**c语言第五次作业**

**自81 刘泓尊 2018011446**

1. 自然数的计算

#include<stdio.h>

#include<math.h>

int LCM(int x,int y)

{

if(x<y){int tmp=x;x=y;y=tmp;}

while(y)

{

int t=x%y;

x=y;

y=t;

}

return x;

}

int GCM(int x,int y)

{

return (x\*y/LCM(x,y));

}

int main()

{

int m,n;

printf("请输入两个整数m,n：");

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

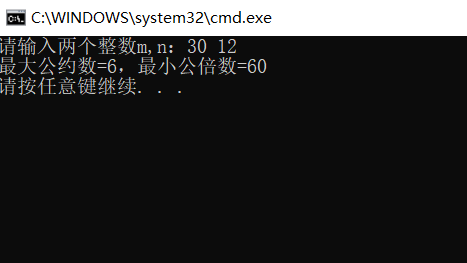
if(m<=0||n<=0)printf("Error! wrong input!");

else

printf("最大公约数=%d，最小公倍数=%d\n",LCM(m,n),GCM(m,n));

return 0;

}



1. 字符串连接

#include<stdio.h>

#include<string.h>

void mystrins(char a[],char b[],int m)

{

char c[120];

int len1=strlen(a),len2=strlen(b);

int i;

if(m>len1)

{

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

c[i]=a[i];

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

c[len1+i]=b[i];

c[len1+i]='\0';

printf("在str1后插入str2):\n");

}

else if(m<0)

{

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

c[i]=b[i];

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

c[len2+i]=a[i];

c[len2+i]='\0';

printf("在str1前连接str2):\n");

}

else if(m>=0&&m<=len1)

{

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

c[i]=a[i];

for(i=m;b[i-m]!='\0';i++)

c[i]=b[i-m];

for(i=m;a[i]!='\0';i++)

c[len2+i]=a[i];

c[len2+i]='\0';

printf("在str1的第%d位插入str2):\n",m);

}

puts(c);

}

int main()

{

int m;

char str1[60],str2[60];

printf("请输入字符串1：\n");

gets\_s(str1);

printf("请输入字符串2：\n");

gets\_s(str2);

printf("请输入插入位置：");

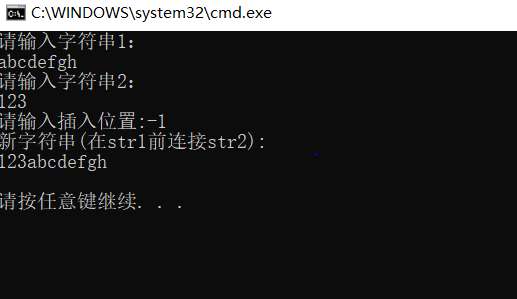
scanf\_s("%d",&m);

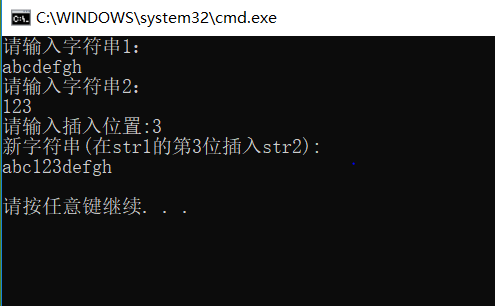
printf("\n新字符串(");

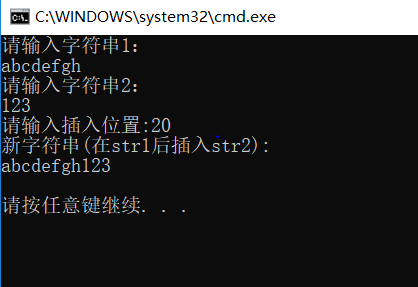
mystrins(str1,str2,m);

return 0;

}







1. 三角函数

#include<stdio.h>

#include<math.h>

double mysin(double x)

{

int i;

double sum=x,tmp=x,total=1.0,t=tmp/total,flag=1.0;

for(i=3;fabs(t)>1e-6;i+=2)

{

flag\*=-1;

total=total\*i\*(i-1);

tmp=tmp\*x\*x;

t=tmp/total;

sum+=flag\*t;

}

return sum;

}

double mycos(double x)

{

int i;

double t=1.0,sum=1.0,tmp=1.0,total=1.0,flag=1.0;

for(i=2;fabs(t)>1e-6;i+=2)

