**C语言 第四次作业**

**自81 2018011446 刘泓尊**

1. **字符筛选**

#include<stdio.h>

#include<string.h>

int main()

{

char a[100],ch;

printf("please input the string: ");

scanf\_s("%s",a,100);

printf("please input the char you want to delete: ");

getchar();

scanf\_s("%c",&ch);

int len=strlen(a);

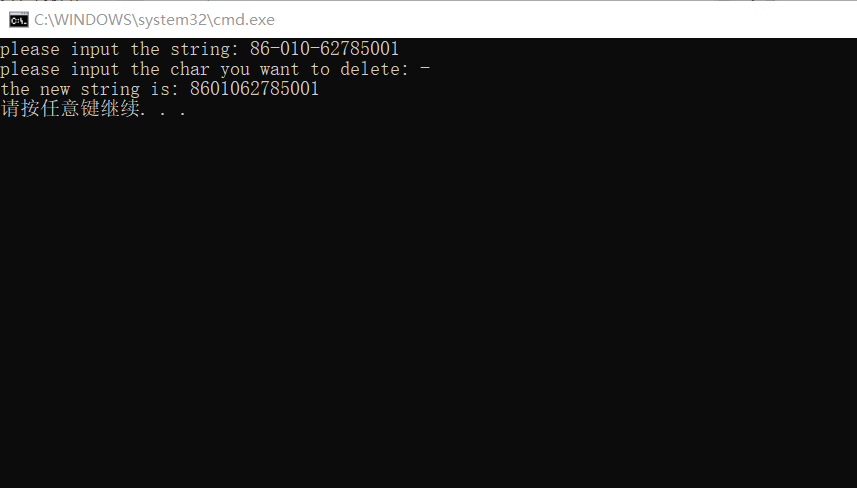
for(int i=0;i<=len;i++)

if(a[i]==ch)for(int j=i+1;j<=len+1;j++)a[j-1]=a[j];

printf("the new string is: %s\n",a);

return 0;

}



**2.子串筛选**

#include<stdio.h>

#include<string.h>

int main()

{

char str1[100],str[10];

int i,j,k;

printf("Please input str1:");

scanf\_s("%s",str1,100);

getchar();

printf("Please input str:");

scanf\_s("%s",str,10);

int len1=strlen(str1),len2=strlen(str);

printf("The new string is:");

for(i=0;i<len1;i++)

{

k=1;

for(j=0;i+j<len1&&j<len2;j++)

if(str1[i+j]!=str[j]){k=0;break;}

if(!k)putchar(str1[i]);

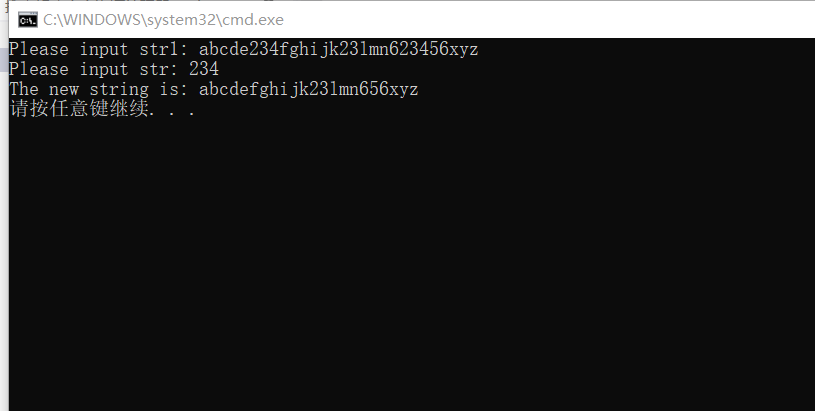
else i+=len2-1;

}

printf("\n");

return 0;

}

****

**3.字符串合并**

#include<stdio.h>

#include<string.h>

int main()

