《计算机程序设计》作业 **№-05及第4次上机**

# 作业四内容要点： 数组 循环 排序

【姓名 马宇骁 学号 PB19151769 】

**【要求】**

1. **在计算机上编程程序，加上必要的注释。**
2. **上机实验，经助教检查通过后，复制源码并记录实验结果，完成报告。**
3. **实验报告：记录调试及改错过程；****知识点或方法技巧的收获心得.**

**1 、 成绩统计。**

一个班有30个学生，每个学生有三门课。输入全部成绩，并计算每个人三门课的**平均成绩**，统计**平均85分及以上的人数**，用冒泡排序法将平均成绩按照从高到低的顺序**排序**、并**输出**排序后的平均成绩。

将下面的数据信息另存为一个文本文件 score.txt，使用**输入重定向**从该文件输入数据。

|  |
| --- |
| 1 46 95 77  2 66 88 15  3 74 87 80  4 36 73 71  5 76 25 69  6 76 82 68  7 13 91 38  8 96 80 90  9 83 80 42  10 30 71 83  11 54 95 74  12 97 77 100  13 81 93 67  14 0 83 79  15 34 92 59  16 83 85 96  17 59 83 50  18 73 80 24  19 75 65 68  20 57 77 52  21 97 88 85  22 55 88 74  23 56 83 33  24 93 80 85  25 38 60 80  26 77 86 71  27 85 86 83  28 35 82 56  29 44 90 74  30 96 82 77 |

1. 【源码】

#include<stdio.h>

#include<stdlib.h>

int main(){

freopen("score.txt","r",stdin);

int ss[30][4];

for(int i=0;i<30;i++){

for(int j=0;j<4;j++){

scanf("%d",&ss[i][j]);

}

}

fclose(stdin);

float s[30];

for(int i=0;i<30;i++){

s[i] = (ss[i][1]+ss[i][2]+ss[i][3])/3.0;

}

//冒泡

int temp;

for(int i=0;i<30-1;i++){

for(int j=0;j<30-i-1;j++){

if(s[j]<s[j+1]){

temp = s[j];

s[j]=s[j+1];

s[j+1]=temp;

}

}

}

//折半查找

int low=0,high=29,mid = (low+high)/2;

int p;

while(1){

if(s[mid]==85){

if(s[mid+1]==85){

mid = mid+1;

}

else{

p = mid;

break;

}

}

else if(s[mid]<85){

if(s[mid-1]>=85){

p = mid-1;

break;

}

else{

high = mid;

mid = (low+mid)/2;

}

}

else{

low = mid;

mid = (mid+high)/2;

}

}

//输出

printf("平均分85分及以上共有：%d人\n",p+1);

printf("所有人平均成绩从高到低为：\n");

for(int i=0;i<30;i++){

printf("%f\n",s[i]);

}

return 1;

}

1. 【运行结果】



1. 【实验报告】

用到了文件的内容，冒泡排序，并用改良的折半查找优化。

**2 、合并两个数组。**

现有数组 int a[20], b[10]; 对数组初始化: a[0]..a[9]中按照从小到大顺序存放了10个整数：{10，20，30，40，50，60，70，80，90，100}，b中有10个无序的整数 {1，43，72，23，9，87，54，3，55，0}。

编程序把b中的数据**合并**到a中，并保持**a数组仍然有序**。

1. 【源码】

int main(){

int a[20];

for(int i=0;i<10;i++){

a[i]=(i+1)\*10;

}

int b[10]={1,43,72,23,9,87,54,3,55,0};

for(int i=10;i<20;i++){

a[i] = b[i-10];

}

int temp;