{

flag\*=-1;

total=total\*i\*(i-1);

tmp=tmp\*x\*x;

t=tmp/total;

sum+=flag\*t;

}

return sum;

}

int main()

{

double x,y1,y2,y3,y4;

double a=3.1415926/3,b=3.1415926/4;

y1=sqrt((1-mycos(a))/2);

y2=2\*mysin(b)\*mysin(b)+mysin(b)\*mycos(b)-mycos(b)\*mycos(b);

printf("自定义函数:\ny1=%f\ny2=%f\n",y1,y2);

y3=sqrt((1-cos(a))/2);

y4=2\*sin(b)\*sin(b)+sin(b)\*cos(b)-cos(b)\*cos(b);

printf("标准函数：\ny1=%f\ny2=%f\n",y3,y4);

double d1,d2;

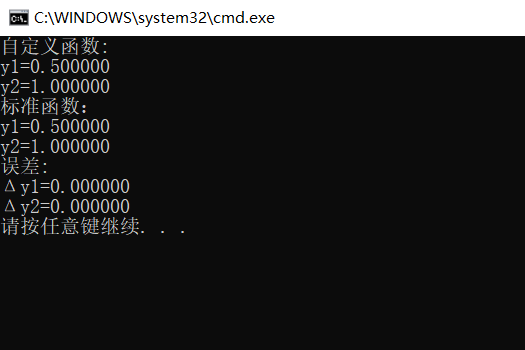
d1=fabs(y1-y3);

d2=fabs(y2-y4);

printf("误差:\nΔy1=%f\nΔy2=%f\n",d1,d2);

return 0;

}



1. 弦截法解方程

#include<stdio.h>

#include<math.h>

float f(float x){return(x\*x\*x-5\*x\*x+16\*x-80);}

float xpoint(float x1,float x2)

{

return((x2\*f(x1)-x1\*f(x2))/(f(x1)-f(x2)));

}

float root(float x1,float x2)

{

float x0;

do

{

x0=xpoint(x1,x2);

if(f(x0)\*f(x1)<0)x2=x0;

else x1=x0;

}while(fabs(f(x0))>1e-3);

return x0;

}

int main()

{

float x1,x2;

printf("输入区间端点：\n");

scanf\_s("%f%f",&x1,&x2);

if(x1>x2){float tmp=x1;x1=x2;x2=tmp;}

if(f(x1)\*f(x2)>0)printf("可能是无零点的区间！\n");

else

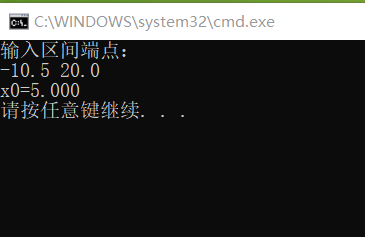
{

printf("x0=%.3f\n",root(x1,x2));

}

return 0;

}



1. 货币计算

#include<stdio.h>

#include<math.h>

void tran(float x,int a[])

{

int m=x\*100;

a[1]=m/10000;

if(m%10000>5000)

{

a[2]=(m%10000)/5000;

a[3]=(m%10000-5000)/1000;

}

else a[3]=(m%10000)/1000;

int tmp1=m-a[1]\*10000-a[2]\*5000-a[3]\*1000;

a[4]=tmp1/100;

if(tmp1%100>50)

{

a[5]=(tmp1%100)/50;

a[6]=(tmp1%100-50)/10;

}

else a[6]=(tmp1%100)/10;

int tmp2=tmp1%100-50\*a[5]-10\*a[6];

if(tmp2>5)

{

a[7]=1;

a[8]=tmp2-5;

}

else a[8]=tmp2;

}

int main()

{

float money;

int num[9]={0};

printf("请输入货币金额(单位：元)(可带两位小数):");

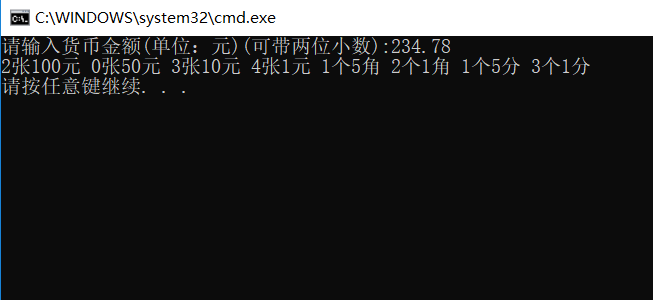
scanf\_s("%f",&money);

tran(money,num);

printf("%d张100元 %d张50元 %d张10元 %d张1元 %d个5角 %d个1角 %d个5分 %d个1分\n",num[1],num[2],num[3],num[4],num[5],num[6],num[7],num[8]);

return 0;

