1. 有1，2,3,三个数字，能组成多少个互不相同且无重复数字的三位数？

#include <stdio.h>

main()

{

int i,j,k;

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

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

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

if(i!=j&&i!=j&&j!=k)

printf("%d%d%d ",i,j,k);

}

1. 判断500到700之间有多个素数，并输出。

#include <stdio.h>

main()

{

int i,j=0,x,flag;

for(x=500;x<=700;x++)

{flag=1;

for(i=2;i<=x-1;i++)

if(x%i==0){flag=0;break;}

if(flag==1){printf("%d ",x);j++;}

}

printf("\nsum=%d",j);

}

1. 输入年月日，判断该过了多少天。

#include <stdio.h>

main()

{

int year,month,day,sum,flag;

scanf("%d,%d,%d",&year,&month,&day);

flag=0;

if(year%400==0 || year%4==0 &&year%100!=0 ) flag=1;

switch(month)

{

case 1: sum=0; break;

case 2: sum=31;break;

case 3: sum=31+28+flag;break;

case 4: sum=31+28+31+flag;break;

case 5: sum=31+28+31+30+flag;break;

case 6: sum=31+28+31+30+31+flag;break;

case 7: sum=31+28+31+30+31+30+flag;break;

case 8: sum=31+28+31+30+31+30+31+flag;break;

case 9: sum=31+28+31+30+31+30+31+30+flag;break;

case 10: sum=31+28+31+30+31+30+31+30+31+flag;break;

case 11: sum=31+28+31+30+31+30+31+30+31+30+flag;break;

case 12: sum=31+28+31+30+31+30+31+30+31+30+31+flag;break;

}

sum=sum+day;

printf(" %d",sum);}

1. 输入6位同学的成绩，输出最大值，平匀值。并按从小到大的顺序进行排序输出。

#include <stdio.h>

main()

{

int a[10];

int i,j,sum,avg,min,temp;

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

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

sum=0;

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

sum=sum+a[i];

printf("\nsum=%d,avg=%d",sum,sum/10);

for(j=0;j<=8;j++)

{

min=j;

for(i=j+1;i<=9;i++)

if(a[min]>a[i]) min=i;

temp=a[j];a[j]=a[min];a[min]=temp;

}

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

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

}

1. 用函数实现，输入一个字符串，判断是否是回文。

#include <stdio.h>

#include <string.h>

main()

{

char s[100];

int i,j,flag;

flag=1;

gets(s);

for(i=0,j=strlen(s)-1;i<=j;i++,j--)

if(s[i]!=s[j]){flag=0;break;}

if(flag==1){printf("yes");}

else printf("no");

}

函数形式

#include <stdio.h>

#include <string.h>

void huiwen(char c[100],int x)

{ int i,j,flag=1;

for(i=0,j=strlen(c)-1;i<=j;i++,j--)

if(c[i]!=c[j]) {flag=0;break;}

if(flag==1) printf("yes");

else printf("no");

}

main()

{

char s[100];

int i,j,flag;

flag=1;

gets(s);

huiwen(s,10);

}

1. 用递归实现，输入N，求1到N之积。

#include <stdio.h>

double fun(int n);

int main(void)

{

int n;

printf("Enter n:");

scanf("%d",&n);

printf("%lf\n",fun(n));

return 0;

}

double fun(int n)

{if(n==0||n==1)

return 1;

else return n\*fun(n-1);

}

7.

#include <stdio.h>  
struct stu  
{ char name[20];  
int english;  
int math;  
};  
int main()  
{ int i,a,b;  
struct stu student[3]={{"hu",87,89},{"wu",68,78},{"mei",99,68}};  
a=0;b=0;  
for(i=0;i<=2;i++)  
a=a+student[i].english,b=b+student[i].math;  
printf("english avg=%d",a/3);  
printf("math avg=%d",b/3);  
}