{

char str1[20],str2[20];

printf("Please input str1:");

scanf\_s("%s",str1,20);

printf("Please input str2:");

scanf\_s("%s",str2,20);

int i;

int len1=strlen(str1),len2=strlen(str2);

int len=(len1<=len2)?len1:len2;

for(i=0;i<len;i++)//先输出有配对字符的字符部分

{

putchar(str1[i]);

putchar(str2[i]);

}

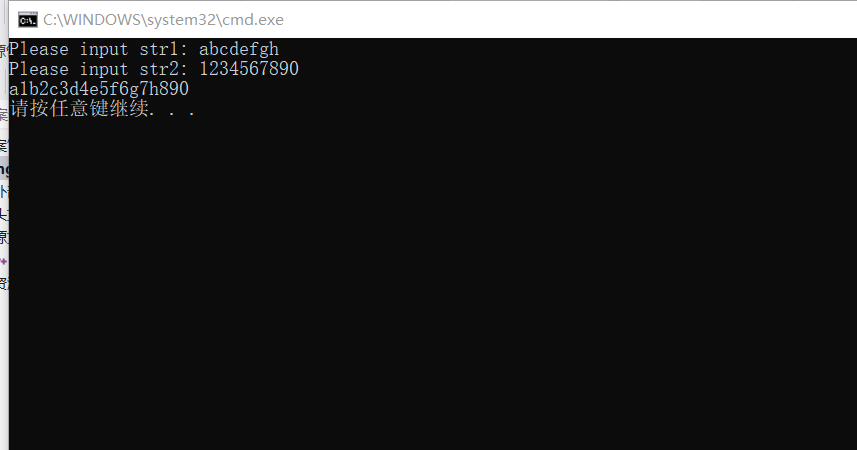
if(len!=len2)for(i=len;i<len2;i++)putchar(str2[i]);//再输出不能配对的剩余字符部分

else for(i=len;i<len1;i++)putchar(str1[i]);

printf("\n");

return 0;

}

****

**4.数据合并**

#include<stdio.h>

int main()

{

int num1[20],num2[20],num[40];

int len1,len2,n,k,m;

printf("Please input the 1st array:");

for(len1=0;len1<20;len1++)

{

scanf\_s("%d",&num1[len1]);

if(num1[len1]==0){len1--;break;}

}

printf("Please input 2nd array:");

for(len2=0;len2<20;len2++)

{

scanf\_s("%d",&num2[len2]);

if(num2[len2]==0){len2--;break;}

}

printf("\nThe new array is:\n");

for(k=0;k<=len1;k++)

num[k]=num1[k];

for(k=len1+1;k<=len1+len2+1;k++)

num[k]=num2[k-len1-1];

for(k=len1+len2+1;k>=0;k--)//冒泡排序

for(m=0;m<k;m++)

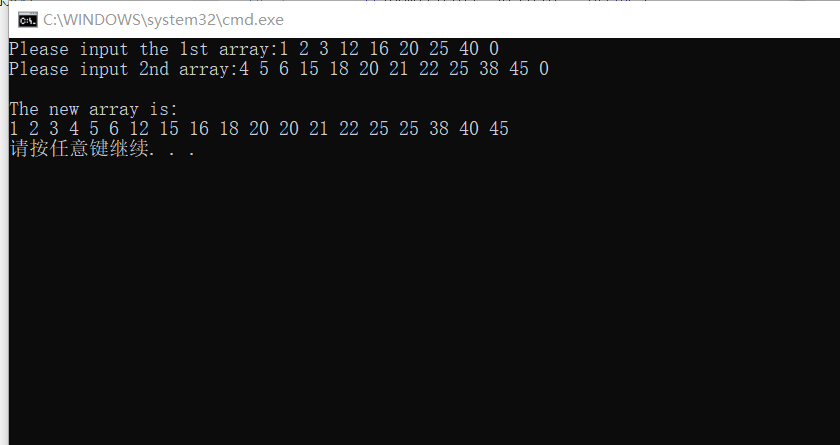
if(num[m]>num[m+1]){int tmp=num[m];num[m]=num[m+1];num[m+1]=tmp;}

for(k=0;k<=len1+len2+1;k++)printf("%d ",num[k]);

printf("\n");

return 0;