}



1. 角夫猜想

#include<stdio.h>

void f(int a)

{

while(a!=1)

{

printf("%d ",a);

if(a%2)a=3\*a+1;

else a/=2;

}

printf("1");

}

int main()

{

int a;

while(1)

{

printf("\nplease input a number:");

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

if(a==0)break;

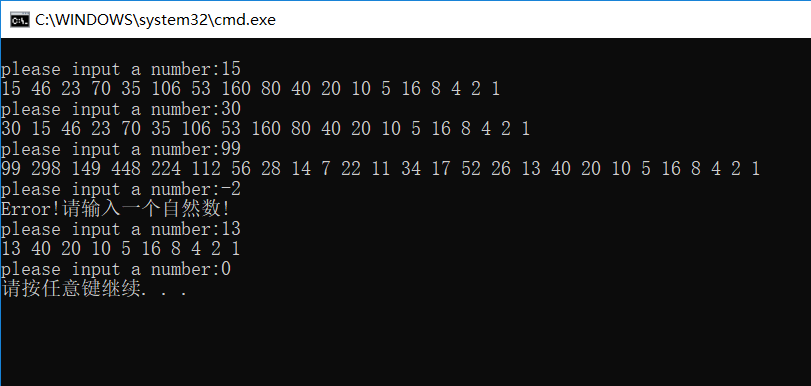
if(a<0)printf("Error!请输入一个自然数!");

if(a>0)f(a);

}

return 0;

}



1. 数学黑洞

#include<stdio.h>

#include<math.h>

void rank(int a,int\*p)

{

int b[5],i,j;

b[1]=a/1000;

b[2]=(a/100)%10;

b[3]=(a%100)/10;

b[4]=a%10;

for(i=4;i>1;i--)//找到最大排列和最小排列

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

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

{

int tmp=b[j];

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

b[j+1]=tmp;

}

p[1]=b[1]\*1000+b[2]\*100+b[3]\*10+b[4];

p[2]=b[4]\*1000+b[3]\*100+b[2]\*10+b[1];

p[3]=p[1]-p[2];

//输出三个数，数位不足左补零

printf("(%04d,%04d,%04d)\n",p[1],p[2],p[3]);

}

int main()

{

int a[5],tmp;

while(1)

{

printf("please input a number(except 'xxxx'):");

scanf\_s("%d",&a[3]);

if(a[3]==0)break;

else if(a[3]/1000==0||a[3]/1000>9)printf("Error!please input

again!\n");

else{

while(1)

{

tmp=a[3];

rank(a[3],a);

if(a[3]==tmp)break;

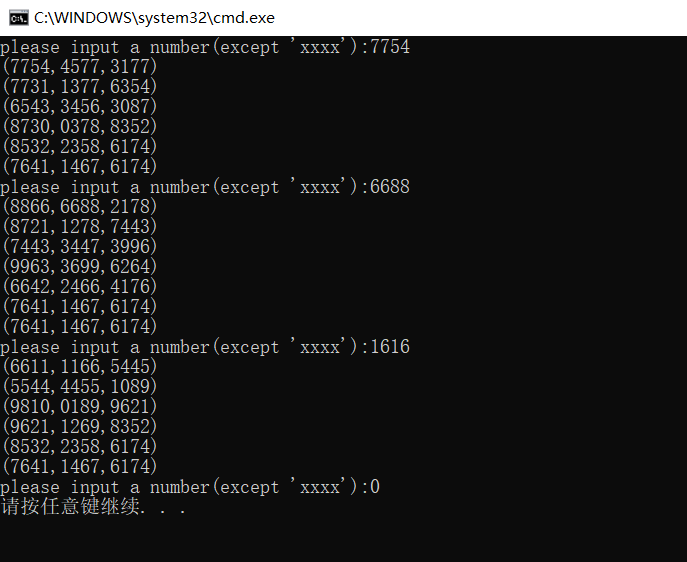
}

}

}

return 0;

}



1. 勒让德多项式

#include<stdio.h>

double legendre(int n,double x)

