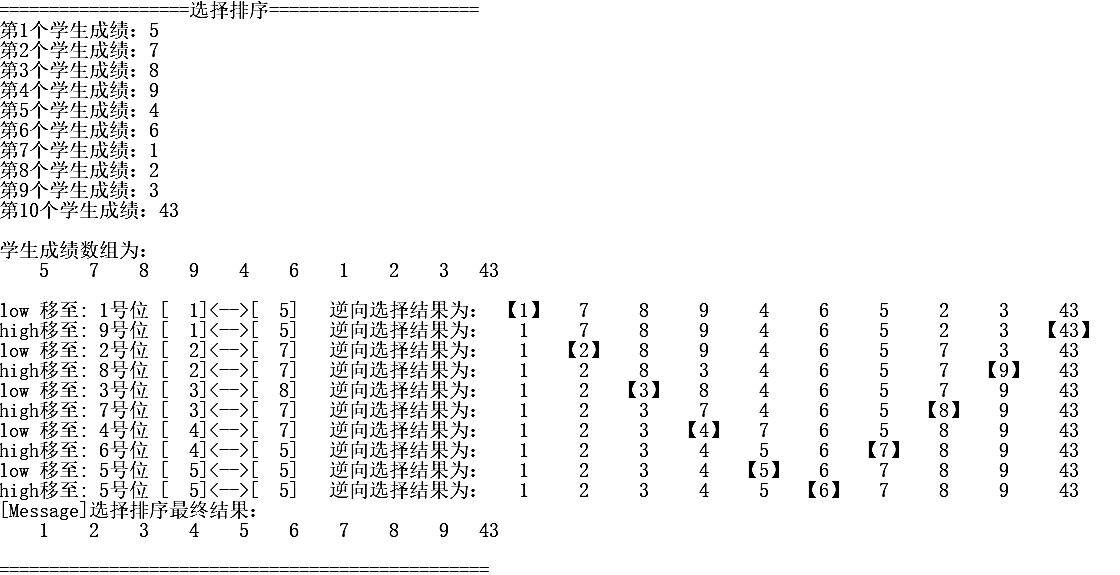
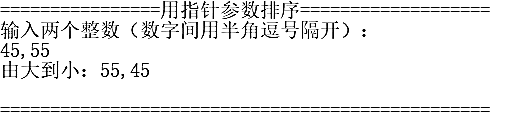
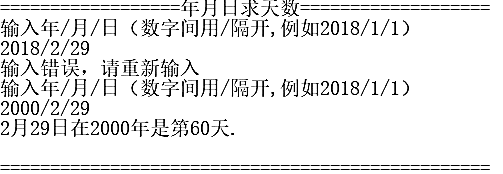
验证码系统

**4. 实现过程：**

**运行测试**

1. 调试



1. 运行
2. 
3. 
4. 
5. 
6. 输入1后
7. 
8. 
9. 任务结束输入a，回到菜单
10. 
11. 输入2，
12. 
13. 
14. 任务结束输入a后退出程序
15. 重启程序，菜单选择中输入2，再回到该界面，输入b，又重新执行了行列交换
16. 接着输入3
17. 
18. 任务结束输入4
19. 
20. 任务结束输入5
21. 
22. 任务结束输入6
23. 
24. 任务结束输入7
25. 
26. 任务结束输入8
27. 
28. 任务结束输入9
29. 

**5.代码分析：**

**需要头文件和规定**

|  |
| --- |
| **#include<stdio.h>**  **#include<math.h>**  **#include <conio.h>**  **#include<windows.h>**  **#include<time.h>**  **#define N 15//数组容量**  **#define M 10//数组容量** |

**执行主函数**

|  |
| --- |
| **//主函数**  **int main()**  **{loadGui();//加载界面**  **system("color f2");//dos窗口输入命令：color f2；f是背景色：亮白；2是文字色：浅绿**  **system("title 程序菜单1.0.0 Beta版");//修改dos窗口名**  **desGui();//介绍界面**  **printf("按任意键继续...");**  **getch();//接受一个任意键的输入，不用按回车就返回，用到头文件conio.h**  **loadGui();//加载界面**  **Sleep(100);//延迟执行语句100毫秒**  **menuGui();//主菜单界面**  **return 0;**  **}** |

**写个加载界面**

|  |
| --- |
| **//加载页面**  **void loadGui()**  **{int flag=3;**  **while(flag)**  **{flag--;system("cls");printf("正在加载：..........");**  **printf("\n█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▄▄▃▂▁ ");system("cls");printf("正在加载：......");**  **printf("\n▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂ ");system("cls");printf("正在加载：...");**  **printf("\n▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▄▄▃▂▁▂▃ ");system("cls");printf("正在加载：...");**  **printf("\n▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▄▄▃▂▁▂▃▄ ");system("cls");printf("正在加载：......");**  **printf("\n▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▄▄▃▂▁▂▃▄▅ ");system("cls"); ;**  **}**  **printf("加载成功！\n");**  **}** |

**写个介绍界面**

|  |
| --- |
| **//介绍界面**  **void desGui()**  **{**  **printf("\t\t++===========================++\n");**  **printf("\t\t|| C语言实验菜单设计 ||\n");Sleep(50);**  **printf("\t\t++===========================++\n");Sleep(50);**  **printf("\t\t|| 开发：陈明新 ||\n");Sleep(50);**  **printf("\t\t|| 班级：计科1171 ||\n");Sleep(50);**  **printf("\t\t|| 版权：广东海洋大学 ||\n");Sleep(50);**  **printf("\t\t|| 数学与计算机学院 ||\n");Sleep(50);**  **printf("\t\t||提示：1.输入后按enter键; ||\n");Sleep(50);**  **printf("\t\t|| 2.数字的输入请在 ||\n");Sleep(50);**  **printf("\t\t|| 个位到千位数内; ||\n");Sleep(50);**  **printf("\t\t|| 3.dos窗口要调大; ||\n");Sleep(50);**  **printf("\t\t|| \*本程序仅供学习交流使用\* ||\n");Sleep(50);**  **printf("\t\t++===========================++\n");Sleep(50);**  **}** |

**写个主菜单界面**

|  |
| --- |
| **//菜单界面**  **void menuGui()**  **{system("cls");//换页**  **printf("\n");**  **printf("\t\t++========================================================++\n");**  **printf("\t\t|| 程序菜单表 ||\n");Sleep(100);**  **printf("\t\t++========================================================++\n");Sleep(100);**  **printf("\t\t|| [1] 求两个正整数的最大公因数 ||\n");Sleep(100);**  **printf("\t\t|| [2] 二维数组的行和列的元素互换 ||\n");Sleep(100);**  **printf("\t\t|| [3] 折半法查找某数是在数组中的第几个元素的值 ||\n");Sleep(100);**  **printf("\t\t|| [4] 求m~n间的全部素数 ||\n");Sleep(100);**  **printf("\t\t|| [5] 用递归的方法求n的阶乘 ||\n");Sleep(100);**  **printf("\t\t|| [6] 对10个学生成绩用冒泡法进行由低到高的排序 ||\n");Sleep(100);**  **printf("\t\t|| [7] 对10个学生成绩用选择法进行由低到高的排序 ||\n");Sleep(100);**  **printf("\t\t|| [8] 用指针变量作为参数函数对两个整数进行大小排序 ||\n");Sleep(100);**  **printf("\t\t|| [9] 计算某一日在本年中的第几天 ||\n");Sleep(100);**  **printf("\t\t++========================================================++\n");Sleep(100);**  **get1();**  **}** |

