实验八

利用回溯法或分支界限法，从1-9中的自然数组成三个分数；

#include<stdio.h>  
#include<iostream>  
using name space std;  
struct myarr  
{  
int a[9];  
struct myarr \*next;  
}mya,myall;  
  
static void showArr(int arr[], int len = 9)  
{  
 cout<<arr[0];  
 cout<<"/";  
 cout<<"("<<arr[1]<<"+"<<arr[2]<<")";  
 cout<<"+ ";  
 cout<<arr[3];  
 cout<<"/";  
 cout<<"("<<arr[4]<<"+"<<arr[5]<<")";  
 cout<<"= ";  
 cout<<arr[6];  
 cout<<"/";  
 cout<<"("<<arr[7]<<"+"<<arr[8]<<")";  
 cout<<endl;  
}

Static int Max(int a,int b,int c)  
{  
intmax=a>b?a:b;  
returnmax=max>c?max:c;  
}  
  
//求最大公倍数  
static int MixCMul\_1(int a, int b,int c)  
{  
 int max = Max(a,b,c);  
 while(true)  
 {  
 if(max%a == 0 && max%b == 0 && max%c == 0)  
 break;  
 max++;  
 }  
 return max;  
}  
  
static void printmol1(int arr[], int len = 9)  
{  
 int mol[3] = {0}; // 分子  
 int Den[3] = {0}; // 分母  
 //按倍数扩充后的  
 int temmol[3] = {0}; // 分子  
 intMixCMul1=0, MixCMul2=0;  
 for(size\_t i = 0; i < 3; i++)  
 {  
 mol[i]= arr[i\*3];  
 Den[i]= arr[i\*3+1] + arr[i\*3+2];  
 }

}

//计算第一和第二个分数分母的最小公倍数，同时扩大分子  
MixCMul1 = MixCMul\_1(Den[0], Den[1],Den[2]);  
temmol[0]= MixCMul1/Den[0] \* mol[0];  
temmol[1]= MixCMul1/Den[1] \* mol[1];  
temmol[2]= MixCMul1/Den[2] \* mol[2];  
//分子的比较，去除分子比分母大的算式，  
  
if((temmol[2] == (temmol[0]+temmol[1])) && Den[0]>mol[0] &&Den[1]>mol[1] && Den[2]>mol[2])  
{  
 showArr(arr);  
}  
  
//数组的各种排列  
static void Grial1(int a[], int n,int m)  
{  
if(n == m)  
 {  
 printmol1(a);  
 }  
else  
{

for(int i = n; i < m; i++)  
 {  
 std::swap(a[i],a[n]);  
 Grial1(a,n+1,m);  
 std::swap(a[i],a[n]);  
 }  
}  
  
void get(int a[],int n)  
{  
int\*p=(int \*)malloc(sizeof(int)\*n);  
int\*q=p,\*head=p;  
for(int i = 0; i < 9; i++)  
{  
 p[i]=a[i];  
}  
for(intj=0;j<n;j++)  
{  
 cout<<p[sizeof(q)+j]<<"";  
}  
}

int main(int argc, char\* argv[])  
{  
 #define ARR\_NUM 9  
 intarr[ARR\_NUM] = {1, 2, 3, 4, 5, 6, 7, 8, 9};  
 Grial1(arr,0,ARR\_NUM);  
 return0;  
}