}

****

**5.数制转换**

#include<stdio.h>

#include<string.h>

#include<math.h>

int main()

{

char Binarynumber[32];

int Decimalnumber=0;

printf("Please input a binary number: ");

scanf\_s("%s",Binarynumber,32);

int len=strlen(Binarynumber),i;

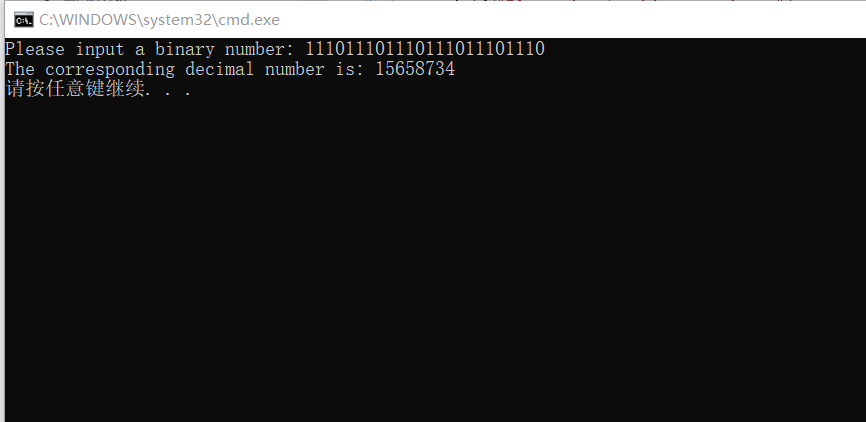
for(i=0;i<len;i++)

if(Binarynumber[i]=='1')Decimalnumber+=pow(2.0,len-1-i);

printf("The corresponding decimal number is: %d\n",Decimalnumber);

return 0;

}

****

**6.对称数字**

#include<stdio.h>

#include<math.h>

int main()

{

int Begin=95859,End,delta;

int v;

int Judge(int x);

if(Judge(12345))printf("1");

for(End=95859+30\*2;End<=95859+120\*2;End++)

{

delta=End-Begin;

if(Judge(End))

{

v=delta/2;

printf("The symmetrical number is: %d\nThe speed is:%3d km/h\n",End,v);

}

}

return 0;

}

int Judge(int x)//判断一个五位数是否为对称数

{

int a[6],i=1,flag=1;

for(i=1;i<=5;i++)

{

a[i]=x%10;

x/=10;

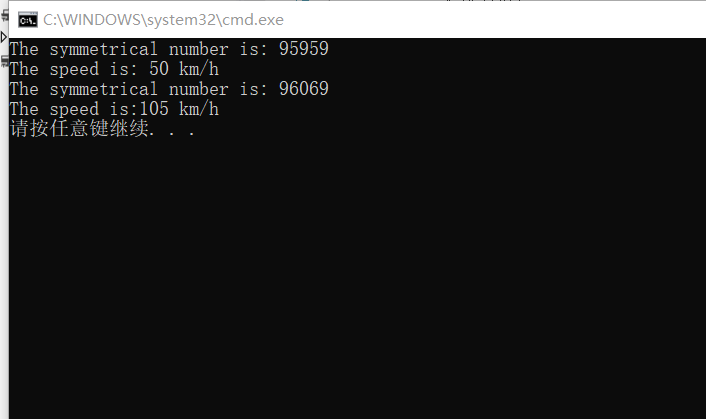
}

for(i=1;i<=3;i++)

if(a[i]!=a[6-i]){flag=0;break;}

return flag;

}

****

**7.矩阵元素的运算**

#include<stdio.h>

int main()