**菜单选项输入1-9容错处理**

|  |
| --- |
| **//菜单选项输入1-9**  **void get1()**  **{int flag=1,bottle;//flag判断输入对错，bottle判断输入缓冲区的第一个字符**  **printf("====================菜单选择=====================\n");**  **printf("输入菜单选项(1至9)：\n");**  **while(flag)**  **{**  **scanf("%c",&opt);**  **bottle=(getchar()!='\n');**  **if(opt>'0'&&opt<='9'&&bottle==0)**  **{menuOptions();**  **flag=0;}**  **else**  **{if(bottle!=0)**  **{while(getchar()!='\n');}//消除输入缓冲区**  **switch(flag)**  **{case 1:printf("[Message]输入好像有点问题，麻烦重新输入:\n");flag++;break;**  **case 2:printf("[Message]看清楚菜单再输入，谢谢配合:\n");flag++;break;**  **case 3:printf("[Message]你是想捣蛋吗？再给你次机会，重新输入:\n");flag++;break;**  **case 4:printf("[Message]Σ( ￣д￣；) 我现在怀疑你运行环境有问题!\n[Message]启动验证码系统!\n");checkCode();flag=1;printf("\n\n输入菜单选项(0至9)：\n");break;**  **}**    **}**  **}**  **}** |

**写任务结束选项界面**

|  |
| --- |
| **//任务结束选项界面**  **void optionGui()**  **{printf("\n=================================================\n");**  **Sleep(500);**  **printf("++=======任务结束========++\n");Sleep(100);**  **printf("|| a.显示菜单 ||\n");Sleep(100);**  **printf("|| b.重新任务 ||\n");Sleep(100);**  **printf("|| c.退出程序 ||\n");Sleep(100);**  **printf("|| 1-9.选择任务 ||\n");Sleep(100);**  **printf("++=======================++\n");Sleep(100);**  **printf("请输入(a-c或1-9):\n");**  **get2();**  **}** |

**任务结束选项输入1-9或a-c容错处理**

|  |
| --- |
| **//任务结束选项输入1-9或a-c**  **void get2()**  **{int flag=1,bottle;**  **char last=opt;//记下上一次的opt选择，用于重新任务**  **while(flag)**  **{**  **scanf("%c",&opt);**  **bottle=(getchar()!='\n');**  **if(opt>'0'&&opt<='9'&&bottle==0)**  **{menuOptions();**  **flag=0;}**  **else if(opt>='a'&&opt<='c'&&bottle==0)**  **{if(opt=='b')**  **{opt=last;}**  **Suboptions();**  **flag=0;}**  **else {if(bottle!=0)**  **{while(getchar()!='\n');}**  **switch(flag)**  **{case 1:printf("[Message]输入好像有点问题，麻烦重新输入:\n");flag++;break;**  **case 2:printf("[Message]看清楚菜单再输入，谢谢配合:\n");flag++;break;**  **case 3:printf("[Message]你是想捣蛋吗？再给你次机会，重新输入:\n");flag++;break;**  **case 4:printf("[Message]Σ( ￣д￣；) 我现在怀疑你运行环境有问题!\n[Message]启动验证码系统!\n");**  **checkCode();flag=1;optionGui();break;**  **}**    **}**  **}**  **menuOptions();**  **}** |

**程序与功能选择**

|  |
| --- |
| **void menuOptions()**  **{switch(opt)**  **{case '1':maxComDivisor();break;**  **case '2':counterExchang();break;**  **case '3':Bisearch();break;**  **case '4':primeNum();;break;**  **case '5':Factorial();break;**  **case '6':bubbleSort();break;**  **case '7':selectSort();break;**  **case '8':PointSort();break;**  **case '9':day();break;**  **}**  **optionGui();**  **}**  **void Suboptions()**  **{switch(opt)**  **{case 'a':menuGui();break;**  **case 'c':exit(0);break;**  **default:menuOptions();break;**  **}**  **optionGui();**  **}** |

