**实验报告**

**课程名称：算法设计与分析**

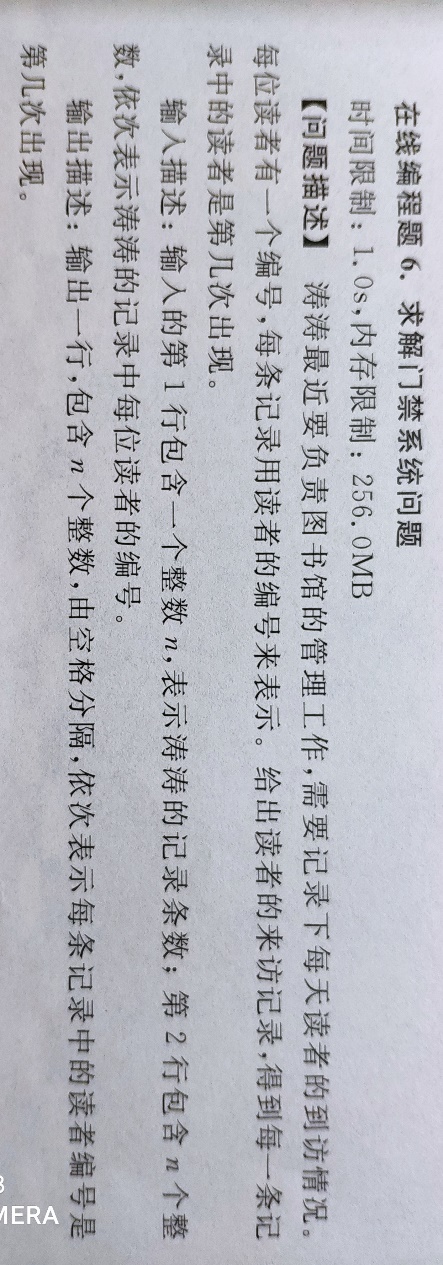
**专业班级：大数据182**

**学号姓名：3180439031陈佳婧**

**实验日期：2020年9月28日**

**第一章在线编程题6： 求解门禁系统问题**

1. **实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

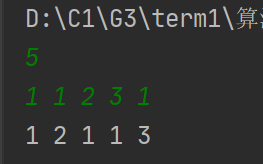
**使用map容器，每读入对应key，value自加一并输出。**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
#include <map>  
using namespace std;  
  
int main() {  
 int n;  
 map<int,int> m;  
 cin>>n;  
 for(int i=0;i<n;i++){  
 int x;  
 cin>>x;  
 m[x]++;  
 cout<<m[x];  
 if(i<n-1)cout<<' ';  
 else cout<<endl;  
 }  
 return 0;  
}

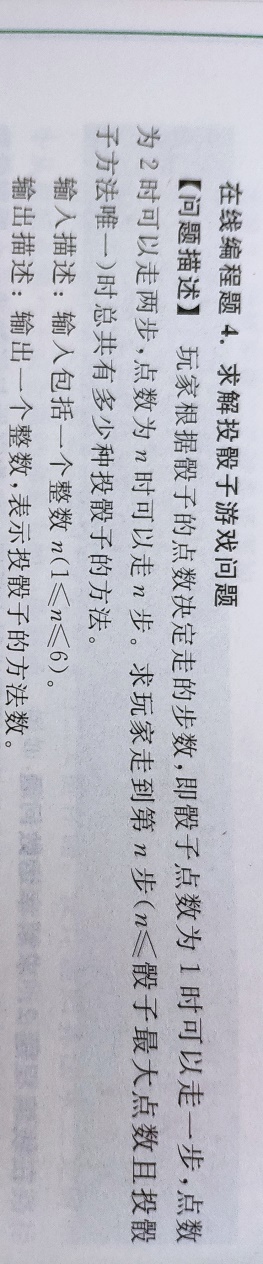
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**第二章在线编程题4： 求解投骰子游戏问题**

**1.实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

**F(x)=0,x<0**

**F(x)=1,x=0**

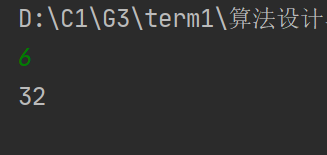
**F(x)=F(x-1)+ F(x-2)+ F(x-3)+ F(x-4)+ F(x-5)+ F(x-6),x>0**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
using namespace std;  
  
int toutouzi(int x){  
 if(x<0)return 0;  
 else if(x==0)return 1;  
 return toutouzi(x-1)+toutouzi(x-2)+toutouzi(x-3)+toutouzi(x-4)+toutouzi(x-5)+toutouzi(x-6);  
}  
int main() {  
 int n;  
 cin>>n;  
 cout<<toutouzi(n)<<endl;  
 return 0;  
}

