# 历年真题程序整理：

## 2018年机试答案

### 求n组数的乘积

#include<stdio.h>

#include<stdlib.h>

int main()

{

printf("Input n:\n");

int n;

scanf("%d",&n);

int sum[n];

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

{

int num1,num2;

scanf("%d %d",&num1,&num2);

sum[i]=num1\*num2;

}

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

{

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

printf("\n");

}

return 0;

}

### 输出阶乘

#include<stdio.h>

#include<stdlib.h>

int jiecheng(int num1)

{

if(num1==1)

{

return 1;

}else

{

return num1\*jiecheng(num1-1);

}

}

int main()

{

printf("Input n:\n");

int n;

scanf("%d",&n);

int sum[n];

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

{

int num1;

scanf("%d",&num1);

sum[i]=jiecheng(num1);

}

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

{

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

printf("\n");

}

return 0;

}

输出C（N，M） ///C(n,m)=n!/(m!\*(n-m)!)

#include<stdio.h>

#include<stdlib.h>

int jc(int num)

{

if(num==1)

{

return 1;

}else

{

return num\*jc(num-1);

}

}

int countNum(int num1,int num2)

{

if(num1>num2)

{

int nump1=jc(num1);

int nump2=jc(num2);

int nump3=jc(num1-num2);

return nump1/(nump2\*nump3);

}else

{

return 0;

}

}

int main()

{

int sum;

int num1,num2;

scanf("%d %d",&num1,&num2);

sum=countNum(num1,num2);

printf("%d",sum);

printf("\n");

return 0;

}

### 倒序输出

#include <stdio.h>

#include <stdlib.h>

#define Maxsize 20

struct ststack{

char data[Maxsize];

int top;

}st;

void display(char exp[],char postexp[])

{

char ch;

int i=0,j=0;

st.top=-1;

ch=exp[i];

i++;

while(ch!='\0')

{

st.data[st.top]=ch;

st.top++;

ch=exp[i];

i++;

}

st.top--;

while(st.top!=-1)

{

postexp[j]=st.data[st.top];

printf("%c",postexp[j]);

j++;

st.top--;

}

printf("%c",st.data[st.top]);

}

int main() {

char exp[20];

char postexp[20];

scanf("%s",exp);

display(exp,postexp);

return 0;

}

### 判断字符是不是一样

#include <stdio.h>

#include <stdlib.h>

int display(char exp[],int len)

{

char ch;

int i=0;

ch=exp[i];

while(ch!='\0')

{

char ch1=exp[len-i-1];

if(ch!=ch1)

{

return 0;

break;

}

ch=exp[i];

i++;

return 1;

}

return 0;

}

int main() {

char exp[100];

scanf("%s",exp);

int len=strlen(exp);

int h=display(exp,len);

if(h==1)

{

printf("succeed");

}else

{

printf("error");

}

return 0;

}

## 2017年机试答案

### 1.求平均时间：

#include<stdio.h>

int main()

{

printf("请输入时间:\n");

int time[6];

int sum[2],sum1,sum2;

int h[2],m[2],s[2];

for(int i=0;i<2;i++){

scanf("%d %d %d %d %d %d",&time[0],&time[1],&time[2],&time[3],&time[4],&time[5]);

}

for(int i=0;i<2;i++){

if(time[0]<11&&time[3]<11){

sum1=time[0]\*3600+time[1]\*60+time[2];

sum2=time[3]\*3600+time[4]\*60+time[5];

sum[i]=(sum1+sum2)/2;

}else

{

printf("ERROR");

break;

}

}

for(int i=0;i<2;i++){

h[i]=sum[i]/3600;

m[i]=(sum[i]%3600)/60;

s[i]=(sum[i]%3600)%60;

printf("%d %d %d \n",h[i],m[i],s[i]);

}

}

/\*就是求平均值，要明白的是，全部算成秒，之后除以二，在计算

小时是/3600，分钟是%3600/60，秒是%3600%60\*/

### 2.输入n组数的排序

#include<stdio.h>

int bubble\_sort(int arr[],int len);

int main()

{

static int total[1000];

char ch;

int n;

printf("Please input n:\n");

scanf("%d",&n);

int num[n][1000];

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

do

{

scanf("%d",&num[i][total[i]++]);

}while((ch=getchar())!= '\n');// 这里用来判断是否输入了回车

bubble\_sort(num[i],total[i]);

}

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

{

for(int j=0;j<total[i];j++){

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

}

printf("\n");

}

return 0;

}

int bubble\_sort(int arr[],int len)

{

int i,j,temp;

for(i=0;i<len-1;i++)

{

for(j=0;j<len-1-i;j++)

{

if(arr[j]>arr[j+1])

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

}

#### 选择排序

#include<stdio.h>

///Selection Sort(选择排序)

void sort(int arr[],int len)

{

int minIndex,temp;

int i,j;

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

{

minIndex=i;

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

{

if(arr[j]<arr[minIndex])

{

minIndex=j;

}

}

temp=arr[i];

arr[i]=arr[minIndex];

arr[minIndex]=temp;

}

return temp;

}

int main()

{

int arr[10];

int i;

for(i=0;i<10;i++)

{

scanf("%d",&arr[i]);

}

sort(arr,10);

for(i=0;i<10;i++)

{

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

}

}

### 3.输入表达式，求表达式的值，，后缀表达式来计算（2019-2-26）

#include<stdio.h>

#include<string.h>

#define MaxSize 100

///符号栈

struct opstack{

char data[MaxSize]; ///存储操作符

int top; ///存储栈顶指针

}op;

///数值栈

struct ststack{

float data[MaxSize]; ///存储操作数

int top;

}st;

///将算数运算符转化为后缀表达式postexp

void trans(char exp[],char postexp[])

{

char ch;

int i=0,j=0; ///i用来扫描exp的下标,j用来扫描postexp的下标

op.top=-1;

ch=exp[i];

i++;

while(ch!='\0')

{

switch(ch){

case '(' :///左括号

op.top++;

op.data[op.top]=ch;

break;

case ')': ///右边括号

///若遇到右括弧“）”，则连续出栈输出，直到遇到左括弧“（”为止。其中，左括弧出栈但是并不输出

while(op.data[op.top]!='(')