**九大程序模块**

|  |
| --- |
| **//求最大公约数**  **void maxComDivisor()**  **{int u,v,r,temp;//u大的数,v小的数,r是余数，temp中转站**  **loadGui();system("cls");**  **printf("===================最大公约数====================\n");Sleep(100);**  **printf("以半角逗号隔开输入2个数\n");**  **scanf("%d,%d",&u,&v);**  **while(!(u>0&&v>0))**  **{while(getchar()!='\n');**  **printf("[Message]输入错误，请重新输入:\n");**  **scanf("%d,%d",&u,&v);**  **}**  **while(getchar()!='\n');**  **if(v>u)**  **{temp=v;v=u;u=temp;}**  **while((r=u%v)!=0)**  **{u=v;v=r;}**  **printf("最大公约数为:%d\n",v);**  **}**  **//行列交换**  **void counterExchang()**  **{int a[M][M],b[M][M],i,j,n,temp;**  **loadGui();system("cls");**  **printf("====================行列交换=====================\n");Sleep(100);**  **printf("定义数组a有n\*n个元素，输入n(1~6)：");**  **scanf("%d",&n);**  **while(!(n>=1&&n<=6))**  **{while(getchar()!='\n');**  **printf("[Message]输入错误，请重新输入:\n");**  **scanf("%d",&n);**  **}**  **while(getchar()!='\n');**  **for(i=0;i<n;i++)**  **{printf("输入第%d行的数组元素%d个(数字间以空格隔开):\n",i+1,n);**  **for(j=0;j<n;j++)**  **{**  **scanf("%d",&a[i][j]);**  **}**  **while(getchar()!='\n');**  **}**  **printf("数组a为：\n");**  **for(i=0;i<n;i++)**  **{for(j=0;j<n;j++)**  **{printf("%5d",a[i][j]);**  **}**  **printf("\n");**  **}**  **for(i=0;i<n;i++)**  **{for(j=i;j<n;j++)**  **{temp=a[i][j];**  **a[i][j]=a[j][i];**  **a[j][i]=temp;**  **}**  **}**  **for(i=0;i<n;i++)**  **{for(j=0;j<n;j++)**  **{b[i][j]=a[i][j];**  **}**  **}**  **printf("数组b为：\n");**  **for(i=0;i<n;i++)**  **{for(j=0;j<n;j++)**  **{printf("%5d",b[i][j]);**  **}**  **printf("\n");**  **}**  **}**  **//折半查找数组中数字**  **void Bisearch()**  **{int a[N]={105,45,30,15,14,13,12,10,9,8,5,4,3,2,1},i,num,flag=1,top,low,mid,find;**  **char c;**  **//i下标,num查找的数,flag判断输入内容,top查找范围最大值,low查找范围最小值,mid中间值,find判断是否找到，c选择是否继续**  **loadGui();system("cls");**  **printf("===================折半查找=====================\n");Sleep(100);**  **printf("[Message]是否手动输入(由大到小的)数组？(是按Y或其他，否按N)\n");**  **c=getchar();**  **while(getchar()!='\n');**  **if(c!='N'&&c!='n')**  **{printf("[Message]:输入15个元素，由大到小 (提示：首元素稍大)\n");**  **for(i=0;i<N;i++)**  **{printf("第%d个元素：",i+1);**  **scanf("%d",&a[i]);**  **while(getchar()!='\n');**  **if(i>0&&a[i]>a[i-1])**  **{printf("[Message]输入错误，请重新输入: (提示：由大到小)\n");**  **i--;flag++;**  **if(flag>5)**  **{checkCode();flag=1;}**  **}**  **}**  **}**  **printf("\n数组为：\n");**  **for(i=0;i<N;i++)**  **{printf("%6d",a[i]);}**  **Sleep(500);**  **flag=1;**  **while(flag)**  **{top=N-1;low=0;mid=0;find=0;**  **printf("\n输入想查找的元素：");**  **scanf("%d",&num);while(getchar()!='\n');**  **if(num<=a[0]&&num>=a[N-1])**  **{while((find==0)&&(low<=top))**  **{**  **mid=(top+low)/2;**  **printf("扫描位置范围:(%2d~%2d)\t中间值:%2d\n",low+1,top+1,mid+1);Sleep(300);//扫描查找范围输出**  **if(num==a[mid])**  **{find=1;}//表示找到了**  **else if(num>a[mid])//缩小扫描区间**  **{top=mid-1;}**  **else {low=mid+1;}**  **}**  **}**  **if(find==1)**  **{printf("[Message]找到数字:%d,于第%d个位置\n",num,mid+1);**  **}**  **else {printf("[Message]无此数数字%d\n",num);}**  **printf("[Message]是否重新查找？(是按Y或其他，否按N)\n");**  **c=getchar();while(getchar()!='\n');**  **if(c=='N'||c=='n')**  **flag=0;**  **}**  **}**  **//找素数**  **void primeNum()**  **{int m=0,n=0,i,num,count=0;**  **//m,n是搜索范围，i数组下标，num是范围内数字，count控制素数的数量**  **loadGui();system("cls");**  **printf("===================找出素数=====================\n");Sleep(100);**  **while(1)**  **{printf("输入起始值（正整数）m：\n");**  **scanf("%d",&m);**  **while(getchar()!='\n');**  **printf("输入终止值（正整数）n：\n");**  **scanf("%d",&n);**  **while(getchar()!='\n');**  **if(m>=n||m<=0)**  **{printf("输入错误，请重新输入\n");}**  **else break;**  **}**  **printf("素数有：\n");**  **for(num=m;num<=n;num++)**  **{for(i=2;i<=(int)sqrt(num);i++)**  **{if(num%i==0)//判断数素**  **{break;}**  **}**  **if(i>=(int)sqrt(num)+1&&num!=1)//判断并输出素数**  **{printf("%5d",num);**  **count++;**  **if(count%10==0&&count>=10)**  **{printf("\n");}**  **}**  **}**  **printf("\n在%d~%d内共有%d个素数.\n",m,n,count);**  **}**  **//算n!**  **void Factorial()**  **{int n,resu;**  **int FactorialFunction(int);**  **loadGui();system("cls");**  **printf("===================n的阶乘=====================\n");Sleep(100);**  **printf("输入(正整数)n: （提示:n不要太大）\n");**  **scanf("%d",&n);**  **while(getchar()!='\n');**  **while(n<=0)**  **{**  **printf("[Message]输入错误，请重新输入:\n");**  **scanf("%d",&n);**  **while(getchar()!='\n');**  **}**  **resu=FactorialFunction(n);**  **printf("%d!=%d\n",n,resu);**  **}**  **//n!功能模块**  **int FactorialFunction(int n)**  **{int f;**  **if(n==1||n==0)**  **f=1;**  **else**  **f=FactorialFunction(n-1)\*n;//递归**  **return f;**  **}**  **//对成绩用冒泡法进行由低到高的排序**  **/\*冒泡法优化，**  **优化1.添加标志位，记录最后一次交换位置的地方，**  **证明最后一次交换位置之后的地方时是排好序的，**  **下一次只需要排最后一次之前（或之后）的地方扫描就好了,**  **优化2.双向冒泡，每趟排序遍历一次找到一个最大值和者最小值，**  **其实都就是最后一次交换位置的地方，再从最大值和者最小值之间扫描排序，**  **使减少了一半的排序趟数 \*/**  **void bubbleSort()**  **{int a[M],i,j,k,temp,posit=0,high=M-1,low=0;**  **//i,j,k数组下标，temp中转站，poist是记下交换位置，high是扫描范围最大值，low是扫描范围最小值**  **loadGui();system("cls");**  **printf("===================冒泡排序=====================\n");Sleep(100);**  **for(i=0;i<M;i++)**  **{printf("第%d个学生成绩：",i+1);**  **scanf("%d",&a[i]);**  **while(getchar()!='\n');**  **}**  **printf("\n学生成绩数组为：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **while(low<high)**  **{for(i=low;i<high;i++)//正向冒泡**  **{if(a[i]>a[i+1])**  **{temp=a[i];**  **a[i]=a[i+1];**  **a[i+1]=temp;**  **posit=i;**  **printf("\nmax移至:%2d号位 [%3d]<-->[%3d] ",i+2,a[i],a[i+1]);printf("正向冒泡结果为：");**  **for(k=0;k<10;k++)//输出每趟排序结果**  **{**  **if(k==i+1) printf(" 【%d】",a[k]);**  **else printf(" %d ",a[k]);**  **}**  **Sleep(200);**  **}**  **}**  **high=posit;**  **for(j=high;j>low;j--)//逆向冒泡**  **{if(a[j]<a[j-1])**  **{temp=a[j];**  **a[j]=a[j-1];**  **a[j-1]=temp;**  **posit=j;**  **printf("\nmin移至:%2d号位 [%3d]<-->[%3d] ",j,a[j-1],a[j]);printf("逆向冒泡结果为：");**  **for(k=0;k<10;k++)//输出每趟排序结果**  **{**  **if(k==j-1) printf(" 【%d】",a[k]);**  **else printf(" %d ",a[k]);**  **}**  **Sleep(200);**  **}**  **}**  **low=posit;**  **}**  **printf("\n[Message]冒泡排序最终结果：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **}**  **/\*舍弃的代码：选择法优化算法一**  **void selectSort()**  **{int a[M],i,j,k,temp,posit=0,high=M-1,low=0;**  **loadGui();system("cls");**  **printf("===================选择排序=====================\n");Sleep(100);**  **for(i=0;i<M;i++)**  **{printf("第%d个学生成绩：",i+1);**  **scanf("%d",&a[i]);**  **while(getchar()!='\n');**  **}**  **printf("\n学生成绩数组为：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **for(i=0;i<M/2;i++)**  **{**  **for(j=low;j<M-i;j++)**  **{if(a[i]>a[j])**  **{temp=a[j];**  **a[j]=a[i];**  **a[i]=temp;**  **low=j;posit=1;**  **}**  **}**  **if(a[low]!=a[i+1]&&posit==1)**  **{temp=a[low];**  **a[low]=a[i+1];**  **a[i+1]=temp;**  **posit=0;**  **}**  **else low=i+1;**  **for(j=high;j>i;j--)**  **{if(a[M-i-1]<a[j])**  **{temp=a[M-i-1];**  **a[M-i-1]=a[j];**  **a[j]=temp;**  **temp=high;**  **high=j;posit=1;**  **}**  **}**  **if(a[high]!=a[M-i-2]&&posit==1)**  **{temp=a[high];**  **a[high]=a[M-i-2];**  **a[M-i-2]=temp;**  **posit=0;**  **}**  **else high=M-i-2;**  **}**  **printf("\n[Message]选择排序最终结果：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **}\*/**  **//对成绩用选择法进行由低到高的排序**  **/\*选择法优化：1.添加标志位，记录最后一次交换位置的地方；**  **2.双向选择法\*/**  **void selectSort()**  **{int a[M],i,j,k,temp,high,low;**  **//i,j,k数组下标，temp中转站，high是扫描范围最大值，low是扫描范围最小值**  **loadGui();system("cls");**  **printf("===================选择排序=====================\n");Sleep(100);**  **for(i=0;i<M;i++)**  **{printf("第%d个学生成绩：",i+1);**  **scanf("%d",&a[i]);**  **while(getchar()!='\n');**  **}**  **printf("\n学生成绩数组为：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **for(i=0;i<M/2;i++)**  **{high=M-i-1;low=i;**  **//正向选择**  **for(j=low;j<M-i;j++)**  **{if(a[low]>a[j])**  **{**  **low=j;//记下要交换的位置**  **}**  **}**  **temp=a[low];a[low]=a[i];a[i]=temp;**  **printf("\nlow 移至:%2d号位 [%3d]<-->[%3d] ",i+1,a[i],a[low]);printf("逆向选择结果为：");**  **for(k=0;k<10;k++)//输出每趟排序结果**  **{**  **if(k==i) printf(" 【%d】",a[k]);**  **else printf(" %d ",a[k]);**  **}**  **Sleep(200);**  **//逆向选择**  **for(j=high;j>i;j--)**  **{if(a[high]<a[j])**  **{**  **high=j;//记下要交换的位置**  **}**  **}**  **temp=a[high];a[high]=a[M-i-1];a[M-i-1]=temp;**  **printf("\nhigh移至:%2d号位 [%3d]<-->[%3d] ",M-i-1,a[i],a[low]);printf("逆向选择结果为：");**  **for(k=0;k<10;k++)//输出每趟排序结果**  **{**  **if(k==M-i-1) printf(" 【%d】",a[k]);**  **else printf(" %d ",a[k]);**  **}**  **Sleep(200);**  **}**  **printf("\n[Message]选择排序最终结果：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **}**  **//用指针变量作参数对两个整数排序**  **void PointSort()**  **{int \*p,\*q;**  **int a,b;**  **void maxSort(int \*p,int \*q);**  **loadGui();system("cls");**  **printf("================用指针参数排序===================\n");Sleep(100);**  **printf("输入两个整数（数字间用半角逗号隔开）：\n");**  **scanf("%d,%d",&a,&b);**  **while(getchar()!='\n');//消除输入缓冲区**  **p=&a;q=&b;**  **maxSort(p,q);**  **printf("由大到小：%d,%d\n",a,b);**  **}**  **//整数排序功能模块**  **void maxSort(int \*p,int \*q)**  **{int temp;**  **if(\*p<\*q)**  **{temp=\*p;**  **\*p=\*q;**  **\*q=temp;**  **}**  **}**  **//年月日结构体**  **struct ymd**  **{int year;**  **int mon;**  **int day;**  **};**  **//用指针变量作参数排序**  **void day()**  **{struct ymd date;int sum=0,flag=1,leap=0;//sum求得第几天，flag判断输入对错，leap判断闰年**  **loadGui();system("cls");**  **printf("==================年月日求天数===================\n");Sleep(100);**  **while(flag)**  **{**  **printf("输入年/月/日（数字间用/隔开,例如2018/1/1）\n");**  **scanf("%d/%d/%d",&date.year,&date.mon,&date.day);**  **while(getchar()!='\n');//消除输入缓冲区**  **if((date.year%4==0&&date.year%100!=0)||date.year%400==0)//判断闰年**  **{leap=1;**  **}**  **switch(date.mon)**  **{case 1:if(date.day<=31)sum=date.day;break;**  **case 2:if((date.day<=29&&leap==1)||(date.day<29))sum=date.day+31;else flag++;break;**  **case 3:if(date.day<=31)sum=date.day+59;else flag++;break;**  **case 4:if(date.day<31)sum=date.day+99;else flag++;break;**  **case 5:if(date.day<=31)sum=date.day+120;else flag++;break;**  **case 6:if(date.day<31)sum=date.day+151;else flag++;break;**  **case 7:if(date.day<=31)sum=date.day+181;else flag++;break;**  **case 8:if(date.day<=31)sum=date.day+212;else flag++;break;**  **case 9:if(date.day<31)sum=date.day+243;else flag++;break;**  **case 10:if(date.day<=31)sum=date.day+273;else flag++;break;**  **case 11:if(date.day<31)sum=date.day+304;else flag++;break;**  **case 12:if(date.day<=31)sum=date.day+334;else flag++;break;**  **default:flag++;break;**  **}**  **flag--;**  **if(flag) printf("输入错误，请重新输入\n");**  **if(date.mon>=3&&leap==1) sum++;**  **}**  **printf("%d月%d日在%d年是第%d天.\n",date.mon,date.day,date.year,sum);**  **}** |

