11.6-数组经典实例2

代码示例1：

约瑟夫环问题：

n个人围成一圈，从第一个人开始报数,数到 m 的人出圈，再由下一个人重新从 1 开始报数，数到 m 的人再出圈，依次类推，直到所有的人都出圈，请输出依次出圈人的编号。

输入格式

输入两个整数n,m（1<=n,m<=100）

输出格式

输出一行 n 个整数，按顺序输出每个出圈人的编号。

样例输入：

10 3

样例输出：

3 6 9 2 7 1 8 5 10 4

int m,n;

cin>>n>>m;

int i=1; //i代表当前报数人的编号

int isout[101]={0}; //isout[i]的值为1代表编号为i的人已经出圈，0代表尚未出圈

int count=0; //用于计数，每到m的时候，当前人出圈

int out\_num=0; //用于记录已经出局的人数，当out\_num==n时循环结束

while(out\_num<n)

{

if(isout[i]==0)

{

count++; //当前编号的人还没出局的情况下，每一次循环代表有1人报数

}

if(count%m==0) // && isout[i]==0

{

isout[i]=1; //当前编号的人出局

cout<<i<<" "; //输出其编号

out\_num++; //出局人数+1

}

i++; //下一次报数是下一个编号的人

if(i==n+1)

{

i=1; //如果到最后一人了，再从1号开始报数

}

}

代码示例2：

编写一个程序，将一个x进制的数转换为y进制的形式。

输入：第一行输入x，y，表示是将x进制的数转换为对应的y进制数；第二行输入一个x进制的数（不超过10位），若输入的数不是x进制的数则输出“ERROR！请重新输入：”，然后接下来的两行重新输入x、y和x进制的数，直到输入合法为止。输入的x，y应为2、8、10、16中的一个，但不需要检验x、y输入的合法性。

输出：对应的y进制数

样例输入1：

16 10

2A

样例输出：

42

样例输入2：

8 10

19

8 2

27

样例输出2：

ERROR！

10111

int x,y;

char num[11];

bool judge=true;

int len,i;

while(judge)

{

judge=false;

cin>>x>>y;

cin.get();

cin.getline(num,10); //[0]位置是最高位

len=strlen(num); //求输入的x进制数有多少位

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

{

//将输入的x进制数由字符形式转换为数值(ASCII码)形式

if(num[i]>='0' && num[i]<='9')

{

num[i]-=48; //使得num[i]由字符0-9变为ASCII码为0-9

}

if(num[i]>='A' && num[i]<='F')

{

num[i]-=55; //使得num[i]由字符A-F变为ASCII码为10-15

}

//cout<<(int)num[i];

if(num[i]>=x)

{

//用ASCII码与x进行比较，此时为不合法输入

cout<<"ERROR!请重新输入:"<<endl;

judge=true;

break;

}

}

}

//将x进制数转换为十进制数

int decimal=0; //x进制转换为十进制的结果

int power=1; //计算乘方时幂的次数

int tmp;

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

{

power=len-1-i;

tmp=num[i];

while(power>0)

{

/\*

//num[i]是char型数据，其对应的ASCII码有限制(不超过256)，所以例如下面的输入则无法通过:

//输入16 2 10E

//原因在于1那一位对应的256数值无法作为ASCII码值赋给char型的num[i],所以这里用int型的变量tmp处理

num[i]\*=x;

cout<<power<<" "<<(int)num[i]<<endl;

\*/

tmp\*=x;

power--;

}

decimal+=tmp;

//cout<<decimal<<endl;

}

//将十进制数转换为y进制

int j=0,res\_len=0;

int res[40]={-1}; //10位十六进制数可转换为40位二进制数

char res\_16[40];

for(i=0;i<40;i++)

{

res\_16[i]='A';

}

while(decimal>0)

{

res[j]=decimal%y; //存储结果的数组res中，[0]位置是最低位

decimal=(decimal-res[j])/y;

if(res[j]>=10)

{

res\_16[i]+=res[j]-10; //相当于将字符A的ASCII码加上对应的数值，实现10-15与A-F的对应

}

j++;

}

for(i=j-1;i>=0;i--) //因为[0]位置是最低位，所以从[j-1]开始输出

{

if(res[i]<10)

{

cout<<res[i];

}

else

{

cout<<res\_16[i];

}

}

验证数据：