{

if(n==0)return 1;

else if(n==1)return (x);

else return (((2\*n-1)\*x\*legendre(n-1,x)-(n-1)\*legendre(n-2,x))/n);

}

int main()

{

int n;

double x,s;

while(1)

{

printf("please input the value of n and x:");

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

if(n<0){

printf("Error!Try again!\n\n");

fflush(stdin);

}

else{

s=legendre(n,x);

printf("y=%.3f\n",s);

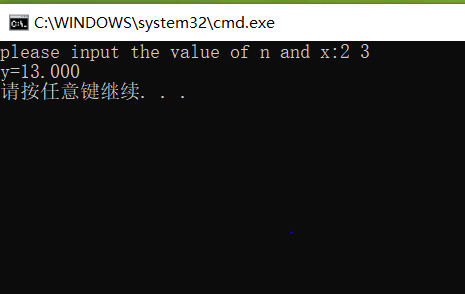
break;

}

}

return 0;

}



1. 设计递归程序

#include <stdio.h>

#include <string.h>

int main()

{

void outputstr(char string[],int n,int len);

char str[100];

int tmp,len;

while(1)

{

printf("please input a string：\n");

gets\_s(str);

len=strlen(str);

if(len<5||len>=100)//不符合条件的处理

{

printf("Error!please input a string again!\n");

}

else{

printf("the new string is:\n");

tmp=len-1;

outputstr(str,tmp,len-1);

printf("\n");

break;

}

}

return 0;

}

void outputstr(char str[],int n,int len)

{

int i=n;

if(i>0)

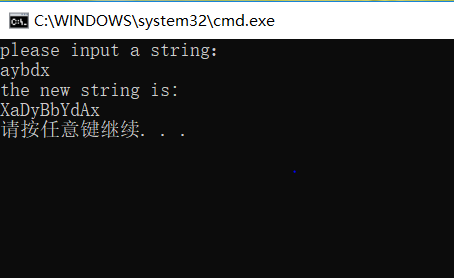
{

outputstr(str,n-1,len);

}

printf("%c%c",str[len-i]-32,str[i]);

}



1. 用递归程序实现

//递归实现

#include<stdio.h>

//x表示已经选了x个，s表示从第s个自然数开始考虑。

void permutation(int x,int s,int n,int r)

{

for(int i=s;n-i>=r-x;i++)

{

if(i!=s)

for(int j=1;j<x;j++)printf(" ");

printf("%4d",i);

if(x==r)putchar('\n');

else permutation(x+1,i+1,n,r);

}

}

int main()

{

int n,r;

while(1){

printf("Enter N R:");

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

if(n<1||n>1000||r>n||r<1)

{

printf("Error!please input again!\n");

}

else

{

printf("combinations:\n");

permutation(1,1,n,r);

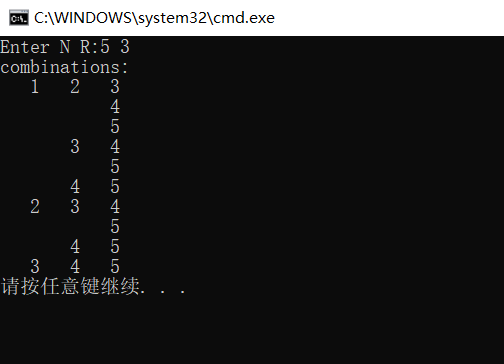
break;

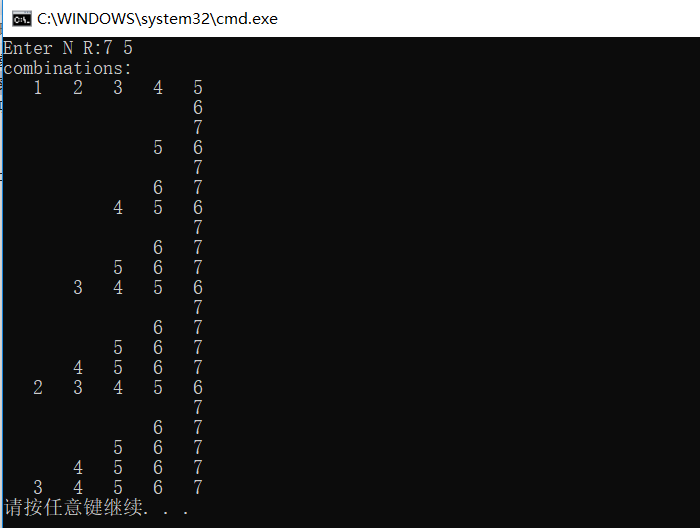
}

}

return 0;