**验证码系统**

|  |
| --- |
| **//模拟验证码**  **void checkCode()**  **{int i,bottle,flag=1;**  **char allCode[] = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJLMNOPQRSTUVWXYZ";//所有验证码**  **char getCode[5],writeCode[5];//getCode[5]是获得随机产生的字符，writeCode[5]是写入随机产生的字符**  **printf("=====================验证码=======================\n");Sleep(100);**  **while(flag)**  **{srand((unsigned)time(NULL));//取随机数种子，指定种子值为时间**  **printf("验证码：\n");**  **printf("+--------------------+\n");**  **printf("|");**  **for (i=0; i<4; i++)**  **{getCode[i]=allCode[rand()%(int)strlen(allCode)];//随机数在[0, allCode的有效长度值)中产生，并且产生的随机数作为allCode数组下标**  **printf("%4c ",getCode[i]);**  **}**  **getCode[i]='\0';**  **printf("|\n");**  **printf("+--------------------+\n");**  **printf("\n请输入验证码（不区分大小写）\n");**  **for (i=0; i<4; i++)**  **{**  **scanf("%c",&writeCode[i]);**  **}**  **bottle=(getchar()!='\n');//判断输入缓冲区是否溢出**  **writeCode[i]='\0';**  **strlwr(getCode);**  **strlwr(writeCode);**  **if(strcmp(getCode,writeCode)==0&&bottle==0)**  **{printf("[M1essage]验证成功\n");**  **flag=0;}**  **else**  **{**  **printf("[Message]验证码错误,请重试\n");**  **flag++;**  **if(bottle!=0)**  **{while(getchar()!='\n');}**  **if(flag>5)**  **{printf("[Message]验证失败，结束程序\n");**  **exit(0);**  **}**  **}**  **}**  **}** |