for(int i=0;i<20-1;i++){

for(int j=0;j<20-i-1;j++){

if(a[j]>a[j+1]){

temp = a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

for(int i=0;i<20;i++){

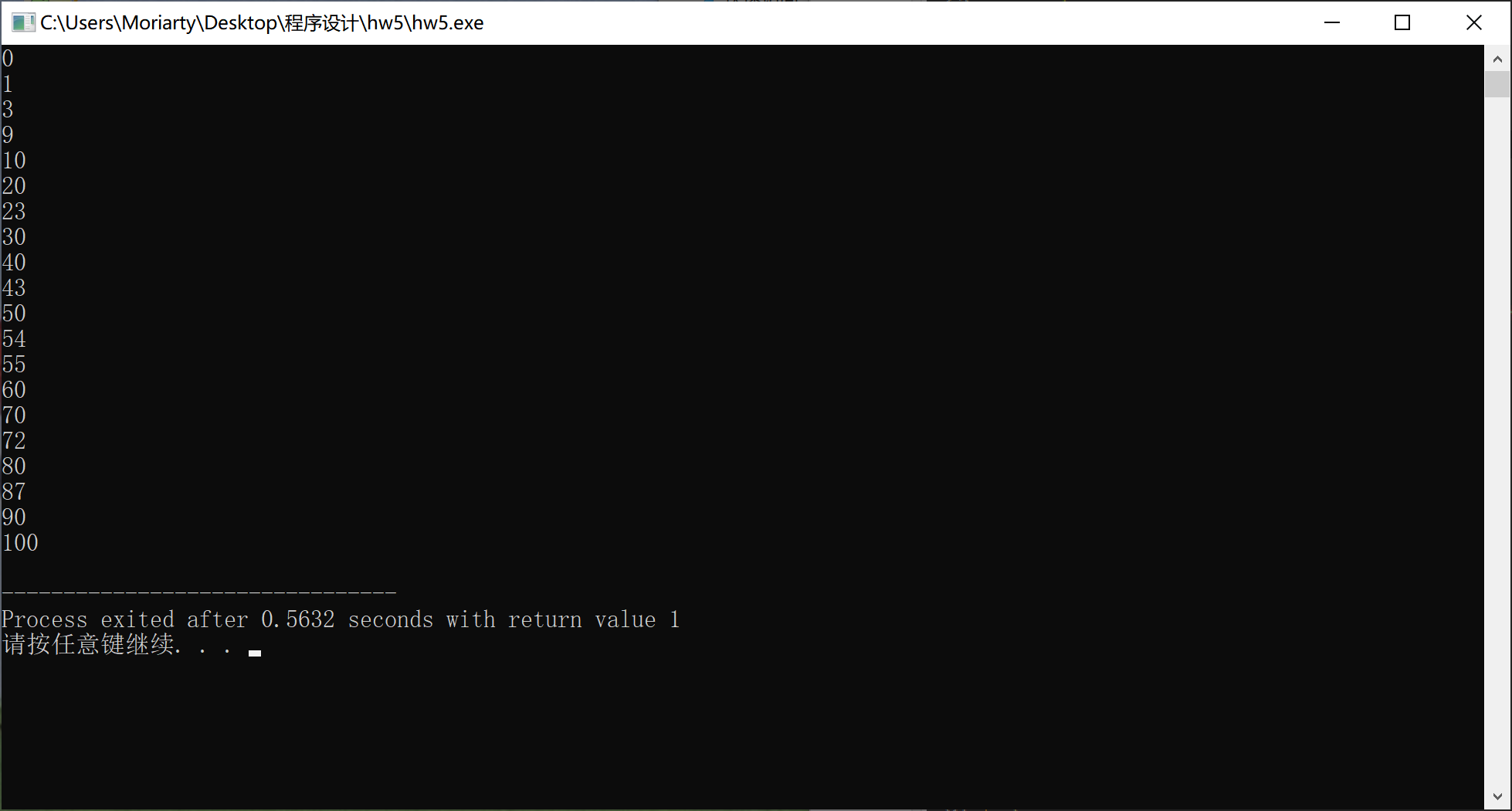
printf("%d\n",a[i]);

}

return 1;

}

1. 【运行结果】



1. 【实验报告】

其实在这里因为a本身有序，使用插入排序更适合，但因为第一题已经写过冒泡可以直接使用，故不做造车轮的功夫。

**3 、 约瑟夫环**

约瑟夫(Josephus)问题是由古罗马的史学家约瑟夫提出的,他参加并记录了公元 66-70 年犹太人反抗罗马的起义。约瑟夫作为一个将领带兵驻守裘达伯特城,在城市沦陷之后,他和 40多名将士在附近的一个洞穴中避难，将士们群情激奋地表示宁死不投降，于是约瑟夫提出自杀规则：所有n个人围坐一圈，从第一个人开始从1到m报数，报到m的人在战友协助下有尊严地结束其生命（对就是杀了他），下一个人重新报数，依次执行，直到最后一人则英勇而悲壮地自杀。故事的结尾显然约瑟夫同志留到了最后，他决定去当一个历史学家记录战士们的英勇事迹☺ 。

现在假设你就是约瑟夫，你也想当历史学家 ☺ 。需要解决当任意给定 n和 m 后, 求出最后留下来的人的编号。

编程序，设置数组 **soldiers[100],** **输入n (设n<=100)和m，** 输出最后一个剩下的编号。

例如：

输入n=5 ，m= 2

前四个被杀死人的顺序是： 2, 4, 3, 1, 最后留下来的是5

输出 5

（上面错了，应该是3）

1. 【源码】

int main(){

int n,m;

scanf("%d %d",&n,&m);

int soldiers[n];

for(int j=0;j<n;j++){

soldiers[j]=1;

}

int temp=0;

int cnt=0;

int i=0;

printf("死的顺序为：\n");

while(cnt<n){

if(i==(n)) i=0;

if (soldiers[i]==1){

temp++;

if(temp==m){

temp=0;

soldiers[i]=0;

cnt++;

printf("%d ",i+1);

}

}

i++;