{

postexp[j]=op.data[op.top];

j++;

op.top--;

}

op.top--;

break;

case '+':

case '-':

while(op.top!=-1 && op.data[op.top]!='(')

{

postexp[j]=op.data[op.top];

j++;

op.top--;

}

op.top++;

op.data[op.top]=ch; ///入栈

break;

case '\*':

case '/': ///为'\*'或'/'时，其优先级不大于栈顶为'\*'或'/'的优先级，直到')'

while(op.top!=-1 && op.data[op.top]!='(' && (op.data[op.top]=='\*'||op.data[op.top]=='/'))

{

postexp[j]=op.data[op.top];

j++;

op.top--;

}

op.top++;

op.data[op.top]=ch;

break;

case ' ':

break;

default:

while(ch>='0'&&ch<='9')

{

postexp[j]=ch;

j++;

ch=exp[i];

i++;

}

i--;

postexp[j]='#';

j++;

}

ch=exp[i];

i++;

}

while(op.top!=-1) { //此时，exp扫描完毕，栈不空时出栈并存放到postexp中

postexp[j]=op.data[op.top];

j++;

op.top--;

}

postexp[j]='\0'; //给postexp表达式添加结束标识

}

///对于后缀表达式postexp求值

float compvalue(char postexp[])

{

float d;

char ch;

int i=0;

st.top=-1;

ch=postexp[i];

i++;

while(ch!='\0')

{

switch(ch)

{

case '+':

st.data[st.top-1]=st.data[st.top-1]+st.data[st.top];

st.top--;

break;

case '-':

st.data[st.top-1]=st.data[st.top-1]-st.data[st.top];

st.top--;

break;

case '\*':

st.data[st.top-1]=st.data[st.top-1]\*st.data[st.top];

st.top--;

break;

case '/':

if(st.data[st.top]!=0){

st.data[st.top-1]=st.data[st.top-1]/st.data[st.top];

}else{

printf("\n\t /0 is error");

exit(0);

}

st.top--;

break;

default:

d=0;

while(ch>='0'&&ch<='9')

{

d=10\*d+ch-'0';

ch=postexp[i];

i++;

}

///这里就是把字符串转为了数字值

st.top++;

st.data[st.top]=d;

}

ch=postexp[i];

i++;

}

return st.data[st.top]; ///输出栈顶元素就是结果

}

int main()

{

int n;

printf("Please input n:");

scanf("%d",&n);

float f[n];

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

char exp[20];

scanf("%s",exp);

char postexp[30];

//求exp的逆波兰式,得到postexp

trans(exp,postexp);

//对postexp求值

f[i]=compvalue(postexp);

}

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

printf("%.2f\n",f[i]);

}

return 0;

}

### 4.匹配表达式

#include<stdio.h>

int match(char exp[])

{

int len=strlen(exp);

for(int i=0;i<len/2;i++)

{

if(exp[i]=='('&&exp[len-1-i]==')')

{

return 1;

}else if(exp[i]=='['&&exp[len-1-i]==']')

{

return 1;

}else if(exp[i]=='{'&&exp[len-1-i]=='}'){

return 1;

}else{

return 0;

}

}

}

int main()

{

int n;

printf("Please input n:");

scanf("%d",&n);

char ch[100];

int judge[n];

printf("Please input like {1[]:\n");

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

{

scanf("%s",ch);

judge[i]=match(ch);

}

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

{

if(judge[i]==1)

{

printf("yes\n");

}else{

printf("no\n");

}

}

}

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define Maxsize 50

typedef char Datatype;

typedef struct

{

Datatype data[Maxsize];

int top;

}SqStack;

void InitStack(SqStack \*S)

{

S->top=-1;

}

///判断是不是栈空

int StackEmpty(SqStack \*S)

{

if(S->top==-1)

{

return 0;

}else

{

return 1;

}

}

int push(SqStack \*S,Datatype x)

{

if(S->top==Maxsize-1)

{

return 0;

}else

{

S->data[++(S->top)]=x;

return 1;

}

}

int Pop(SqStack \*S,Datatype \*x)

{

if(S->top==-1)

{

return 0;

}else

{

\*x=S->data[(S->top)--];

return 1;

}

}

int GetTop(SqStack S,Datatype \*x)

{

if(S.top==-1)

return 0;

else

\*x=S.data[S.top];

return 1;

}

int Bracketscheck(SqStack \*S,char \*str)

{

InitStack(S);

char e;

int i=0;

while(str[i]!='\0')

{

switch(str[i])

{

case '(':

push(S,'(');

break;

case '{':

push(S,'({');

break;

case '[':

push(S,'[');

break;

case ')':

Pop(S,&e);

if(e!='(')

{

return 0;

}

break;

case ']':

Pop(S,&e);

if(e!='[')

{

return 0;

}

break;

case '}':

Pop(S,&e);

if(e!='{')

{

return 0;

}

break;

default:

break;

}

i++;

}

int h=StackEmpty(S);

if(h==1)

{

return 0;

}else

{

return 1;

}

}

int main()

{

SqStack S;

char str[Maxsize];

//printf("%s",str);

gets(str);

int h=Bracketscheck(&S,str);

if(h==0)

printf("括号不匹配");

else

printf("括号匹配");

return 0;

}

## 2016年题目程序答案：

### 1.求平均时间

#include<stdio.h>

int main()

{

int n;

printf("Please input n:");

scanf("%d",&n);

int arr[9];

static int sum[100];

int num[3];

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

{

for(int j=0;j<9;j++){

scanf("%d",&arr[j]);

}

sum[i]=((arr[0]+arr[3]+arr[6])\*60\*1000+(arr[1]+arr[4]+arr[7])\*1000+

(arr[2]+arr[5]+arr[8]))/3;

}

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

num[0]=sum[i]/60000;

num[1]=sum[i]/1000%60;

num[2]=sum[i]%1000;

printf("%d %d %d\n",num[0],num[1],num[2]);

}

}

### 4.用海伦公式求三角形面积

#include<stdio.h>

int ptest(int a,int b,int c)

{

if((a+b)<c||(a+c)<b||(b+c)<a)

{

return 0;

}else if(abs(a-b)>c||abs(a-c)>b||abs(b-c)>a)