**整合代码**

|  |
| --- |
| **/\*\*\*\*\*\*\*\*\*程序菜单开发\*\*\*\*\*\*\*\*\*\*\***  **开发人员：陈明新**  **开发环境：visual C++6.0**  **本程序只适用于　Windows系统 \*/**  **#include<stdio.h>**  **#include<math.h>**  **#include <conio.h>**  **#include<windows.h>**  **#include<time.h>**  **#define N 15//数组容量**  **#define M 10//数组容量**  **//函数声明**  **//界面显示**  **void desGui();//介绍界面**  **void loadGui();//加载页面**  **void menuGui();//菜单界面**  **void optionGui();//任务结束选项界面**  **//对opt数据处理**  **void get1();//菜单选项输入1-9**  **void menuOptions();//菜单得到数据后的执行函数**  **void get2();//任务结束选项输入1-9或a-c**  **void Suboptions();//任务结束选项得到数据后的执行函数**  **//菜单中的9个程序**  **void maxComDivisor();//求最大公约数**  **void counterExchang();//行列交换**  **void Bisearch();//折半查找**  **void primeNum();//素数**  **void Factorial();//n!**  **void bubbleSort();//冒泡排序**  **void selectSort();//选择排序**  **void PointSort();//用指针变量作参数排序**  **void day();//计算某一日在本年中的第几天**  **//错输入处理**  **void checkCode();//模拟验证码**  **//控制选择的变量**  **char opt;**  **//主函数**  **int main()**  **{loadGui();**  **system("color f2");//dos窗口输入命令：color f2；f是背景色：亮白；2是文字色：浅绿**  **system("title 程序菜单1.0.0 Beta版");//修改dos窗口名**  **desGui();**  **printf("按任意键继续...");**  **getch();//接受一个任意键的输入，不用按回车就返回，用到头文件conio.h**  **loadGui();**  **Sleep(100);//延迟执行语句100毫秒**  **menuGui();**  **return 0;**  **}**  **//介绍界面**  **void desGui()**  **{**  **printf("\t\t++===========================++\n");**  **printf("\t\t|| C语言实验菜单设计 ||\n");Sleep(50);**  **printf("\t\t++===========================++\n");Sleep(50);**  **printf("\t\t|| 开发：陈明新 ||\n");Sleep(50);**  **printf("\t\t|| 班级：计科1171 ||\n");Sleep(50);**  **printf("\t\t|| 版权：广东海洋大学 ||\n");Sleep(50);**  **printf("\t\t|| 数学与计算机学院 ||\n");Sleep(50);**  **printf("\t\t||提示：1.输入后按enter键; ||\n");Sleep(50);**  **printf("\t\t|| 2.数字的输入请在 ||\n");Sleep(50);**  **printf("\t\t|| 个位到千位数内; ||\n");Sleep(50);**  **printf("\t\t|| 3.dos窗口要调大; ||\n");Sleep(50);**  **printf("\t\t|| \*本程序仅供学习交流使用\* ||\n");Sleep(50);**  **printf("\t\t++===========================++\n");Sleep(50);**  **}**  **//加载页面**  **void loadGui()**  **{int flag=3;**  **while(flag)**  **{flag--;system("cls");printf("正在加载：..........");**  **printf("\n█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▄▄▃▂▁ ");system("cls");printf("正在加载：......");**  **printf("\n▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂ ");system("cls");printf("正在加载：...");**  **printf("\n▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▄▄▃▂▁▂▃ ");system("cls");printf("正在加载：...");**  **printf("\n▅▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▄▄▃▂▁▂▃▄ ");system("cls");printf("正在加载：......");**  **printf("\n▄▃▂▁▂▃▄▅▆▇█▇▆▅▄▃▂▁▂▃▄▅▆▇█▇▆▄▄▃▂▁▂▃▄▅ ");system("cls"); ;**  **}**  **printf("加载成功！\n");**  **}**  **//菜单界面**  **void menuGui()**  **{system("cls");//换页**  **printf("\n");**  **printf("\t\t++========================================================++\n");**  **printf("\t\t|| 程序菜单表 ||\n");Sleep(100);**  **printf("\t\t++========================================================++\n");Sleep(100);**  **printf("\t\t|| [1] 求两个正整数的最大公因数 ||\n");Sleep(100);**  **printf("\t\t|| [2] 二维数组的行和列的元素互换 ||\n");Sleep(100);**  **printf("\t\t|| [3] 折半法查找某数是在数组中的第几个元素的值 ||\n");Sleep(100);**  **printf("\t\t|| [4] 求m~n间的全部素数 ||\n");Sleep(100);**  **printf("\t\t|| [5] 用递归的方法求n的阶乘 ||\n");Sleep(100);**  **printf("\t\t|| [6] 对10个学生成绩用冒泡法进行由低到高的排序 ||\n");Sleep(100);**  **printf("\t\t|| [7] 对10个学生成绩用选择法进行由低到高的排序 ||\n");Sleep(100);**  **printf("\t\t|| [8] 用指针变量作为参数函数对两个整数进行大小排序 ||\n");Sleep(100);**  **printf("\t\t|| [9] 计算某一日在本年中的第几天 ||\n");Sleep(100);**  **printf("\t\t++========================================================++\n");Sleep(100);**  **get1();**  **}**  **//任务结束选项界面**  **void optionGui()**  **{printf("\n=================================================\n");**  **Sleep(500);**  **printf("++=======任务结束========++\n");Sleep(100);**  **printf("|| a.显示菜单 ||\n");Sleep(100);**  **printf("|| b.重新任务 ||\n");Sleep(100);**  **printf("|| c.退出程序 ||\n");Sleep(100);**  **printf("|| 1-9.选择任务 ||\n");Sleep(100);**  **printf("++=======================++\n");Sleep(100);**  **printf("请输入(a-c或1-9):\n");**  **get2();**  **}**  **//菜单得到数据后的执行函数**  **void menuOptions()**  **{switch(opt)**  **{case '1':maxComDivisor();break;**  **case '2':counterExchang();break;**  **case '3':Bisearch();break;**  **case '4':primeNum();;break;**  **case '5':Factorial();break;**  **case '6':bubbleSort();break;**  **case '7':selectSort();break;**  **case '8':PointSort();break;**  **case '9':day();break;**  **}**  **optionGui();**  **}**  **//任务结束选项得到数据后的执行函数**  **void Suboptions()**  **{switch(opt)**  **{case 'a':menuGui();break;**  **case 'c':exit(0);break;**  **default:menuOptions();break;**  **}**  **optionGui();**  **}**  **//菜单选项输入1-9**  **void get1()**  **{int flag=1,bottle;//flag判断输入对错，bottle判断输入缓冲区的第一个字符**  **printf("====================菜单选择=====================\n");**  **printf("输入菜单选项(1至9)：\n");**  **while(flag)**  **{**  **scanf("%c",&opt);**  **bottle=(getchar()!='\n');//bottle判断输入缓冲区的第一个字符是否为回车**  **if(opt>'0'&&opt<='9'&&bottle==0)**  **{menuOptions();**  **flag=0;}**  **else**  **{if(bottle!=0)**  **{while(getchar()!='\n');}//消除输入缓冲区**  **switch(flag)**  **{case 1:printf("[Message]输入好像有点问题，麻烦重新输入:\n");flag++;break;**  **case 2:printf("[Message]看清楚菜单再输入，谢谢配合:\n");flag++;break;**  **case 3:printf("[Message]你是想捣蛋吗？再给你次机会，重新输入:\n");flag++;break;**  **case 4:printf("[Message]Σ( ￣д￣；) 我现在怀疑你运行环境有问题!\n[Message]启动验证码系统!\n");checkCode();flag=1;printf("\n\n输入菜单选项(0至9)：\n");break;**  **}**    **}**  **}**  **}**  **//任务结束选项输入1-9或a-c**  **void get2()**  **{int flag=1,bottle;//flag判断输入对错，bottle判断输入缓冲区的第一个字符**  **char last=opt;//记下上一次的opt选择，用于重新任务**  **while(flag)**  **{**  **scanf("%c",&opt);**  **bottle=(getchar()!='\n');//bottle判断输入缓冲区的第一个字符是否为回车**  **if(opt>'0'&&opt<='9'&&bottle==0)**  **{menuOptions();**  **flag=0;}**  **else if(opt>='a'&&opt<='c'&&bottle==0)**  **{if(opt=='b')**  **{opt=last;}**  **Suboptions();**  **flag=0;}**  **else {if(bottle!=0)**  **{while(getchar()!='\n');}**  **switch(flag)**  **{case 1:printf("[Message]输入好像有点问题，麻烦重新输入:\n");flag++;break;**  **case 2:printf("[Message]看清楚菜单再输入，谢谢配合:\n");flag++;break;**  **case 3:printf("[Message]你是想捣蛋吗？