1.百鸡百钱问题二重优化

#include<iostream>

using namespace std;

int main()

{

int i,j,k,price,sum;

for(i=0;i<=20;i++) //100/5

for(j=0;j<=33;j++) //100/3

{

k=100-i-j;

price=i\*5+j\*3+k/3;

if(price==100)

cout<<"公鸡："<<i<<",母鸡："<<j<<",小鸡："<<k<<endl;

}

return 0;

**}**

2.求n内所以的完全数//如果一个数除自身以外，其它所有的约数的和等于该数，则该数是一个完全数

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int i,j,n,sum;**

**cin>>n;**

**for(i=1;i<=n;i++)**

**{**

**sum=0;**

**for(j=1;j<=i-1;j++)**

**if(i%j==0) sum+=j;**

**if(sum==i) cout<<i<<' ';**

**}**

**return 0;**

**}**

3.判断一个数是否是素数（提高执行效率）

**#include<iostream>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**int n,i;**

**for(n=2;n<=100;n++)**

**{**

**for(i=2;i<=sqrt(n);i++)**

**if(n%i==0) break;**

**if(i>sqrt(n)) cout<<n<<' ';**

**}**

**return 0;**

**}**

4. 一位法官在审理一起盗窃案时，对涉及到的四名嫌疑犯A、B、C、D进行了审问。

四人分别供述如下：

A：“罪犯在B、C、D三人之中。”

 B：“我没有作案，是C偷的。”

 C：“在A和D中间有一个是罪犯。”

 D：“B说的是事实”