}





1. 子串查找

#include<stdio.h>

#include<string.h>

void strsearch(char a[],char b[],int\*s)

{

int len2=strlen(b);

int i,j,flag;

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

if(a[i]==b[0])

{

flag=1;

for(j=i+1;j<i+len2;j++)

if(a[j]!=b[j-i])

{

flag=0;

break;

}

if(flag)s[1]++;

if(s[1]==1)s[0]=i+1;

}

}

int main()

{

char a[100],b[100];

int result[2]={-1,0};

printf("please input str1:\n");

gets\_s(a);

printf("please input str2:\n");

gets\_s(b);

strsearch(a,b,result);

if(result[1])

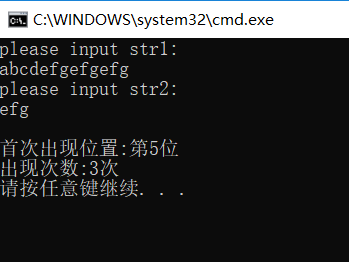
printf("\n首次出现位置:第%d位\n出现次数:%d次\n",result[0],result[1]);

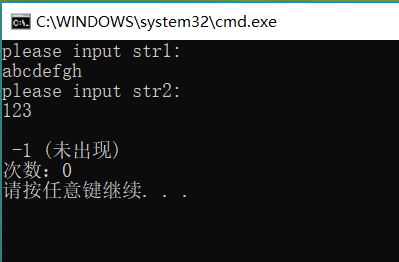
else

printf("\n %d (未出现)\n次数：%d\n",result[0],result[1]);

return 0;

}





1. 最长单词

//最长单词查找，支持标点符号和连字符‘-’

#include<stdio.h>

#include<string.h>

int judge(char a)

{

int k;

if((a>='a'&&a<='z')||(a>='A')&&(a<='Z')||a=='-')k=1;

else k=0;

return k;

}

char \*search(char s[],int n)

{

int i,j,k=1,count[100],max=0;

static char b[20];

if(judge(s[0]))

{

for(i=0;s[i]!=' ';i++);

count[0]=i;

max=count[0];

}

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

if(!judge(s[i])&&judge(s[i+1]))

{

for(j=i+1;judge(s[j]);j++);

count[k]=j-i-1;

if(count[k]>max)max=count[k];

k++;

i=j-1;

}

if(judge(s[0]))

{

for(i=0;judge(s[i]);i++);

if(max==i)

{

for(j=0,i=0;s[i]!=' ';i++)

b[j++]=s[i];

}

}

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

if(!judge(s[i])&&judge(s[i+1]))

{

for(j=i+1;judge(s[j]);j++);

if(max==j-i-1)

{

for(j=0,i=i+1;judge(s[i]);i++)

{

b[j++]=s[i];

}

break;

}

i=j-1;

}

b[j]='\0';

return b;

}

int main()

{

int i;

char s[1000];

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

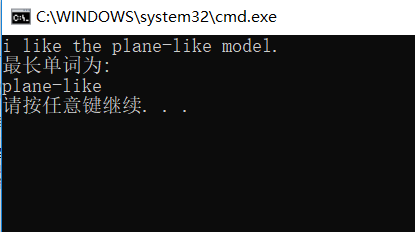
s[i]='\0';

printf("最长单词为:\n");

puts(search(s,i));

return 0;

}



1. 最长公共子串

#include<stdio.h>

#include<string.h>

void searchlong(char a[],char b[],char\*p,int\*ju)

{

int i,j,k,length[300]={0},t=0,max;

char flag[300][50];

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

for(j=0;b[j]!='\0';j++)

if(a[i]==b[j])

{

t++;

for(k=0;a[i+k]==b[j+k]&&a[i+k]!='\0'&&b[i+k]!='\0';k++);

length[t]=k;

for(k=0;k<length[t];k++)

flag[t][k]=a[i+k];

flag[t][k]='\0';

}

if(t>0)

{

ju[0]=1;

max=1;

for(i=2;i<=t;i++)if(length[i]>length[max])max=i;

strcpy\_s(p,length[max]+1,flag[max]);

}

else ju[0]=0;

}

int main()