再给你次机会，重新输入:\n");flag++;break;**  **case 4:printf("[Message]Σ( ￣д￣；) 我现在怀疑你运行环境有问题!\n[Message]启动验证码系统!\n");**  **checkCode();flag=1;optionGui();break;**  **}**    **}**  **}**  **menuOptions();**  **}**  **//模拟验证码**  **void checkCode()**  **{int i,bottle,flag=1;**  **char allCode[] = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJLMNOPQRSTUVWXYZ";//所有验证码**  **char getCode[5],writeCode[5];//getCode[5]是获得随机产生的字符，writeCode[5]是写入随机产生的字符**  **printf("=====================验证码=======================\n");Sleep(100);**  **while(flag)**  **{srand((unsigned)time(NULL));//取随机数种子，指定种子值为时间**  **printf("验证码：\n");**  **printf("+--------------------+\n");**  **printf("|");**  **for (i=0; i<4; i++)**  **{getCode[i]=allCode[rand()%(int)strlen(allCode)];//随机数在[0, allCode的有效长度值)中产生，并且产生的随机数作为allCode数组下标**  **printf("%4c ",getCode[i]);**  **}**  **getCode[i]='\0';**  **printf("|\n");**  **printf("+--------------------+\n");**  **printf("\n请输入验证码（不区分大小写）\n");**  **for (i=0; i<4; i++)**  **{**  **scanf("%c",&writeCode[i]);**  **}**  **bottle=(getchar()!='\n');//判断输入缓冲区是否溢出**  **writeCode[i]='\0';**  **strlwr(getCode);//大写变小写**  **strlwr(writeCode);//大写变小写**  **if(strcmp(getCode,writeCode)==0&&bottle==0)**  **{printf("[M1essage]验证成功\n");**  **flag=0;}**  **else**  **{**  **printf("[Message]验证码错误,请重试\n");**  **flag++;**  **if(bottle!=0)**  **{while(getchar()!='\n');}**  **if(flag>5)**  **{printf("[Message]验证失败，结束程序\n");**  **exit(0);**  **}**  **}**  **}**  **}**  **//求最大公约数**  **void maxComDivisor()**  **{int u,v,r,temp;//u大的数,v小的数,r是余数，temp中转站**  **loadGui();system("cls");**  **printf("===================最大公约数====================\n");Sleep(100);**  **printf("以半角逗号隔开输入2个数\n");**  **scanf("%d,%d",&u,&v);**  **while(!(u>0&&v>0))**  **{while(getchar()!='\n');**  **printf("[Message]输入错误，请重新输入:\n");**  **scanf("%d,%d",&u,&v);**  **}**  **while(getchar()!='\n');**  **if(v>u)**  **{temp=v;v=u;u=temp;}**  **while((r=u%v)!=0)//辗转相除法**  **{u=v;v=r;}**  **printf("最大公约数为:%d\n",v);**  **}**  **//行列交换**  **void counterExchang()**  **{int a[M][M],b[M][M],i,j,n,temp;//i行，j列，n：数组a有n\*n个元素**  **loadGui();system("cls");**  **printf("====================行列交换=====================\n");Sleep(100);**  **printf("定义数组a有n\*n个元素，输入n(1~6)：");**  **scanf("%d",&n);**  **while(!(n>=1&&n<=6))**  **{while(getchar()!='\n');**  **printf("[Message]输入错误，请重新输入:\n");**  **scanf("%d",&n);**  **}**  **while(getchar()!='\n');**  **for(i=0;i<n;i++)**  **{printf("输入第%d行的数组元素%d个(数字间以空格隔开):\n",i+1,n);**  **for(j=0;j<n;j++)**  **{**  **scanf("%d",&a[i][j]);**  **}**  **while(getchar()!='\n');**  **}**  **printf("数组a为：\n");**  **for(i=0;i<n;i++)**  **{for(j=0;j<n;j++)**  **{printf("%5d",a[i][j]);**  **}**  **printf("\n");**  **}**  **for(i=0;i<n;i++)**  **{for(j=i;j<n;j++)**  **{temp=a[i][j];**  **a[i][j]=a[j][i];**  **a[j][i]=temp;**  **}**  **}**  **for(i=0;i<n;i++)**  **{for(j=0;j<n;j++)**  **{b[i][j]=a[i][j];**  **}**  **}**  **printf("数组b为：\n");**  **for(i=0;i<n;i++)**  **{for(j=0;j<n;j++)**  **{printf("%5d",b[i][j]);**  **}**  **printf("\n");**  **}**  **}**  **//折半查找数组中数字**  **void Bisearch()**  **{int a[N]={105,45,30,15,14,13,12,10,9,8,5,4,3,2,1},i,num,flag=1,top,low,mid,find;**  **char c;**  **//i下标,num查找的数,flag判断输入内容,top查找范围最大值,low查找范围最小值,mid中间值,find判断是否找到，c选择是否继续**  **loadGui();system("cls");**  **printf("===================折半查找=====================\n");Sleep(100);**  **printf("[Message]是否手动输入(由大到小的)数组？(是按Y或其他，否按N)\n");**  **c=getchar();**  **while(getchar()!='\n');**  **if(c!='N'&&c!='n')**  **{printf("[Message]:输入15个元素，由大到小 (提示：首元素稍大)\n");**  **for(i=0;i<N;i++)**  **{printf("第%d个元素：",i+1);**  **scanf("%d",&a[i]);**  **while(getchar()!='\n');**  **if(i>0&&a[i]>a[i-1])**  **{printf("[Message]输入错误，请重新输入: (提示：由大到小)\n");**  **i--;flag++;**  **if(flag>5)**  **{checkCode();flag=1;}**  **}**  **}**  **}**  **printf("\n数组为：\n");**  **for(i=0;i<N;i++)**  **{printf("%6d",a[i]);}**  **Sleep(500);**  **flag=1;**  **while(flag)**  **{top=N-1;low=0;mid=0;find=0;**  **printf("\n输入想查找的元素：");**  **scanf("%d",&num);while(getchar()!='\n');**  **if(num<=a[0]&&num>=a[N-1])**  **{while((find==0)&&(low<=top))**  **{**  **mid=(top+low)/2;**  **printf("扫描位置范围:(%2d~%2d)\t中间值:%2d\n",low+1,top+1,mid+1);Sleep(300);//扫描查找范围输出**  **if(num==a[mid])**  **{find=1;}//表示找到了**  **else if(num>a[mid])//缩小扫描区间**  **{top=mid-1;}**  **else {low=mid+1;}**  **}**  **}**  **if(find==1)**  **{printf("[Message]找到数字:%d,于第%d个位置\n",num,mid+1);**  **}**  **else {printf("[Message]无此数数字%d\n",num);}**  **printf("[Message]是否重新查找？(是按Y或其他，否按N)\n");**  **c=getchar();while(getchar()!='\n');**  **if(c=='N'||c=='n')**  **flag=0;**  **}**  **}**  **//找素数**  **void primeNum()**  **{int m=0,n=0,i,num,count=0;**  **//m,n是搜索范围，i数组下标，num是范围内数字，count控制素数的数量**  **loadGui();system("cls");**  **printf("===================找出素数=====================\n");Sleep(100);**  **while(1)**  **{printf("输入起始值（正整数）m：\n");**  **scanf("%d",&m);**  **while(getchar()!='\n');**  **printf("输入终止值（正整数）n：\n");**  **scanf("%d",&n);**  **while(getchar()!='\n');**  **if(m>=n||m<=0)**  **{printf("输入错误，请重新输入\n");}**  **else break;**  **}**  **printf("素数有：\n");**  **for(num=m;num<=n;num++)**  **{for(i=2;i<=(int)sqrt(num);i++)**  **{if(num%i==0)//判断数素**  **{break;}**  **}**  **if(i>=(int)sqrt(num)+1&&num!=1)//判断并输出素数**  **{printf("%5d",num);**  **count++;**  **if(count%10==0&&count>=10)**  **{printf("\n");}**  **}**  **}**  **printf("\n在%d~%d内共有%d个素数.\n",m,n,count);**  **}**  **//算n!**  **void Factorial()**  **{int n,resu;**  **int FactorialFunction(int);**  **loadGui();system("cls");**  **printf("===================n的阶乘=====================\n");Sleep(100);**  **printf("输入(正整数)n: （提示:n不要太大）\n");**  **scanf("%d",&n);**  **while(getchar()!='\n');**  **while(n<=0)**  **{**  **printf("[Message]输入错误，请重新输入:\n");**  **scanf("%d",&n);**  **while(getchar()!='\n');**  **}**  **resu=FactorialFunction(n);**  **printf("%d!=%d\n",n,resu);**  **}**  **//n!功能模块**  **int FactorialFunction(int n)**  **{int f;**  **if(n==1||n==0)**  **f=1;**  **else**  **f=FactorialFunction(n-1)\*n;//递归**  **return f;**  **}**  **//对成绩用冒泡法进行由低到高的排序**  **/\*冒泡法优化，**  **优化1.