经过调查，证实四人中只有两人说了真话，并且罪犯只有一个，请确定真正的罪犯。

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int a,b,c,d,m1,m2,m3,m4;**

**for(a=0;a<=1;a++)**

**for(b=0;b<=1;b++)**

**for(c=0;c<=1;c++)**

**for(d=0;d<=1;d++)**

**{**

**m1=(b+c+d==1); m2=(!b&&c);**

**m3=(a+d==1); m4=m2;**

**if(m1+m2+m3+m4==2&&a+b+c+d==1)**

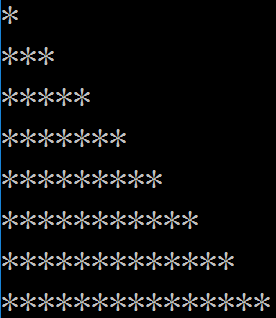
**cout<<a<<‘,'<<b<<‘,'<<c<<‘,'<<d;**

**}**

**return 0;**

**}**

5.图形输出



**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n,i,j;**

**cin>>n;**

**for(i=1;i<=n;i++)**

**{**

**for(j=1;j<=2\*i-1;j++)**

**cout<<"\*";**

**cout<<endl;**

**}**

**return 0;**

**}**



**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n,i,j;**

**cin>>n;**

**for(i=1;i<=n;i++)**

**{**

**for(j=1;j<=41-i;j++) cout<<' ';**

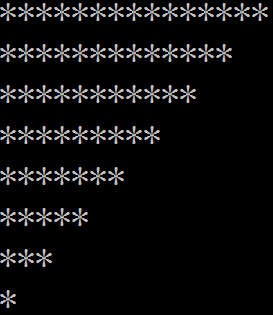
**for(j=1;j<=2\*i-1;j++) cout<<'\*';**

**cout<<endl;**

**}**

**return 0;**

**}**



**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n,i,j;**

**cin>>n;**

**for(i=n;i>=1;i--)**

**{**

**for(j=1;j<=2\*i-1;j++)**

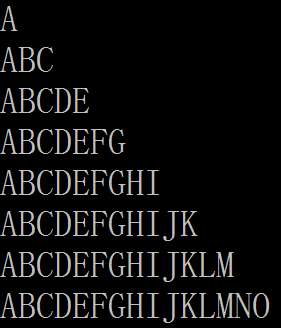
**cout<<'\*';**

**cout<<endl;**

**}**

**return 0;**

**}**



**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n,i,j;**

**char ch;**

**cin>>n;**

**for(i=1;i<=n;i++)**

**{**

**for(j=1;j<=2\*i-1;j++)**

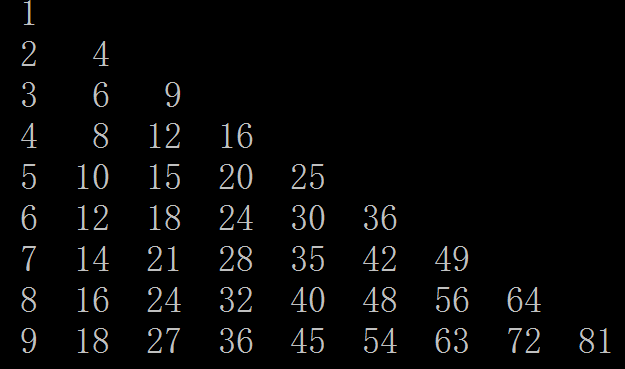
**{ ch='A'+j-1; cout<<ch;}**

**cout<<endl;;**

**}**

**return 0;**

**}**

（九九乘法表）

**#include<iostream>**

**#include<iomanip>**

**using namespace std;**

**int main()**

**{**

**int m,n;**

**//cout<<left;**

**for(m=1;m<10;m++)**

**{**

**for(n=1;n<=m;n++)**

**cout<<setw(3)<<m\*n;**

**cout<<endl;**

**}**

**return 0;**

**}**

6.求值 1-2+3-4+5-6+……-100

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int i,t=1,s=0;**

**for(i=1;i<=100;i++)**

**{**

**s+=t\*i;**

**t=-t;**

**}**

**cout<<s;**

**return 0;**

**}**

7.计算**1+1/2+2/3+3/5+……**

**#include<iostream>**

**#include<iomanip>**

**using namespace std;**

**int main()**

**{**

**int a=1,b=1,t,i;**

**double s=0;**

**for(i=1;i<=100;i++)**

**{**

**s+=(double)a/b;**

**t=b; b=a+b; a=t;**

**}**

**cout<<fixed<<setprecision(2)<<s;**

**return 0;**

**}**

8**求如下式子的值 1+1/2+1/3+1/4+…… 收敛判断fbas(1.0/i)<1e-6**

**#include<iostream>**

**#include<cmath>**

**#include<iomanip>**

**using namespace std;**

**int main()**

**{**

**int i=1;double s=1,m;**

**do**

**{**

**s+=m;**

**i++;**

**m=1.0/i;**

**}while(fabs(m)>=1e-6);**

**cout<<fixed<<setprecision(2)<<s;**

**return 0;**

**}**

9.请输入某班级数学成绩，再输出

#include<iostream>

#define ARR\_SIZE 40

using namespace std;

int main()

{

double score[ARR\_SIZE];

int n,i,num[ARR\_SIZE];

cout<<"从键盘输入学生人数 :";

cin>>n;

cout<<endl<<“分别输入学生的学号和成绩:“<<endl;

for(i=0;i<n;i++)

cin>>num[i]>>score[i];

cout<<"请输出学生的学号和成绩:"<<endl;

for(i=0;i<n;i++)

cout<<num[i]<<','<<score[i]<<endl;

return 0;