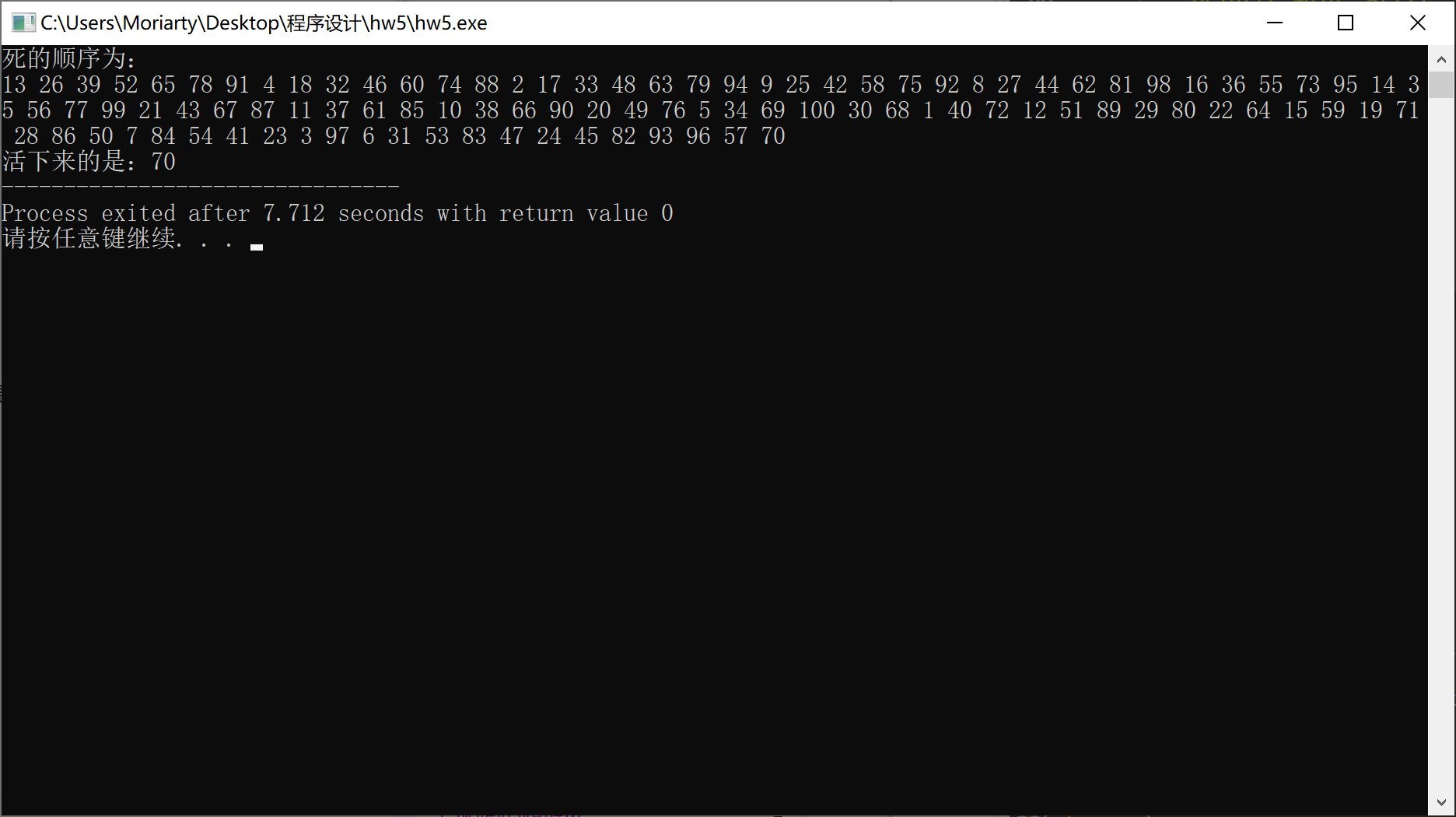
}

printf("\n");

printf("活下来的是：%d",i);

}

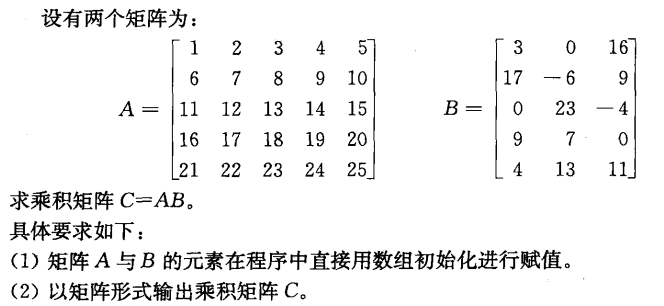
1. 【运行结果】



1. 【实验报告】

就是一个头尾相接的扩展版的报数游戏。

**4 、 矩阵相乘。**

****

1. 【源码】

int main(){

int A[5][5];

int B[5][3] = {{3,0,16},{17,-6,9},{0,23,-4},{9,7,0},{4,13,11}};

int C[5][3];

int temp=1;

for(int i=0;i<5;i++){

for(int j=0;j<5;j++){

A[i][j] = temp;

temp++;

}

}

for(int i =0;i<5;i++){

for(int j=0;j<3;j++){

temp = 0;

for(int k=0;k<5;k++){

temp = temp+A[i][k]\*B[k][j];

}

C[i][j]=temp;

printf("%d\t",C[i][j]);

}

printf("\n");

}

return 1;

}

1. 【运行结果】



1. 【实验报告】

考验矩阵乘法的知识。