添加标志位，记录最后一次交换位置的地方，**  **证明最后一次交换位置之后的地方时是排好序的，**  **下一次只需要排最后一次之前（或之后）的地方扫描就好了,**  **优化2.双向冒泡，每趟排序遍历一次找到一个最大值和者最小值，**  **其实都就是最后一次交换位置的地方，再从最大值和者最小值之间扫描排序，**  **使减少了一半的排序趟数 \*/**  **void bubbleSort()**  **{int a[M],i,j,k,temp,posit=0,high=M-1,low=0;**  **//i,j,k数组下标，temp中转站，poist是记下交换位置，high是扫描范围最大值，low是扫描范围最小值**  **loadGui();system("cls");**  **printf("===================冒泡排序=====================\n");Sleep(100);**  **for(i=0;i<M;i++)**  **{printf("第%d个学生成绩：",i+1);**  **scanf("%d",&a[i]);**  **while(getchar()!='\n');**  **}**  **printf("\n学生成绩数组为：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **while(low<high)**  **{for(i=low;i<high;i++)//正向冒泡**  **{if(a[i]>a[i+1])**  **{temp=a[i];**  **a[i]=a[i+1];**  **a[i+1]=temp;**  **posit=i;**  **printf("\nmax移至:%2d号位 [%3d]<-->[%3d] ",i+2,a[i],a[i+1]);printf("正向冒泡结果为：");**  **for(k=0;k<10;k++)//输出每趟排序结果**  **{**  **if(k==i+1) printf(" 【%d】",a[k]);**  **else printf(" %d ",a[k]);**  **}**  **Sleep(200);**  **}**  **}**  **high=posit;**  **for(j=high;j>low;j--)//逆向冒泡**  **{if(a[j]<a[j-1])**  **{temp=a[j];**  **a[j]=a[j-1];**  **a[j-1]=temp;**  **posit=j;**  **printf("\nmin移至:%2d号位 [%3d]<-->[%3d] ",j,a[j-1],a[j]);printf("逆向冒泡结果为：");**  **for(k=0;k<10;k++)//输出每趟排序结果**  **{**  **if(k==j-1) printf(" 【%d】",a[k]);**  **else printf(" %d ",a[k]);**  **}**  **Sleep(200);**  **}**  **}**  **low=posit;**  **}**  **printf("\n[Message]冒泡排序最终结果：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **}**  **/\*舍弃的代码：选择法优化算法一**  **void selectSort()**  **{int a[M],i,j,k,temp,posit=0,high=M-1,low=0;**  **loadGui();system("cls");**  **printf("===================选择排序=====================\n");Sleep(100);**  **for(i=0;i<M;i++)**  **{printf("第%d个学生成绩：",i+1);**  **scanf("%d",&a[i]);**  **while(getchar()!='\n');**  **}**  **printf("\n学生成绩数组为：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **for(i=0;i<M/2;i++)**  **{**  **for(j=low;j<M-i;j++)**  **{if(a[i]>a[j])**  **{temp=a[j];**  **a[j]=a[i];**  **a[i]=temp;**  **low=j;posit=1;**  **}**  **}**  **if(a[low]!=a[i+1]&&posit==1)**  **{temp=a[low];**  **a[low]=a[i+1];**  **a[i+1]=temp;**  **posit=0;**  **}**  **else low=i+1;**  **for(j=high;j>i;j--)**  **{if(a[M-i-1]<a[j])**  **{temp=a[M-i-1];**  **a[M-i-1]=a[j];**  **a[j]=temp;**  **temp=high;**  **high=j;posit=1;**  **}**  **}**  **if(a[high]!=a[M-i-2]&&posit==1)**  **{temp=a[high];**  **a[high]=a[M-i-2];**  **a[M-i-2]=temp;**  **posit=0;**  **}**  **else high=M-i-2;**  **}**  **printf("\n[Message]选择排序最终结果：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **}\*/**  **//对成绩用选择法进行由低到高的排序**  **/\*选择法优化：1.添加标志位，记录最后一次交换位置的地方；**  **2.双向选择法\*/**  **void selectSort()**  **{int a[M],i,j,k,temp,high,low;**  **//i,j,k数组下标，temp中转站，high是扫描范围最大值，low是扫描范围最小值**  **loadGui();system("cls");**  **printf("===================选择排序=====================\n");Sleep(100);**  **for(i=0;i<M;i++)**  **{printf("第%d个学生成绩：",i+1);**  **scanf("%d",&a[i]);**  **while(getchar()!='\n');**  **}**  **printf("\n学生成绩数组为：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **for(i=0;i<M/2;i++)**  **{high=M-i-1;low=i;**  **//正向选择**  **for(j=low;j<M-i;j++)**  **{if(a[low]>a[j])**  **{**  **low=j;//记下要交换的位置**  **}**  **}**  **temp=a[low];a[low]=a[i];a[i]=temp;**  **printf("\nlow 移至:%2d号位 [%3d]<-->[%3d] ",i+1,a[i],a[low]);printf("逆向选择结果为：");**  **for(k=0;k<10;k++)//输出每趟排序结果**  **{**  **if(k==i) printf(" 【%d】",a[k]);**  **else printf(" %d ",a[k]);**  **}**  **Sleep(200);**  **//逆向选择**  **for(j=high;j>i;j--)**  **{if(a[high]<a[j])**  **{**  **high=j;//记下要交换的位置**  **}**  **}**  **temp=a[high];a[high]=a[M-i-1];a[M-i-1]=temp;**  **printf("\nhigh移至:%2d号位 [%3d]<-->[%3d] ",M-i-1,a[i],a[low]);printf("逆向选择结果为：");**  **for(k=0;k<10;k++)//输出每趟排序结果**  **{**  **if(k==M-i-1) printf(" 【%d】",a[k]);**  **else printf(" %d ",a[k]);**  **}**  **Sleep(200);**  **}**  **printf("\n[Message]选择排序最终结果：\n");**  **for(k=0;k<10;k++)**  **{printf("%5d",a[k]);}**  **printf("\n");**  **}**  **//用指针变量作参数对两个整数排序**  **void PointSort()**  **{int \*p,\*q;**  **int a,b;**  **void maxSort(int \*p,int \*q);**  **loadGui();system("cls");**  **printf("================用指针参数排序===================\n");Sleep(100);**  **printf("输入两个整数（数字间用半角逗号隔开）：\n");**  **scanf("%d,%d",&a,&b);**  **while(getchar()!='\n');//消除输入缓冲区**  **p=&a;q=&b;**  **maxSort(p,q);**  **printf("由大到小：%d,%d\n",a,b);**  **}**  **//整数排序功能模块**  **void maxSort(int \*p,int \*q)**  **{int temp;**  **if(\*p<\*q)**  **{temp=\*p;**  **\*p=\*q;**  **\*q=temp;**  **}**  **}**  **//年月日结构体**  **struct ymd**  **{int year;**  **int mon;**  **int day;**  **};**  **//用指针变量作参数排序**  **void day()**  **{struct ymd date;int sum=0,flag=1,leap=0;//sum求得第几天，flag判断输入对错，leap判断闰年**  **loadGui();system("cls");**  **printf("==================年月日求天数===================\n");Sleep(100);**  **while(flag)**  **{**  **printf("输入年/月/日（数字间用/隔开,例如2018/1/1）\n");**  **scanf("%d/%d/%d",&date.year,&date.mon,&date.day);**  **while(getchar()!='\n');//消除输入缓冲区**  **if((date.year%4==0&&date.year%100!=0)||date.year%400==0)//判断闰年**  **{leap=1;**  **}**  **switch(date.mon)**  **{case 1:if(date.day<=31)sum=date.day;break;**  **case 2:if((date.day<=29&&leap==1)||(date.day<29))sum=date.day+31;else flag++;break;**  **case 3:if(date.day<=31)sum=date.day+59;else flag++;break;**  **case 4:if(date.day<31)sum=date.day+99;else flag++;break;**  **case 5:if(date.day<=31)sum=date.day+120;else flag++;break;**  **case 6:if(date.day<31)sum=date.day+151;else flag++;break;**  **case 7:if(date.day<=31)sum=date.day+181;else flag++;break;**  **case 8:if(date.day<=31)sum=date.day+212;else flag++;break;**  **case 9:if(date.day<31)sum=date.day+243;else flag++;break;**  **case 10:if(date.day<=31)sum=date.day+273;else flag++;break;**  **case 11:if(date.day<31)sum=date.day+304;else flag++;break;**  **case 12:if(date.day<=31)sum=date.day+334;else flag++;break;**  **default:flag++;break;**  **}**  **flag--;**  **if(flag) printf("输入错误，请重新输入\n");**  **if(date.mon>=3&&leap==1) sum++;**  **}**  **printf("%d月%d日在%d年是第%d天.\n",date.mon,date.day,date.year,sum);**  **}** |