{

int judge[1];

char str1[100],str2[200],c[200];

printf("please input str1:\n");

gets\_s(str1);

printf("please input str2:\n");

gets\_s(str2);

searchlong(str1,str2,c,judge);

if(judge[0])

{

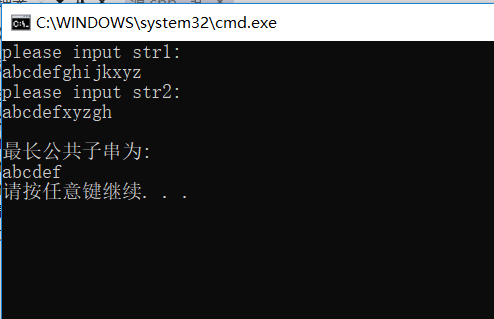
printf("\n最长公共子串为:\n%s\n",c);

}

else printf("\n未找到公共子串！\n");

return 0;

}



1. 成绩排名

#include<stdio.h>

#include<string.h>

#include<math.h>

void swapnum(int&a,int&b){int t=a;a=b;b=t;}

void swapstr(char s1[],char s2[])

{

char t[20];

strcpy\_s(t,20,s1);

strcpy\_s(s1,20,s2);

strcpy\_s(s2,20,t);

}

int ChangetoNum(char\*str)//将字符串转化为数字

{

int len=strlen(str);

int num=0;

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

num+=(int)pow(10.0,len-i-1)\*(int)(str[i]-'0');

return num;

}

void input(int a[],char b[][20],int c[],int n)

{

int i;

printf("please input the information of students:\n");

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

{

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

scanf\_s(" %s",b[i],20);

scanf\_s("%d",&c[i]);

fflush(stdin);

}

}

void rank(int a[],char b[][20],int c[],int d[],int n)

{

int i,j;

for(i=n;i>1;i--)//按成绩递减排序

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

{

if((c[j]<c[j+1])||(c[j]==c[j+1]&&a[j]>a[j+1]))

{

swapnum(c[j],c[j+1]);

swapnum(a[j],a[j+1]);

swapstr(b[j],b[j+1]);

}

}

for(i=1,d[1]=1;i<n;i++)//排名

{

if(c[i+1]<c[i])d[i+1]=i+1;

else if(c[i+1]==c[i])d[i+1]=d[i];

}

}

void bsearch(int a[],char b[][20],int c[],int d[],int n)

{

int low,high,mid,flag,num;

char numstr[20];

int i,j;

for(i=n;i>1;i--)//按学号递增排序

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

{

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

{

swapnum(c[j],c[j+1]);

swapnum(a[j],a[j+1]);

swapnum(d[j],d[j+1]);

swapstr(b[j],b[j+1]);

}

}

while(1)

{

printf("please input the number you want to search:");

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

fflush(stdin);

low=1;high=n;flag=0;

if(strcmp(numstr,"0000000000")==0)break;

else

{

num=ChangetoNum(numstr);

while(low<=high)//折半查找

{

mid=(low+high)/2;

if(a[mid]==num)

{

printf("name:%s score:%d rank:%d\n",b[mid],c[mid],d[mid]);

flag=1;

break;

}

else if(a[mid]<num)

low=mid+1;

else if(a[mid]>num)

high=mid-1;

}

if(flag==0)printf("Error!this number doesn't exist!\n");

}

}

}

int main()

{

int N;

int number[200],score[200],r[200];

char name[200][20];

printf("please input the number of students:");

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

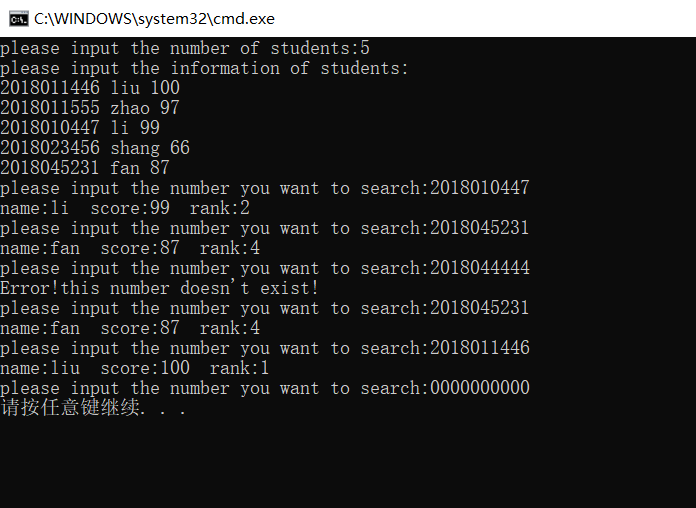
input(number,name,score,N);

rank(number,name,score,r,N);

bsearch(number,name,score,r,N);

return 0;