}

10.**从键盘输入学生人数n；从键盘输入所有学生的学号和成绩分别存入数组num和score**

**假设其中的一个学生成绩为最高maxScore = score[0]; 对所有学生成绩进行比较，即**

**for (i=1; i<n; i++)**

**{**

**若score[i] > maxScore**

**则修改maxScore值为score[i]**

**}打印最高分maxScore**

**#include<iostream>**

**#define ARR\_SIZE 40**

**using namespace std;**

**int main()**

**{**

**double score[ARR\_SIZE],maxScore;**

**int n,i;**

**int maxNum,num[ARR\_SIZE];**

**cout<<"请输入学生的总人数:";**

**cin>>n;**

**cout<<endl<<"请输入学生的学号和成绩:";**

**for(i=0;i<n;i++)**

**cin>>num[i]>>score[i];**

**maxScore = score[0];**

**maxNum=num[0];**

**for(i=1; i<n; i++)**

**if(score[i] > maxScore)**

**{**

**maxScore = score[i];**

**maxNum=num[i];**

**}**

**cout<<"maxScore="<<maxScore<<','<<"maxNum="<<maxNum;**

**return 0;**

**}**

11. **交换排序算法，将分数按从高到低顺序排列**

**#include<iostream>**

**#define ARR\_SIZE 40**

**using namespace std;**

**int main()**

**{**

**double score[ARR\_SIZE],temp1;**

**int n,i,j;**

**int num[ARR\_SIZE],temp2;**

**cout<<"请输入学生的总人数:"<<endl;**

**cin>>n;**

**cout<<"请输入学生的学号和成绩:"<<endl;**

**for(i=0;i<n;i++)**

**cin>>num[i]>>score[i];**

**for(i=0;i<=n-2;i++)**

**for(j=i+1;j<=n-1;j++)**

**if(score[j]>score[i])**

**{**

**temp1=score[j];score[j]=score[i];score[i]=temp1; temp2=num[j];num[j]=num[i];num[i]=temp2;**

**}**

**cout<<"Sorted result:"<<endl;**

**for(i=0;i<n;i++)**

**cout<<num[i]<<'\t'<<score[i]<<endl;**

**return 0;**

**}**

12.选择排序算法

**#include<iostream>**

**#define ARR\_SIZE 40**

**using namespace std;**

**int main()**

**{**

**double score[ARR\_SIZE],temp1;**

**int n,i,j,k;**

**int num[ARR\_SIZE],temp2;**

**cout<<"请输入学生的总人数:"<<endl;**

**cin>>n;**

**cout<<"请输入学生的学号和成绩:"<<endl;**

**for(i=0;i<=n-2;i++)**

**{ k=i;**

**for(j=i+1;j<=n-1;j++)**

**if(score[j]>score[k])**

**k=j;**

**if(k!=i)**

**{**

**temp1=score[k];score[k]=score[i];score[i]=temp1;**

**temp2=num[k];num[k]=num[i];num[i]=temp2;**

**}**

**}**

**cout<<"Sorted result:"<<endl;**

**for(i=0;i<n;i++)**

**cout<<num[i]<<'\t'<<score[i]<<endl;**

**return 0;**

**}**

13.顺序查找

**#include<iostream>**

**#define N 10**

**using namespace std;**

**int main()**

**{**

**int a[N],i,j,x,flag=0;**

**cout<<"请输入数组:\n";**

**for(i=0;i<=N-1;i++) cin>>a[i];**

**cout<<"请输入待查找的数:\n";**

**cin>>x;**

**for(i=0;i<=N-1;i++)**

**if(a[i]==x)**

**{**

**cout<<x<<"的下标是"<<i<<endl;**

**flag=1;**

**}**

**if(flag==0) cout<<x<<"没有找到";**

**return 0;**

**}**

14.折半查找

**#include<iostream>**

**#define N 10**

**using namespace std;**

**int main()**

**{**

**int low,high,mid,i,a[N],x,pos,find=0;**

**cout<<"请输入从小到大排序的10个正整数\n";**

**for(i=0;i<=N-1;i++)**

**cin>>a[i];**

**cout<<"请输入待查找的数\n";**

**cin>>x;**

**low=0; high=N-1;**

**while (low <= high)**

**{**

**mid = (high + low) / 2;**

**if(x > a[mid])**

**low = mid + 1;**

**else if(x < a[mid])**

**high = mid - 1;**

**else**

**{find=1;pos=mid;break;}**

**}**

**if(find==1) cout<<x<<"已将找到，位置为"<<mid;**

**else cout<<"没有找到";**

**return 0;**

**}**

15. **求n!。**

#include<iostream>

#include<cstdlib>

using namespace std;

int main()

{

double fac(int n);

int n;

double m;

cin>>n;

if (n<0) exit(0);

m=fac(n);

cout<<n<<"!="<<m<<endl;

return 0;

}

double fac(int n)