**6.系统测试出现的问题和解决的方案：**

1．输入缓冲区问题，多个scanf语句的输入对于回车符或其他多余字符乱接收的问题，

用循环getchar（）解决

如

**有flag=1;**

**while(flag)**

**{**

**scanf("%c",&opt);**

**bottle=(getchar()!='\n');//bottle判断输入缓冲区的第一个字符是否为回车**

**if(opt>'0'&&opt<='9'&&bottle==0)**

**{menuOptions();**

**flag=0;}**

2．算法优化对各种情况考虑不周全问题

对算法认真分析，逐步排错。

找到特殊情况，多加判断

**7.系统优缺点：**

优点：用延迟命令做动画效果，输出信息位置准确到位，更美观；对冒泡算法，选择算法优化，效率更高，且输出算法过程，更清晰；有一定的容错性，排除输入缓冲区错误，附加验证码功能；操作提示照顾周全，让用户更好执行

缺点：容错性并不完全，不按照提示输入有可能出bug；代码过于繁杂；功能不多；页面不够整洁，用户需阅读量偏多

**8.心得体会：**

这次的程序设计练习，我渴望实现一些自己想要的效果。为追求高效，我尝试做算法的优化，为追求美观，我尝试去做动画输出，为追求容错性，我尝试去消除输入缓冲区和做验证码，为追求可读性，我去尝试用规范的命名，让代码的架构与分层合理，为追求正确性，我不断地测试和完善自己的程序……；有些时候我是找不到方向的，从而我也知道了自己有很多不足之处，有很多地方都无法理解，但我愿意去学习自己未知的领域，摸索新知识。于是，我百度找问题解决方法，去书上找资料，去思考代码其中的道理，掌握其中的知识，我对c语言和算法有了更深刻的理解。