{

return 0;

}else{

return 1;

}

}

int sumArea(int a,int b,int c)

{

int p=(a+b+c)/2;

int sum=(p\*(p-a)\*(p-b)\*(p-c));

int area=sqrt(sum);

return area;

}

int main()

{

int n;

printf("Please input n:");

scanf("%d",&n);

int num1[n],num2[n],num3[n];

float tr[n];

int i=0;

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

scanf("%d %d %d",&num1[i],&num2[i],&num3[i]);

}

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

if(ptest(num1[i],num2[i],num3[i])==1){

tr[i]=(float)sumArea(num1[i],num2[i],num3[i]);

}

else{

tr[i]=0.00;

}

}

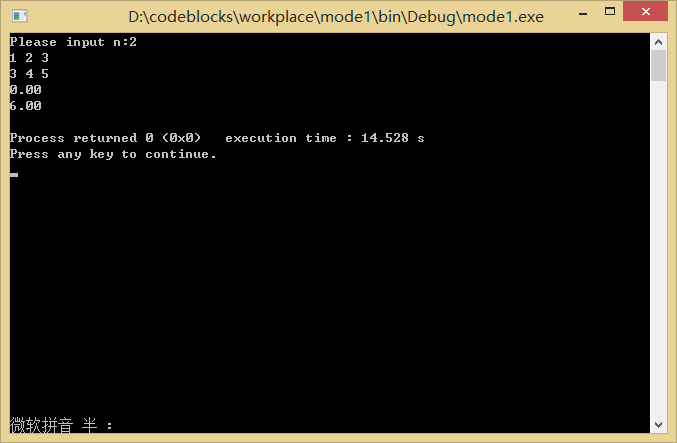
for(i=0;i<n;i++)

{

printf("%.2f\n",tr[i]);

}

}



## 2015年题目程序

### 最小数

#include<stdio.h>

int searchMinNum(int a,int b,int c)

{

if(a<b&&a<c)

{

return a;

}else if(b<a&&b<c)

{

return b;

}else if(c<a&&c<b)

{

return c;

}else if(a==b&&a==c)

{

return a;

}

}

int main()

{

int num1,num2,num3;

printf("Please input a b c:");

scanf("%d %d %d",&num1,&num2,&num3);

int num=searchMinNum(num1,num2,num3);

printf("The min num is:%d\n",num);

}

### 十进制转化为二进制

A

#include<stdio.h>

#define MAXSIZE 10000

int main()

{

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

int num;

static int total=0;

scanf("%d",&num);

char ch[MAXSIZE];

if(num<=10000){

while((num/2)!=0)

{

if((num%2)==0)

{

ch[total]='0';

}else{

ch[total]='1';

}

num=num/2;

total++;

}

ch[total]='1';

for(int i=total;i>=0;i--){

printf("%c",ch[i]);

}

}

else

{

printf("The number %d is beyond the limit.",num);

}

return 0;

}

B

#include<stdio.h>

#define MAXSIZE 10000

int main()

{

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

int num;

int total=0;

scanf("%d",&num);

char ch[MAXSIZE];

if(num<=10000){

while(num!=0)

{

if((num%2)==0)

{

ch[total]='0';

}else{

ch[total]='1';

}

total++;

num=num/2;

}

for(int i=total-1;i>=0;i--){

printf("%c",ch[i]);

}

}

else

{

printf("The number %d is beyond the limit.",num);

}

return 0;

}

C

#include<stdio.h>

int main(void){

int k,i;

int remainder[3000];

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

int num;

static int total=0;

scanf("%d",&num);

//定义了一个remainder数组，用来收集短除法除得的余数，栈倒序输出。

k=0;

while(num!=0){

k++;

remainder[k]=num%2;

num=num/2;

};

for (i=k;i>=1;i--){

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

}

return 0;

}

**栈的形式来实现**

#include<stdio.h>

#include<stdlib.h>

typedef struct SqNode

{

int elem;

struct SqNode \*next;

}SqNode;

typedef struct SqStack

{

SqNode \*base;

SqNode \*top;

}SqStack;

void systemConvert(int number,int system,SqStack \*SS);

void InitSatck(SqStack \*SS);

int PushStack(SqStack \*SS,int q);

int popStack(SqStack \*SS);

void systemConvert(int number,int system,SqStack \*SS)

{

int q; ///余数

int r; ///商

r=number/system;

q=number%system;

PushStack(SS,q);

while(r!=0)

{

q=r%system;

r=r/system;

PushStack(SS,q);

}

}

void InitStack(SqStack \*SS)

{

SS->base=(SqNode\*)malloc(sizeof(SqNode));

SS->top=SS->base;

}

int PushStack(SqStack \*SS,int e)

{

SqNode \*SN=(SqNode\*)malloc(sizeof(SqNode));

if(!SN)

{

printf("内存申请失败！\n");

return 0;

}

SN->elem=e;

SN->next=SS->top;

SS->top=SN;

return 1;

}

int PopStack(SqStack \*SS)

{

SqStack \*S=SS;

if(S->top==S->base)

{

printf("栈是空的\n");

return 0;

}

printf("%d\n",S->top->elem);

SqNode \*ss=S->top->next;

free(S->top);

S->top=ss;

return 1;

}

int main()

{

SqStack SS;

InitStack(&SS);

int number,system;

scanf("%d",&number);

scanf("%d",&system);

systemConvert(number,system,&SS);

while(SS.top!=SS.base)

{

PopStack(&SS);

}

return 0;

}

#include<stdio.h>

#include<stdlib.h>

typedef struct SqNode

{

int elem;

struct SqNode \*next;

}SqNode;

typedef struct SqStack

{

SqNode \*base;

SqNode \*top;

}SqStack;

/\*\*

这里要明白一点就是这里说明了栈的四个要素，就是数据，栈底和栈顶，还是就是指向下一个指针

\*\*/

void systemConvert(int number,int system,SqStack \*SS);

void InitSatck(SqStack \*SS);

int PushStack(SqStack \*SS,int q);

int popStack(SqStack \*SS);

void systemConvert(int number,int system,SqStack \*SS)

{

int q; ///余数

int r; ///商

r=number/system;

q=number%system;