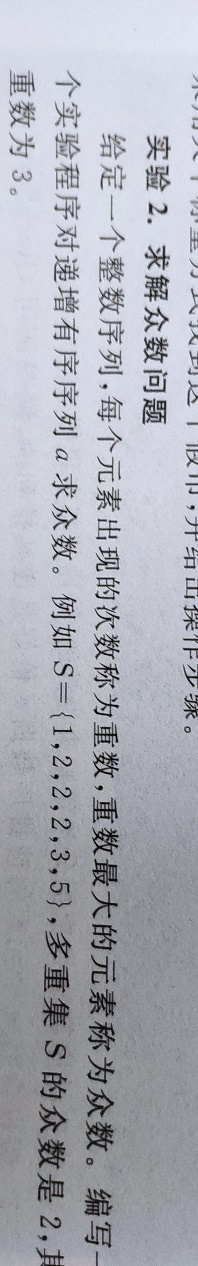
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**第三章在线编程题2： 求解众数问题**

**1.实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

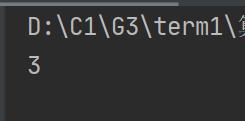
**由顶至底二路归并排序模板中，题中前提为从小到大，省去排序，分解，在合并中最先达成a[low]=a[high]的为众数。**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
using namespace std;  
  
int MergeFind(int a[],int low,int mid,int high){  
 int i=low,j=mid+1;  
 if(a[i]==a[j])return a[j];  
}  
int MergeZhongshu(int a[],int low,int high){  
 int mid;  
 if(low<high){  
 mid=(low+high)/2;  
 MergeZhongshu(a,low,mid);  
 MergeZhongshu(a,mid+1,high);  
 return MergeFind(a,low,mid,high);  
 }  
}  
int main() {  
 int a[]={1,2,2,3,3,3,5};  
 cout<<MergeZhongshu(a,0,6)<<endl;  
 return 0;  
}

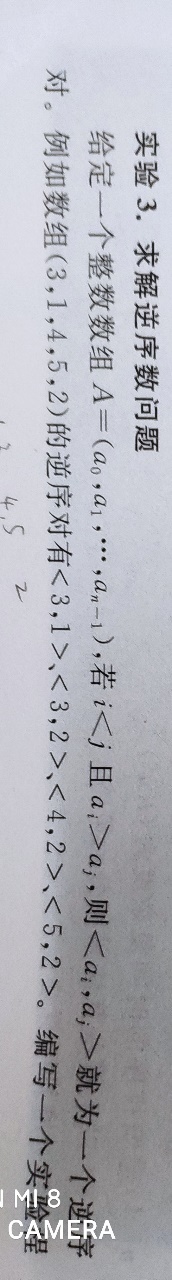
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**在线编程题3： 求解移动字符串问题**

**1.实验目标和要求：**



序采用分治法求A中逆序对的个数，即逆序数。

**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

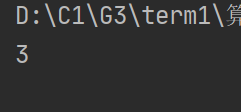
**在从顶至底二路归并排序中，合并函数Merge中每当a[i]>a[j]，令逆序对pairs+=mid-i+1。**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
using namespace std;  
  
int Merge(int a[],int low,int mid,int high){  
 int \*tmpa;  
 int pairs=0;  
 int i=low,j=mid+1,k=0;  
 tmpa=(int \*)malloc((high-low+1)\*sizeof(int));  
 while(i<=mid&&j<=high){  
 if(a[i]<=a[j]){  
 tmpa[k]=a[i];  
 i++;k++;  
 }else{  
 tmpa[k]=a[j];  
 j++;k++;  
 pairs+=mid-i+1;  
 }  
 }  
 while(i<=mid){  
 tmpa[k]=a[i];  
 i++;k++;  
 }  
 while(j<=high){  
 tmpa[k]=a[j];  
 j++;k++;  
 }  
 for(k=0,i=low;i<high;k++,i++){  
 a[i]=tmpa[k];  
 }  
 free(tmpa);  
 return pairs;  
}  
int MergeSort(int a[],int low,int high){  
 int mid;  
 if(low<high){  
 mid=(low+high)/2;  
 MergeSort(a,low,mid);  
 int hcnt=MergeSort(a,mid+1,high);  
 Merge(a,low,mid,high);  
 return hcnt;  
 }  
}  
  
int main() {  
 int a[]={3,1,5,2};  
 cout<<MergeSort(a,0,3)<<endl;  
 return 0;  
}

**6. 运行结果展示：**



**7. 实验体会（可选）：**