{

int i,j,n,a[11][11],sum=0;

printf("Please input the value of n:");

scanf\_s("%d",&n);

printf("Please input the elements of the matrix:\n");

for(i=1;i<=n;i++)for(j=1;j<=n;j++)scanf\_s("%d",&a[i][j]);

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

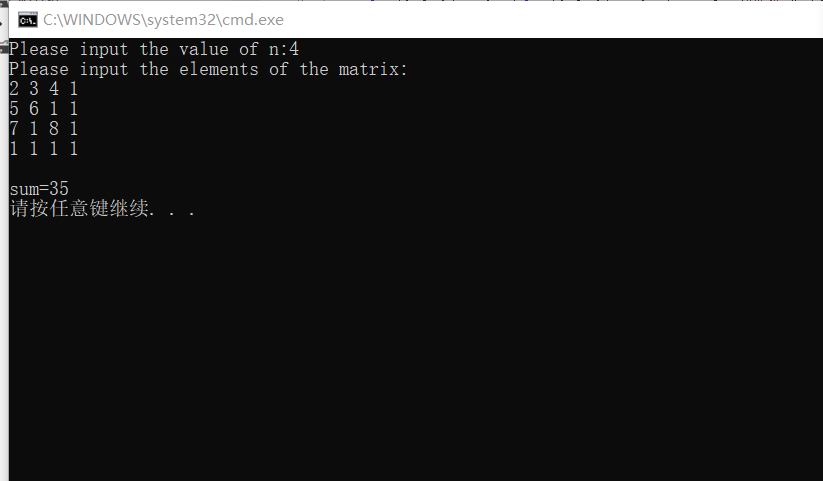
for(j=1;j<n;j++)

if((i+j)!=(n+1))sum+=a[i][j];

printf("\nsum=%d\n",sum);

return 0;

}

****

**8.加法口诀表**

#include<stdio.h>

int main()

{

int a[11][11],i,j,n;

printf("Please input the value of n:");

scanf\_s("%d",&n);

for(j=1;j<=n;j++)a[0][j]=j;

for(i=1;i<=n;i++)a[i][0]=i;

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

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

a[i][j]=a[i][0]+a[0][j];

printf(" +");

for(j=1;j<=n;j++)printf("%3d",a[0][j]);

printf("\n");

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

{

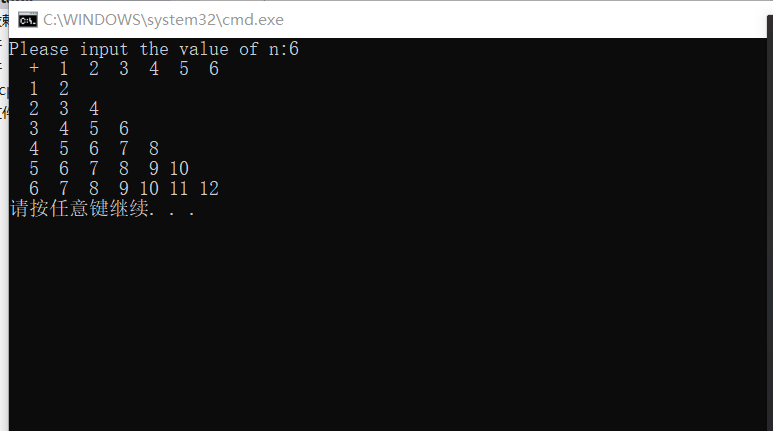
for(j=0;j<=i;j++)printf("%3d",a[i][j]);

printf("\n");

}

return 0;

}

****

**9.矩阵排序**

#include<stdio.h>

#include<math.h>

int main()

{

int m,n,a[22][22],b[21][21],tmp[110];

printf("Please input the value of m and n:");

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

int i,j,t,k=1,flag;

printf("Please input the elements of a:\n");

for(i=1;i<=m;i++)for(j=1;j<=n;j++)scanf\_s("%d",&a[i][j]);

printf("\n Matrix a:\n");

for(i=1;i<=m;i++)//输出数组a

{

for(j=1;j<=n;j++)printf("%4d",a[i][j]);

printf("\n");