PushStack(SS,q);

while(r!=0)

{

q=r%system;

r=r/system;

PushStack(SS,q);

}

}

///栈的初始化

void InitStack(SqStack \*SS)

{

SS->base=(SqNode\*)malloc(sizeof(SqNode));

SS->top=SS->base;

}

///压栈

int PushStack(SqStack \*SS,int e)

{

SqNode \*SN=(SqNode\*)malloc(sizeof(SqNode));

if(!SN)

{

printf("内存申请失败！\n");

return 0;

}

SN->elem=e;

SN->next=SS->top;

SS->top=SN;

return 1;

}

int PopStack(SqNode \*SS)

{

SqStack \*S = SS;

if(S->top == S->base)

{

printf("栈为空！\n");

return 0;

}

printf("%d\n",S->top->elem);

SqNode \* ss = S->top->next;

free(S->top);

S->top = ss;

return 1;

}

int main()

{

SqStack SS;

InitStack(&SS);

int number,system;

scanf("%d",&number);

scanf("%d",&system);

systemConvert(number,system,&SS);

while(SS.top!=SS.base)

{

PopStack(&SS);

}

return 0;

}

### 3.迷宫游戏 N\*N wasd的++--

#include<stdio.h>

int main()

{

int N;

int m,n;

char ch;

printf("Please input N:");

scanf("%d",&N);

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

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

printf("Please input 'w' or 's' or 'a' or 'd':");

while(1){

scanf("%c",&ch);

if(ch=='w'){

m=m-1;

if(m<=0)

{

printf("NO\n");

break;

}

else{

printf("YES\n");

}

printf("Please input 'w' or 's' or 'a' or 'd':");

}

else if(ch=='a'){

n=n-1;

if(n==0)

{

printf("NO\n");

break;

}

else{

printf("YES\n");

}

printf("Please input 'w' or 's' or 'a' or 'd':");

}

else if(ch=='s'){

n=n+1;

if(n>N)

{

printf("NO\n");

break;

}else{

printf("YES\n");

}

printf("Please input 'w' or 's' or 'a' or 'd':");

}

else if(ch=='d'){

m=m+1;

if(m>N)

{

printf("NO\n");

break;

}

else{

printf("YES\n");

}

printf("Please input 'w' or 's' or 'a' or 'd':");

}

}

}

### 迷宫游戏2

#include<stdio.h>

int main()

{

int N;

int m,n;

int ch;

printf("Please input N:");

scanf("%d",&N);

int buff[N][N];

for(int i=0;i<N;i++){

for(int j=0;j<N;j++)

{

scanf("%d",&buff[i][j]);

}

}

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

while(1){

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

if(buff[m][n]==0)

{

printf("YES\n");

printf("Please input 0,1,2,3,4:");

scanf("%d",&ch);

if(ch==0)

{

n=n-1;

if(buff[m][n]==1)

{

printf("NO");

break;

}

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

}else if(ch==1)

{

m=m+1;

if(buff[m][n]==1)

{

printf("NO");

break;

}

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

}

else if(ch==2)

{

n=n+1;

if(buff[m][n]==1)

{

printf("NO");

break;

}

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

}

else if(ch==3)

{

m=m-1;

if(buff[m][n]==1)

{

printf("NO");

break;

}

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

}

}

else

{

printf("NO");

break;

}

}

}

### 字符串的替换

/\*思路梳理、

首先是判断是不是有，找到第一个位置，把需要的替换的添加上去

最后是再以这个新的字符串来判别是不是还有，再一次有相同的操作

其实最多用到的是一个复制函数，就是把需要添加的从那个位置开始添加

\*/

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

///字符串替换函数

///ps是源，pd是子串

///函数的执行的成功与不成功，决定了就是函数的返回值是0还是1

///这里返回的还是指针

///也说明就是有没有子字符串

char \*my\_strstr(char \*ps,char \*pd)

{

char \*pt=pd;

int c=0;

while(\*ps!='\0')

{

if(\*ps==\*pd)

{

while(\*ps==\*pd &&\*pd!='\0')

{

ps++;

pd++;

c++;

}

}else

{

ps++;

}

if(\*pd=='\0')

{

return(ps-c);

}

c=0;

pd=pt;

}

return 0;

}

///复制一个内存区域到另外一个内存区域

void \*memcpy(void \*dest,const void \*src,size\_t count)

{

char \*tmp=(char \*)dest,\*s=(char \*) src;

while(count--)

\*tmp++=\*s++;

return dest;

}

///P\_source要查找的母字符串，

///P\_seach是要查找的子字符串

///P\_repstr是要替换的字符串

///p\_result是要存放结果的字符串

int str\_replace(char \*p\_result,char\* p\_source,char\* p\_seach,char \*p\_repstr)

{

int c=0;

int repstr\_leng=0;

int searchstr\_leng=0;

char \*p1;

char \*presult=p\_result;

char \*psource=p\_source;

char \*prep=p\_repstr;

char \*psearch=p\_seach;

int nLen=0;

repstr\_leng=strlen(prep);

searchstr\_leng=strlen(psearch);

printf("The result:%s\r\n",p\_result);

do{

p1=my\_strstr(psource,p\_seach);

if(p1==0)

{

strcpy(presult,psource);

// printf("The result:%s\r\n",p\_result);

return c;

}

c++;///匹配的子串加1

printf("The source char is:%s\r\n",p\_source);

/// 拷贝上一个替换点和下一个替换点中间的字符串

nLen=p1-psource;

memcpy(presult,psource,nLen);

printf("The result:%s\r\n",p\_result);

///拷贝需要替换的字符串

memcpy(presult+nLen,p\_repstr,repstr\_leng);

psource=p1+searchstr\_leng;

presult=presult+nLen+repstr\_leng;

}while(p1);

return c;

}

#define MAX 200

int main(void)

{

int i = 0;

char s[MAX] ={0}; //存放源字串

char s1[MAX]={0}; //存放子字串

char s2[MAX]={0}; //存放替换字串

char result\_a[2000] = {0};//存放替换结果;

char \*p,\*ptm,\*pr;

puts("Please input the string for s:");

scanf("%s",s);

puts("Please input the string for s1:");

scanf("%s",s1);

puts("Please input the string for s2:");

scanf("%s",s2);

ptm = s;

pr = result\_a;

i = str\_replace(pr,ptm,s1,s2);

printf("替换%d个子字符串;\r\n",i);

printf("替换后结果:%s\r\n",result\_a);

system("pause");