{

double f;

if (n==0)

f=1;

else

f=fac(n-1)\*n;

return f;

}

16.（指针）**输入两个数至两个变量，并交换这两个变量的值**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int \*p1,\*p2,\*p,a,b,t;**

**cout<<"请输入a、b的值：“<<endl;**

**cin>>a>>b;**

**p1=&a;p2=&b;p=&t;**

**cout<<"交换前\n----------------\n";**

**cout<<"a=“<<a<<“b=“<<b<<endl;**

**cout<<"\*p1=“<<\*p1<<“\*p2=“<<\*p2<<endl;**

**cout<<"p1=“<<p1<<“p2=“<<p2<<endl;**

**if(a<b)**

**{\*p=\*p1;\*p1=\*p2;\*p2=\*p;}**

**cout<<“交换后\n----------------\n";**

**cout<<"a=“<<a<<“b=“<<b<<endl;**

**cout<<"\*p1=“<<\*p1<<“\*p2=“<<\*p2<<endl;**

**cout<<"p1=“<<p1<<“p2=“<<p2<<endl;**

**}**

17.杨辉三角

**#include <iostream>**

**#define M 3**

**#define N 2**

**using namespace std;**

**int main()**

**{**

**int i,j,n=0,a[N][N]={0};**

**while(n<1 || n>10)**

**{**

**cout<<"请输入杨辉三角形的行数:";**

**cin>>n;**

**}**

**for(i=0;i<=n-1;i++)**

**{**

**a[i][0]=1;**

**a[i][i]=1;**

**}**

**for(i=2;i<=n-1;i++)**

**for(j=1;j<i;j++)**

**a[i][j]=a[i-1][j-1]+a[i-1][j];**

**for(i=0;i<=n-1;i++)**

**{**

**for(j=0;j<=i;j++)**

**cout<<a[i][j]<<'\t';**

**cout<<endl;**

**}**

**return 0;**

**}**

单链表

头插法

LinkList CreateFromHead()

{

LinkList L; Node \*s; int flag=1; ElemType x;

L=new Node; //为头结点分配存储空间

//c语言语法形式 L= (LinkList)malloc(sizeof(Node))；

L->next=NULL;

while(flag)

{ cin>>x;

if(x!=-1)

{ **s=new Node;** //c语言语法形式 :s=(Node \*)malloc(sizeof(Node));

s->data=x;

**s->next=L->next; L->next=s;**

} else flag=0;

}

return L;

}

尾插法

LinkList CreateFromTail() //将新增的字符追加到链表的末尾

{

LinkList L; Node \*r,\*s; int flag =1; ElemType x;

L=new Node; //生成头节点

L->next=NULL;

r=L; //r指针始终动态指向链表的当前表尾

while(flag) //标志,初值为1,输入'$'时flag为0,建表结束

{ cin>>x;

if(x!=-1)

{ **s=new Node**; s->data=x; **r->next=s**; **r=s**; }

else { flag=0; r->next=NULL;}

}

return L;

}

单链表插入

int InsList(LinkList L,int i,ElemType e)

{ int k=0;

Node \*pre=L,\*s; //先找到第i-1个数据元素的存储位置,使指针pre指向它

while( pre!=NULL && k<i-1 )

{ pre=pre->next; k=k+1; }

if(pre==NULL)

{ puts("插入位置不合理!");return 0; }

s=new Node; //为e申请一个新的结点

s->data=e; //将待插入结点的值e赋给s的数据域

s->next=pre->next;

pre->next=s; //注意次序及头节点的作用

return 1;

}

单链表删除

int DelList(LinkList L,int i,ElemType \*e)

{

Node \*pre=L,\*r; int k=0;

while( pre->next!=NULL && k<i-1 ) //寻找被删除结点i的前驱结点

{ pre=pre->next; k=k+1; }

if(pre->next==NULL)

{ puts("删除结点的位置i不合理!"); return 0; }

\*e=pre->next->data;

r=pre->next; pre->next=r->next; //删除结点r

delete r; //释放被删除的结点所占的内存空间

return 1;

}