}

printf("\n Matrix b:\n");

for(i=0;i<=m;i++){a[i][0]=0;a[i][n+1]=0;}

for(j=0;j<=n;j++){a[0][j]=0;a[m+1][j]=0;}//使边界元素的值为0

for(i=1;i<=m;i++)

for(j=1;j<=n;j++)

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

for(i=1;i<=m;i++)//输出数组b并将其放入一维数组tmp

{

for(j=1;j<=n;j++)

{

tmp[k]=b[i][j];

k++;

printf("%4d",b[i][j]);

}

printf("\n");

}

printf("\n Matrix c:\n");

for(k=m\*n;k>=1;k--)//冒泡排序

for(j=1;j<k;j++)

if(tmp[j]<tmp[j+1]){t=tmp[j];tmp[j]=tmp[j+1];tmp[j+1]=t;}

for(k=1;k<=m\*n;k++)//找到b数组中与tmp[k]对应的元素并输出数组a中的对应元素

{

flag=0;

for(i=1;i<=m;i++)

{

for(j=1;j<=n;j++)

if(b[i][j]==tmp[k])

{

printf("%4d",a[i][j]);

b[i][j]=tmp[1]+1;

flag=1;

break;

}

if(flag)break;

}

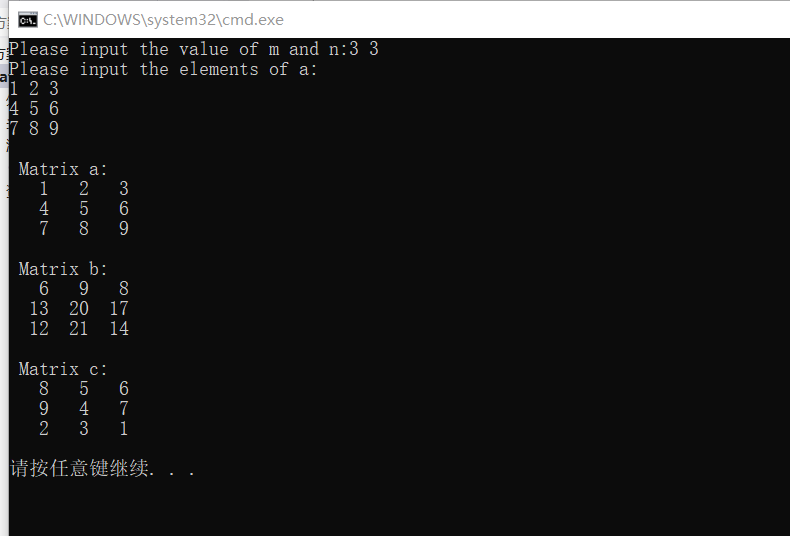
if(!(k%n))printf("\n");//每行的元素输出满时，换行

}

printf("\n");

return 0;

}

****

**10.字符与单词**

#include<stdio.h>

#include<conio.h>

#include<string.h>

int main()

{

char s[1201],ch1[10],ch2[10];

char \*p;

int count=0;

int i,k=0,flag=0,Countletter=0,Countblank=0,Countnumber=0,Countelse=0;

void str\_replace(char \* cp, int n, char \* str);

while((s[k]=\_getch())!='\n')

{

if(s[k]>='a'&&s[k]<='z')putchar(s[k]-'a'+'A');

else if(s[k]>='A'&&s[k]<='Z')putchar(s[k]-'A'+'a');

else if(s[k]!='.') putchar(s[k]);

else {putchar(s[k]); break;}

k++;

}

k++;

printf("\n");

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

{

if((s[i]>='a'&&s[i]<='z')||(s[i]>='A'&&s[i]<='Z'))Countletter++;

else if(s[i]==' ')Countblank++;

else if(s[i]>='0'&&s[i]<='9')Countnumber++;

else Countelse++;