}

### 5.六数码问题

#include <stdio.h>

long turna(long n);

long turnb(long n);

int find(int a[10]);

long turna(long n)

{

int m=0,i=1;

int a[7]={0};

while(n>0)

{

a[i]=n%10;

i++;

n=n/10;

}

m=a[3]\*100000+a[6]\*10000+a[4]\*1000+a[2]\*100+a[5]\*10+a[1];

return(m);

}

long turnb(long n)

{

int m=0,i=1;

int a[7]={0};

while(n>0)

{

a[i]=n%10;

i++;

n=n/10;

}

m=a[6]\*100000+a[2]\*10000+a[5]\*1000+a[3]\*100+a[1]\*10+a[4];

return(m);

}

int find(int a[10])

{

int t[66666]={0};

long n=0,m=1,i;

long d[10000]={0};

int s,e;

for(i=1;i<=6;i++)

{

n=n+a[i]\*m;

m=m\*10;

}

s=0;e=1;

t[n/10]=1;

d[s]=n;

i=0;

while(s<e)

{

n=d[s];

if(n==123456)

{

i=1;

break;

}

m=turna(n);

if(t[m/10]==0)

{

t[m/10]=1;

d[e]=m;

e++;

}

m=turnb(n);

if(t[m/10]==0)

{

t[m/10]=1;

d[e]=m;

e++;

}

s++;

}

return(i);

}

int main()

{

int a[10]={0};

while(scanf("%d%d%d%d%d%d",&a[6],&a[5],&a[4],&a[3],&a[2],&a[1])!=EOF)

{

if(find(a)==1)

printf("Yes\n");

else

printf("No\n");

}

return 0;

}

### 7 输入命题公式的真值表

### 8 计算单词的个数

# include <stdio.h>

int main(void)

{

printf("Please input a word:\n");

char str[100];

///在这里明白gets();和scanf()的区别

///gets()：获取一行数据，并作为字符串处理。

///scanf();输入的遇到空格的时候就要不可以了

gets(str);

int length=strlen(str);

int count=0;

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

{

if(str[i]==' ')

count++;

}

count++;

printf("%d",count);

}

## 2014年机试题目程序

### 2 猴子挖花生

#include <stdio.h>

#include<stdlib.h>

/\*\*这里的思路是计算就是每一次到达下一个或者是回来的差值的绝对值的和

就可以算出她所走的路的全部的长度\*/

int main()

{

int N;

printf("Please input a number N:\n");

scanf("%d",&N);

int m[N],n[N];

int x=0,y=0,sum=0;

///花生的位置的输入

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

{

scanf("%d %d",&m[i],&n[i]);

}

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

{

sum=sum+abs(m[i]-x);

sum=sum+abs(n[i]-y);

x=m[i];

y=n[i];

}

sum=sum+abs(x-0);

sum=sum+abs(y-0);

printf("%d",sum);

return 0;

}

## 自己练习题目

### 贪吃蛇

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<conio.h>

typedef struct snake

{

int a;

int b;

struct snake \*u;

struct snake \*n;

}snake,\*snake1;

typedef struct food

{

int a;

int b;

}food;

void main()

{

char c,c0 = 'd';

int i,j,k,n=1,t,at;

snake p,q;

snake \*dd,\*dd0,\*dd1,\*dd2;

food f;

srand(time(NULL));

p.u = NULL;

p.n = &q;

p.a = 5;p.b = 6;q.a = 5;q.b = 5;

q.u = &p;q.n = NULL;

dd=dd2= &q;

f.a=(rand()%15+1);

f.b=(rand()%15+1);

while(1)

{

srand(time(NULL));

system("cls");

for(i = 0;i < 17;i ++)

{

for(j = 0; j < 17;j++)

{

if(i == 0 )

printf("▁");

else if(i == 16)

printf("▔");

else if(j == 0)

printf("▕");

else if(j == 16)

printf("▏");

else if(i == p.a && j == p.b)

printf("■");

else if(i == f.a && j == f.b)

printf("★");

else

{

t = 0;

dd = dd2;

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

{

if(i == dd->a && j == dd->b)

{

printf("□");

t = 1;

break;

}

dd = dd->u;

}

if(t == 0)

printf(" ");

}

}printf("\n");

}

at = 0;

dd =dd2;

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

{

if(p.a == dd->a && p.b == dd->b)

{

printf("game over!!\n");

exit(0);

}

dd = dd->u;

}

if(p.a == f.a && p.b == f.b)

{

dd = dd2;

at =1;

f.a = (rand()%15+1);

f.b = (rand()%15+1);

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

{

if(f.a == dd->a && f.b == dd->b)

{

f.a = dd2->a;

f.b = dd2->b;

break;

}

}

n++;

}

if(kbhit())

{

c = getch();

dd = dd2;

if(c == 'w' && c0 != 's')

{

if(at == 1)

{

dd0 =(snake1)malloc(sizeof(snake));

dd0->a = dd2->a;dd0->b = dd2->b;

dd0->n = NULL;dd0->u = dd2;

dd2=dd0;

}

dd = dd2;

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

{

dd1 = dd->u;

dd->b = dd1->b;

dd->a = dd1->a;

dd = dd->u;

}

if(p.a == 1)

p.a = 15;

else

p.a = (p.a-1)%15;

}

else if(c == 's' && c0 != 'w')

{

if(at == 1)

{

dd0 =(snake1)malloc(sizeof(snake));

dd0->a = dd2->a;dd0->b = dd2->b;

dd0->n = NULL;dd0->u = dd2;

dd2=dd0;

}

dd = dd2;

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

{

dd1 = dd->u;

dd->b = dd1->b;

dd->a = dd1->a;

dd = dd->u;

}

p.a = (p.a%15)+1;

}

else if(c == 'a' && c0 != 'd')

{

if(at == 1)

{

dd0 =(snake1)malloc(sizeof(snake));

dd0->a = dd2->a;dd0->b = dd2->b;

dd0->n = NULL;dd0->u = dd2;

dd2=dd0;