}



1. 日历计算

//日历计算

#include<stdio.h>

int leap(int a)

{

if(((a%4==0)&&((a%100)!=0))||(a%400==0))return 1;

else return 0;

}

void time(int year,int month,int day,int\*p)

{

int days=day,total=0,i,week1,week2,week,t;

switch(month)

{

case 12:days+=30;

case 11:days+=31;

case 10:days+=30;

case 9:days+=31;

case 8:days+=31;

case 7:days+=30;

case 6:days+=31;

case 5:days+=30;

case 4:days+=31;

case 3:{

if(leap(year))days+=29;

else days+=28;

}

case 2:days+=31;

case 1:break;

}

p[1]=days;//今年的第几天

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

{

total+=365;

if(leap(i))total+=1;

}

week1=total/7;//从公元1年1月1日到上一年年末经过的星期数

t=total%7;//上一年最后一天是星期几

total+=days;//从1年1月1日到输入日期的总天数

if(!(total%7))p[3]=7;

else p[3]=total%7;

week2=total/7;//从1年1月 1日到现在经过几周

week=days/7;//输入日期处于该年第几周

//若该年第一天不是星期一，该年的第一天按位于 第一周 计算！

//以 星期一 作为新一周的开始！

if(week==0)week++;

else if((week!=0)&&(t!=0&&days%7==0)||days%7!=0)week++;

p[2]=week;

}

int judge(int a,int b,int c)//判断输入的日期是否合理

{

int z=1;

int m[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};

if(leap(a))m[2]=29;

if(a<=0||b<=0||c<=0)z=0;

if(b<1||b>12)z=0;

else{

switch(b)

{

case 12:{if(c>m[12])z=0;break;}

case 11:{if(c>m[11])z=0;break;}

case 10:{if(c>m[10])z=0;break;}

case 9:{if(c>m[9])z=0;break;}

case 8:{if(c>m[8])z=0;break;}

case 7:{if(c>m[7])z=0;break;}

case 6:{if(c>m[6])z=0;break;}

case 5:{if(c>m[5])z=0;break;}

case 4:{if(c>m[4])z=0;break;}

case 3:{if(c>m[3])z=0;break;}

case 2:{if(c>m[2])z=0;break;}

case 1:{if(c>m[1])z=0;break;}

}

}

return z;

}

int main()

{

int y,m,d,date[4];

while(1)

{

printf("请输入日期(格式为:2000 1 1):");

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

if(judge(y,m,d))

{

time(y,m,d,date);

if(date[3]==7)printf("%d年 第%d天，第%d周，星期日\n",y,date[1],date[2]);

else printf("%d年 第%d天，第%d周，星期%d\n",y,date[1],date[2],date[3]);

break;

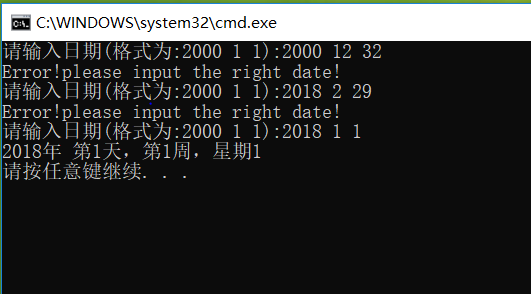
}

else printf("Error!please input a reasonable date!\n");

}

return 0;

}



1. 分书问题

//分书问题

//回溯递归

#include<stdio.h>

int like[6][6]={{0},

{0,0,0,1,1,0},

{0,1,1,0,0,1},

{0,0,1,1,0,1},

{0,0,0,0,1,0},

{0,0,1,0,0,1}},

flag[6]={0},

taken[6]={0};

void solution(int i)

{

int j,k;

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

{

if((flag[j]!=0)||(like[i][j]==0))continue;

taken[i]=j;

flag[j]=1;

if(i==5)

{

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

printf("%d ",taken[k]);

putchar(10);

}

else solution(i+1);

taken[i]=0;

flag[j]=0;

}

}

int main()

{

printf("分书方案：\nA B C D E\n");

solution(1);

return 0;

}