}

printf("\n英文字母个数:%d\n空格个数为:%d\n数字个数为:%d\n其他字符个数:%d\n",Countletter,Countblank,Countnumber,Countelse-1);

printf("\n请输入被替换与替换的单词：");

scanf\_s("%s",ch1,10);

scanf\_s("%s",ch2,10);

p = strstr(s,ch1);

while(p)

{

count++;

//每找到一个str2，就用str3来替换

str\_replace(p,strlen(ch1),ch2);

p = p+strlen(ch2);

p = strstr(p,ch1);

}

s[k+strlen(ch2)-strlen(ch1)]='\0';

printf("%s\n",s);

return 0;

}

void str\_replace(char \* cp, int n, char \* str)

{

int lengthofstr;

char \* tmp;

lengthofstr = strlen(str);

if(lengthofstr < n)//str3比str2短，往前移动

{

tmp = cp+n;

while(\*tmp)

{

\*(tmp-(n-lengthofstr)) = \*tmp; //n-lenofstr为移动的距离

tmp++;

}

\*(tmp-(n-lengthofstr)) = \*tmp; //移动'\0'

}

else

if(lengthofstr > n)//str3比str2长，往后移动

{

tmp = cp;

while(\*tmp) tmp++;

while(tmp>=cp+n)

{

\*(tmp+(lengthofstr-n)) = \*tmp;

tmp--;

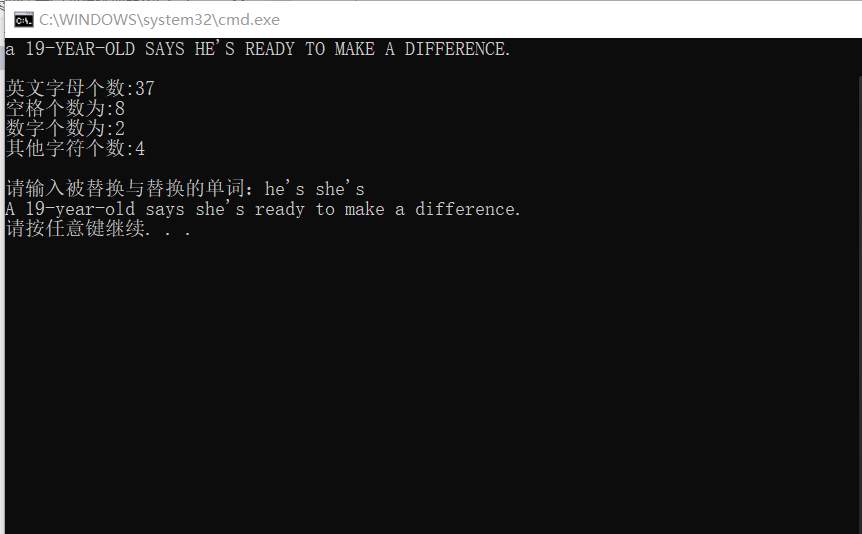
}

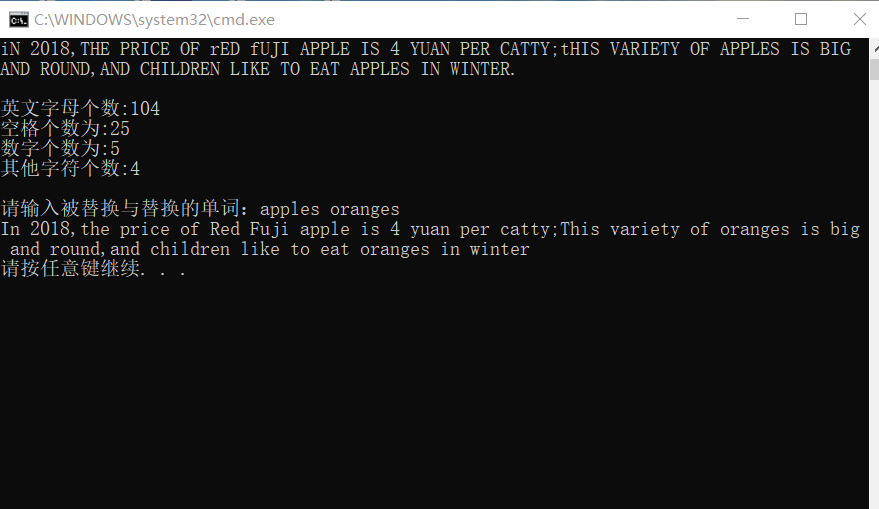
\*(tmp+(lengthofstr-n)) = \*tmp; //移动'\0'