}

dd = dd2;

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

{

dd1 = dd->u;

dd->b = dd1->b;

dd->a = dd1->a;

dd = dd->u;

}

if(p.b == 1)

p.b = 15;

else

p.b = (p.b-1)%15;

}

else if(c == 'd' && c0 != 'a')

{

if(at == 1)

{

dd0 =(snake1)malloc(sizeof(snake));

dd0->a = dd2->a;dd0->b = dd2->b;

dd0->n = NULL;dd0->u = dd2;

dd2=dd0;

}

dd = dd2;

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

{

dd1 = dd->u;

dd->b = dd1->b;

dd->a = dd1->a;

dd = dd->u;

}

p.b = (p.b%15)+1;

}

else

{

goto qq;

}

c0 = c;

}

else

{

qq: if(c0 == 'w')

{

if(at == 1)

{

dd0 =(snake1)malloc(sizeof(snake));

dd0->a = dd2->a;dd0->b = dd2->b;

dd0->n = NULL;dd0->u = dd2;

dd2=dd0;

}

dd = dd2;

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

{

dd1 = dd->u;

dd->b = dd1->b;

dd->a = dd1->a;

dd = dd->u;

}

if(p.a == 1)

p.a = 15;

else

p.a=(p.a-1)%15;

}

else if(c0 == 's')

{

if(at == 1)

{

dd0 =(snake1)malloc(sizeof(snake));

dd0->a = dd2->a;dd0->b = dd2->b;

dd0->n = NULL;dd0->u = dd2;

dd2=dd0;

}

dd = dd2;

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

{

dd1 = dd->u;

dd->b = dd1->b;

dd->a = dd1->a;

dd = dd->u;

}

p.a=(p.a%15)+1;

}

else if(c0 == 'a')

{

if(at == 1)

{

dd0 =(snake1)malloc(sizeof(snake));

dd0->a = dd2->a;dd0->b = dd2->b;

dd0->n = NULL;dd0->u = dd2;

dd2=dd0;

}

dd = dd2;

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

{

dd1 = dd->u;

dd->b = dd1->b;

dd->a = dd1->a;

dd = dd->u;

}

if(p.b == 1)

p.b = 15;

else

p.b=(p.b-1)%15;

}

else if(c0 == 'd')

{

if(at == 1)

{

dd0 =(snake1)malloc(sizeof(snake));

dd0->a = dd2->a;dd0->b = dd2->b;

dd0->n = NULL;dd0->u = dd2;

dd2=dd0;

}

dd = dd2;

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

{

dd1 = dd->u;

dd->b = dd1->b;

dd->a = dd1->a;

dd = dd->u;

}

p.b=(p.b%15)+1;

}

}

fflush(stdin);

dd = &q;

\_sleep(200);

}

}

### 计算机

#include <stdio.h>

#include <math.h>

#include<malloc.h>

void jingtailianbiao();

void dongtailianbiao();

int stu\_num;

#define LEN sizeof(struct student)

struct student

{

long num;

float score;

struct student \*next;

};

struct student\_score

{

long num;

char name[10];

float score;

};

int main()

{

int select;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*这是一个计算器\*\*\*\*\*\*\*\*\*\*\n");

printf("\* 1 加法 \*\n");

printf("\* 2 减法 \*\n");

printf("\* 3 乘法 \*\n");

printf("\* 4 除法 \*\n");

printf("\* 5 取模 \*\n");

printf("\* 6 圆的面积 \*\n");

printf("\* 7 判断闰年 \*\n");

printf("\* 8 成绩转化 \*\n");

printf("\* 9 计算从1到输入数之间的奇数和(for) \*\n");

printf("\* 10 计算从1到输入数之间的偶数和(while) \*\n");

printf("\* 11 输出从1到输入数之间所有3的倍数 \*\n");

printf("\* 12 求即被3整除又被5整除的数 \*\n");

printf("\* 13 判断数字位数 \*\n");

printf("\* 14 求水仙花数 \*\n");

printf("\* 15 求最大值 \*\n");

printf("\* 16 求最大最小值 \*\n");

printf("\* 17 冒泡排序 \*\n");

printf("\* 18 矩阵转置 \*\n");

printf("\* 19 求对角线和 \*\n");

printf("\* 20 统计单词个数 \*\n");

printf("\* 21 静态创建链表 \*\n");

printf("\* 22 动态创建链表 \*\n");

printf("\* 23 学生成绩排序 \*\n");

printf("\* 24 学生成绩普涨10分 \*\n");

printf("\* 输入任意非功能选项序号退出小小计算器 \*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("输入菜单项：\n");

scanf("%d",&select);

switch (select)

{

case 1:

{

float num1,num2;

printf("做加法。\n");

printf("输入两个数，逗号分隔。\n");

scanf("%f,%f",&num1,&num2);

printf("%f+%f=%f\n",num1,num2,num1+num2);

break;

}

case 2:

{

float num1,num2;

printf("做减法。\n");

printf("输入两个数，逗号分隔。\n");

scanf("%f,%f",&num1,&num2);

printf("%f-%f=%f\n",num1,num2,num1-num2);

break;

}

case 3:

{

float num1,num2;

printf("做乘法。\n");

printf("输入两个数，逗号分隔。\n");

scanf("%f,%f",&num1,&num2);

printf("%f\*%f=%f\n",num1,num2,num1\*num2);

break;

}

case 4:

{

float num1,num2;

printf("做除法。\n");

printf("输入两个数，逗号分隔。\n");

scanf("%f,%f",&num1,&num2);

printf("%f/%f=%f\n",num1,num2,num1/num2);

break;

}

case 5:

{

float num1,num2;

printf("取模运算。\n");

printf("输入两个数，逗号分隔。\n");

scanf("%f,%f",&num1,&num2);

printf("%f%%%f=%f\n",num1,num2,fmod(num1,num2));

break;

}

case 6:

{

float r;

double s;

printf("计算圆面积。\n");

printf("请输入半径：\n");

scanf("%f",&r);

if (r >= 0)

{

s=3.14\*r\*r;

printf("圆面积是：%lf\n",s);

}

else

printf ("半径输入错误！\n");

break;

}

case 7:

{

int year,leap;

printf("判断闰年。\n");

printf("enter year");

scanf("%d",&year);

if((year%4==0&&year%100!=0)||(year%400==0))

leap=1;

else

leap=0;

if(leap)

printf("%d is",year);

else

printf("%d is not",year);

printf("a leap year\n");

break;

}

case 8:

{

int a;

printf("成绩转化。\n");

scanf("%d",&a);

if(a>=90)

printf("优");

else if(a>=80)

printf("良");

else if(a>=70)

printf("中");

else if(a>=60)

printf("差");

else printf("不及格");

break;

}

case 13:

{

long int num,num\_t;

int digits=0,i=1;

printf("判断数字位数。\n");

printf("输入待判断数字：");

scanf("%ld",&num);

num\_t=num;

do

digits++;

while((num\_t/=10)>0);

printf("%ld是%d位数。\n",num,digits);

while(digits>=i);

{

printf("%ld的第%d位数是：%d\n",num,i,(num/(long)pow(10,(digits-i)))%10);

i++;

}

break;

}

case 9:

{

int n,sum=0,i;

printf("输入一个数：\n");

scanf("%d",&n);

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

if(i%2!=0)

sum+=i;

printf("从1到输入数之间的奇数和是：%d\n",sum);

break;

}

case 10:

{

int sum,i,n;

i=1;

sum=0;

printf("输入一个数：\n");

scanf("%d",&n);

while(i<=n)

{

if(i%2==0)

sum+=i;

i++;

}

printf("从1到输入数之间的偶数和是：%d\n",sum);

break;

}

case 12:

{

int n,i,m;

printf("输入两个数：\n");

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

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

if(i%3==0&&i%5==0)

{

printf("既能被3整除又能被5整除的数：%d\n",i);

}

break;

}

case 11:

{

int sum,i,m;

i=1;

sum=0;

printf("输入一个数：\n");

scanf("%d",&m);

do

{

if(i%3==0)

sum+=i;

i++;

}

while(i<=m);

printf("从1到输入数之间所有3的倍数：%d\n",sum);

break;

}

case 14:

{

int i=100;

int a,b,c;

for(i=100;i<1000;i++)

{

a=i/100;

b=(i-a\*100)/10;

c=(i-a\*100-b\*10);

if(a\*a\*a+b\*b\*b+c\*c\*c==i)

{

printf("水仙花数：%d\n",i);

}

}

break;

}

case 15:

{

int a,b,c;

printf("请输入两个数\n");

scanf("%d,%d",&a,&b);

if(a>=b)

c=a;

else

c=b;

printf("最大值为：%d\n",c);

break;

}

case 16:

{

int a,b,c,d;

printf("请输入两个数\n");

scanf("%d,%d",&a,&b);

if(a>=b)

{

c=a;

d=b;

}

else

{

c=b;

d=a;

}

printf("最大值为：%d\n",c);

printf("最小值为：%d\n",d);

break;

}

case 17:

{

int i,j,k;

int a[10];

printf("输入十个数： \n");

for(i=0;i<10;i++)

scanf("%d",&a[i]);

printf("\n");

for(j=0;j<9;j++)

for(i=0;i<9-j;i++)

if(a[i]>a[i+1])

{

k=a[i];

a[i]=a[i+1];

a[i+1]=k;

}

printf("排序后： \n");

for(i=0;i<10;i++)

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

printf("\n");

break;

}

case 18:

{

int i,j;

int array[4][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};

int temp;

printf("原矩阵：\n");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

printf("%d\t",array[i][j]);

}

printf("\n");

}

for(i=0;i<4;i++)

{

for(j=i;j<4;j++)

{

temp=array[i][j];

array[i][j]=array[j][i];

array[j][i]=temp;

}

}

printf("转置矩阵：\n");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

printf("%d\t",array[i][j]);

}

printf("\n");

}

break;

}

case 19:

{

int i,j;

int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};

int sum=0;

printf("原矩阵：\n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

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

}

sum+=a[i][i];

printf("\n");

}

printf("对角线的和等于：%d\n",sum);

break;

}

case 20:

{

char string[81],c;

int i,num=0,word=0;

printf("请输入一句话:\n");

c=getchar();

gets(string);

for(i=0;(c=string[i])!='\0';i++)

{

if(c==' ')

{

word=0;

}

else if(word==0)

{

word=1;

num++;

}

}

printf("%d words\n",num);

break;

}

case 21:

{

jingtailianbiao();

break;

}

case 22:

{

printf("请输入两个数，逗号隔开。输入负数结束。\n");

dongtailianbiao();

break;

}

case 23:

{

float sum=0;

struct student\_score stu\_temp,stu[10];

printf("分别输入10位学生的学号姓名和成绩 ，用空格隔开。\n");

for (int i=0;i<10;i++)

{

scanf("%d %s %f",&stu[i].num,&stu[i].name,&stu[i].score);

sum+=stu[i].score;

}

for (int i=0;i<9;i++)

for (int j=0;j<9-i;j++)

if (stu[j].score>stu[j+1].score)

{

stu\_temp = stu[j];

stu[j] = stu[j+1];

stu[j+1] = stu\_temp;

}

printf("按成绩从低到高排序为:\n");

for (int i=0;i<10;i++)

{

printf("学号:%d\t姓名:%s\t成绩:%.1f\n",stu[i].num,stu[i].name,stu[i].score);

}

printf("成绩总和为:%.1f\n",sum);

break;

}

case 24:{

printf("将为d盘根目录下的文本文件my.txt内的10位学生的学生成绩分别加10分,请确保此文件存在！\n");

int i,a[10];

FILE \*fp;

fp=fopen("d:\\my.txt","r+");

for(i=0;i<10;i++)

fscanf(fp,"%d",&a[i]);

rewind(fp);

for(i=0;i<10;i++)

fprintf(fp,"%d\n",a[i]+10);

fclose(fp);

printf("保存完毕！\n");

break;

}

}

return 0 ;

}

void jingtailianbiao()

{

struct student a,b,c,d,\*head,\*p;

a.num=1;

a.score=96;

b.num=2;

b.score=99.5;

c.num=3;

c.score=96;

d.num=4;

d.score=98;

head=&a;

a.next=&b;

b.next=&c;

c.next=&d;

d.next=0;

p=head;

do

{

printf("%ld %5.1f\n",p->num,p->score);

p=p->next;

}

while(p!=0);

}

struct student \* creat()

{

struct student \*head;

struct student \*p1,\*p2;

stu\_num=0;

p1=p2=(struct student\*)malloc(LEN);

scanf("%ld,%f",&p1->num,&p1->score);

head=0;

while(p1->num>=0)

{

stu\_num=stu\_num+1;

if(stu\_num==1)

head=p1;

else

p2->next=p1;

p2=p1;

p1=(struct student\*)malloc(LEN);

scanf("%ld,%f",&p1->num,&p1->score);

}

p2->next=0;

return (head);

}

void print(struct student \*head)

{

struct student \*p;

p=head;

printf("你输入的动态链表为：\n",stu\_num);

if(head!=0)

do

{

printf("%ld %5.1f\n",p->num,p->score);

p=p->next;

}

while(p!=0);

}

void dongtailianbiao()

{

struct student \*link;

link=creat();

print(link);

}

### 算e的值

：自然底数 e=2.718281828…，e 的计算公式如下：  
e=1+1/1!+1/2!+1/3!+…  
要求当最后一项的值小于 10-10 时结束。

#include<stdio.h>

int main()

{

float e=1.0,n=1.0;

int i=1;

while(1/n>1e-10)

{

e=e+1/n;

i++;

n=i\*n; ///这里的求阶乘的值

}

printf("e=%f",e);\*/

}

### 给出素数（1——1000）

#include<stdio.h>

int sushu(int num)

{

if(num<=1)

return 0;

if(num==2)

return 1;

for(int i=2;i<num;i++)

{

if(num%i==0)

return 0;

///如果不加这一句话就是只判断一次

else if(num!=i+1)

continue;

else

return 1;

}

}

int main()

{

int i;

for(i=0;i<1000;i++)

{

if(sushu(i)==1)

{

printf("%d\t",i);

}

}

}

### 矩阵变换

#include<stdio.h>

#include<stdlib.h>

int main()

{

int a[101][101],b[101][101];

int m,n;

printf("please input the number of rows(<=100)\n");

scanf("%d",&m);

printf("please input the number of columns(<=100)\n");

scanf("%d",&n);

printf("please input the element\n");

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

for(int j=0;j<n;j++)

scanf("%d",&a[i][j]);

printf("array a：\n");

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

{

for(int j=0;j<n;j++)

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

printf("\n");

}

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

{

for(int j=0;j<n;j++)

{

b[j][i]=a[i][j];

}

}

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

{

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

{

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

}

printf("\n");

}

}

### 狼追兔子

#include <stdio.h>

#include<stdlib.h>

int main()

{

int i;

int pos[10]={0};

int lang=0;

for(i=0;i<100;i++)

{

pos[lang]=1;

lang++;

lang+=i;

lang=lang%10; ///兔子的位置要知道，就是循环

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*%d\t",lang);

}

for(i=0;i<10;i++)

if(pos[i]==0)

printf("兔子可能在第%d洞中\n",i+1);

return 0;

}

### 循环链表实现约瑟夫环

#include <stdio.h>

#include<stdlib.h>

typedef struct node

{

int number;

struct node\*next;

}person;

///这个链表没有带头节点

person \*initLink(int n)

{

person \*head=(person \*)malloc(sizeof(person));

head->number=1;

head->next=NULL;

person \*cyclic=head;

for(int i=2;i<=n;i++)

{

person\*body=(person\*)malloc(sizeof(person));

body->number=i;

body->next=NULL;

cyclic->next=body;

cyclic=cyclic->next;

}

cyclic->next=head; ///首尾相连

return head;

}

void findAndKillK(person \*head,int k,int m)

{

person \*tail=head;

///第一个节点的上一个节点

while(tail->next!=head)

{

tail=tail->next;

}

person \*p=head;

while(p->number!=k)

{

tail=p;

p=p->next;

}

while(p->next!=p)

{

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

{

tail=p;

p=p->next;

}

tail->next=p->next;

free(p);

p=tail->next;

}

printf("最后留下的是:%d",p->number);

free(p);

}

int main()

{

printf("输入圆桌上的人数n:");

int n;

scanf("%d",&n);

person \* head=initLink(n);

printf("从第k人开始报数(k>1且k<%d)：",n);

int k;

scanf("%d",&k);

printf("数到m的人出列：");

int m;

scanf("%d",&m);

findAndKillK(head, k, m);

return 0;

}

### 数字的倒序输出

普通方法：

#include<stdio.h>

#include<stdlib.h>

int main()

{

int ch[100];

int num;

int count=0;

scanf("%d",&num);

while((num/10)!=0)

{

ch[count]=num%10;

printf("%d",ch[count]);

num=num/10;

count++;

}

ch[count]=num%10;

printf("%d",ch[count]);}

递归的思想：

#include <stdio.h>

int convert(char s[],int n)

{

int i;

if((i=n/10)!=0)

convert(s+1,i);///在加一的过程里面，就是把前面的放到了后面

\*s=n%10+'0';

return 0;

}

int main()

{

int num;

char str[10]=" ";

printf("input integer data：");

scanf("%d",&num);

convert(str,num);

printf("output string：\n");

puts(str);

return 0;

}

### 输出连续的字母

#include<stdio.h>

#define MAXSIZE 1000

int main()

{

char arr[MAXSIZE];

scanf("%s",arr);

int len,i;

len=strlen(arr);

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

{

if(arr[i]+1==arr[i+1]&&arr[i]+2==arr[i+2])

printf("%c%c%c\n",arr[i],arr[i+1],arr[i+2]);

}

}

#include <stdio.h>

#include <stdlib.h>

#define LINE 1024

int count(char buff[])

{

int len=strlen(buff);

int i,num=0;

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

{

if(buff[i]==' ')

{

num++;

}

}

return num+1;

}

int main()

{

FILE \*fp;

char buff[LINE];

int i=0;

int con=0;

fp=fopen("D:\\test.txt","r");

for(i=0;i<2;i++)

{

fgets(buff,LINE,fp);

con=con+count(buff);

}

printf("%d",con);

}