}

strncpy(cp,str,lengthofstr);

}



**11.环形铁路路径**

#include<stdio.h>

int main()

{

int StationNumber[150],CheckOrder[150];//CheckOrder存放检查顺序

int m,n,i,j,count=1,t;

printf("Please input the value of n,i,m:");

scanf\_s("%d%d%d",&n,&i,&m);

for(j=1;j<=n;j++)StationNumber[j]=j;//车站编号初始化

CheckOrder[1]=i;

StationNumber[i]=0;//检查过之后标记为0

j=i;

while(count<n)

{

t=0;

while(t<=m)//实现每隔m个检查一个，并且已检查的车站跳过

{

j=(j+1)%n;

if(!j)j=n;//实现环形检查

if(StationNumber[j]!=0)t++;//判断是否被检查过

}

count++;

CheckOrder[count]=StationNumber[j];

StationNumber[j]=0;//检查过之后标记为0

}

printf("\nThe order is:\n");

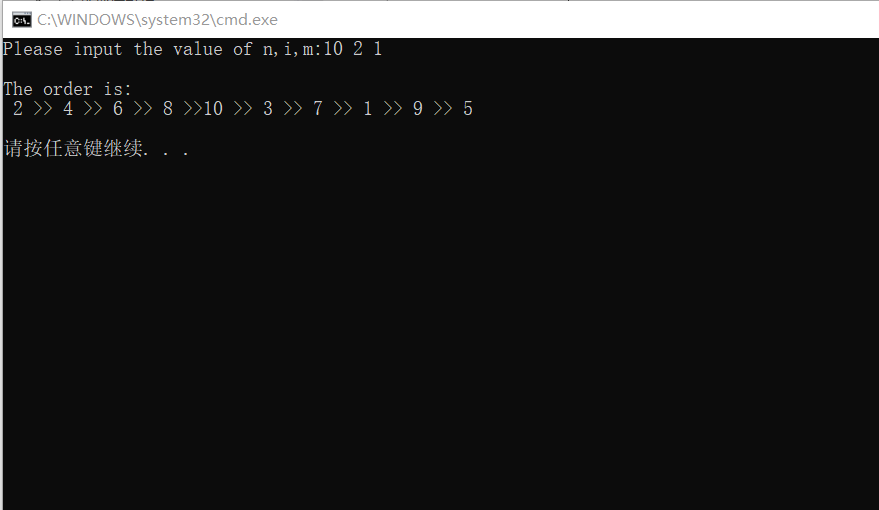
for(j=1;j<n;j++)

printf("%2d >>",CheckOrder[j]);

printf("%2d\n\n",CheckOrder[n]);

return 0;

}

****

**12.矩阵的初始化**

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

int main()

{

int m,n,j,k;

printf("Please input the m and n:");

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

int a[20][20];

srand(time(NULL));

printf("the matrix is:\n");

for(j=0;j<m;j++)

{

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

{

a[j][k]=(rand()%(m\*n+1));

printf("%4d",a[j][k]);

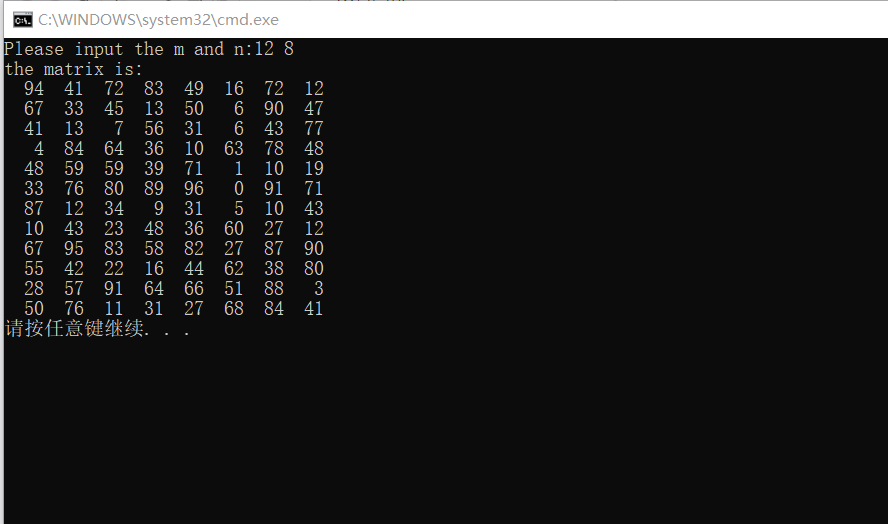
}

printf("\n");

}

return 0;

}

****

**13.大数求和**

#include<stdio.h>

#include<string.h>

int main()

{

char num1[51],num2[51];

int sum[60],tmp[51],i,j,len1,len2,len;

printf("please input the first number :");

for(i=0;(num1[i]=getchar())!='\n';i++);

num1[i]='\0';

printf("please input the second number:");

for(i=0;(num2[i]=getchar())!='\n';i++);

num2[i]='\0';

len1=strlen(num1);

len2=strlen(num2);

len=(len1>=len2)?len1:len2;

for(i=0;i<=len;i++)//各位初始化为0

{

sum[i]=0;

tmp[i]=0;

}

for(i=len1-1,j=0;i>=0;i--,j++)//将num1倒序存入sum中，方便 进位运算

sum[j]=num1[i]-'0';

for(i=len2-1,j=0;i>=0;i--,j++)

tmp[j]=num2[i]-'0';//将num2倒序存入tmp中

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

{

sum[i]+=tmp[i];

if(sum[i]>9)

{

sum[i]-=10;

sum[i+1]+=1;//进位

}

}

if(sum[len]!=0)len+=1;

printf("sum=");

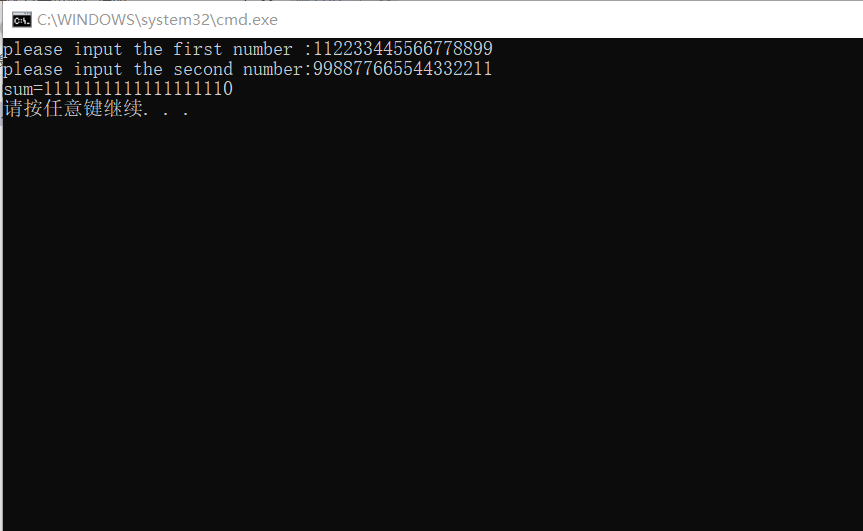
for(i=len-1;i>=0;i--)//倒序输出sum

printf("%d",sum[i]);

printf("\n");

return 0;

}

****