

1 编程：要求输入整数 a 和 b，若 $a*a+b*b$ 大于 100，则输出 $a*a+b*b$ 百位以上的数字，否则输出两数之和 $a+b$ 。

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
int a,b,c;
```

```
if(a*a+b*b>100)
```

```
{
```

```
c=(a*a+b*b)/100;
```

```
}
```

```
else
```

```
}
```

2 编程：根据以下函数关系：当 $x \leq 0$ ， $y=0$ ；当 $0 < x \leq 10$ ， $y=x$ ；当 $10 < x \leq 20$ ， $y=10$ ；当 $x > 20$ ， $y=\sin(x)$ ；输入一个 x 值，计算出相应的 y 值并输出。（提示： $\sin(x)$ 属于 `math.h`）

```
#include<stdio.h>
```

```
#include<math.h>
```

```
main()
```

```
{
```

```
int x;
```

```
float y;
```

```
if(x<=0)
```

```
{
```

```
y=0;
```

```
}
```

```
else if((0<x)&&(x<=10))
```

```
{
```

```
y=x;
```

```
}
```

```
else if ((10<x)&&(x<=20))
```

```
{
```

```
y=10;
```

```
}
```

```
else
y=sin(x);

}
```

3 编程：根据以下函数关系：当 $x \leq 1$ ， $y=x$ ；当 $1 < x \leq 2$ ， $y=\sqrt{x}$ ；当 $x > 2$ ， $y=\sin(x)$ ；输入一个 x 值，计算出相应的 y 值并输出。（提示： $\sin(x)$ 、 $\sqrt{x}$ 属于 `math.h`） 参考 2

4 编程：输入三个正整数 x,y,z ，将这三个数由小到大输出。

```
#include<stdio.h>
#include<math.h>
main()
{
int a[3],i,max,j;

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

}

for(i=0;i<3;i++)
{
for(j=0;j<2;j++)
if(a[j]>a[j+1])
{
max=a[j];
a[j]=a[j+1];
a[j+1]=max;
}
}

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

}
```

```
#include <stdio.h>
```

```

main()

{

int a,b,c,t;


if(a>b)

{

t=a;a=b;b=t;

}

if(a>c)

{

t=a;a=c;c=t;

}

if(b>c)

{

t=b;b=c;c=t;

}

}

```

5 编程：从键盘输入一个年号，判断该年号是否为闰年，并输出结果。闰年是能被 4 整除但不能被 100 整除；或能被 100 整除且能被 400 整除的年号。 练习书上有，第一页

6 编程实现四位电话号码的升位。原来电话号码为 1、3、5、7、9 打头的号码前加 2，原来电话号码为 2、4、6、8 打头的加 3，例如：输入 1234，则输出：21234。

```
#include<stdio.h>
```

```

main()
{
long x,y;

```

```

y=x/1000;
switch(y)
{
case 1:
case 3:
case 5:
case 7:
case 9:x=2*10000+x;break;
case 2:
case 4:
case 6:
case 8:x=3*10000+x;break;
}

}

```

7 编程实现四位电话号码的升位。原来电话号码为 1、2、3、4、5 打头的号码前加 3，原来电话号码为 6、7、8、9 打头的加 2，例如：输入 1234，则输出：31234。 参考 6

8 编程：已知分段函数：当 $x > 10$ 时， $y = 2 * x + 1$; 当 $1 < x \leq 10$ 时， $y = 5 * (x + 1) * (x - 2)$; 当 $x \leq 1$ 时， $y = 1 - 2 * x$; 从键盘输入 x 的值，计算 y 的值。 参考 2

9 编程：输入 3 个正整数，判断能否组成三角形，如果能够组成三角形，则输出三角形的面积，否则输出错误。三角形面积公式： $area = \sqrt{s(s-a)(s-b)(s-c)}$ ，其中 $s = 1/2(a+b+c)$ 。（提示： $\sqrt{x}$ 属于 `math.h`）

```

#include<stdio.h>
#include<math.h>
main()
{
float x,y,z;
float area,s;

if((x+y<=z)|| (z+y<=x)|| (x+z<=y))

else
{
s=(x+y+z)/2;

```

```
area=sqrt(s*(s-x)*(s-y)*(s-z));
```

```
}  
}
```

10 编程：已知分段函数，当 $x \leq 0$ 时， $y=3*x$ ；当 $0 < x \leq 10$ 时， $y=(x+1)*(x-2)$ ；当 $x > 10$ 时， $y=2*x*x$ ；从键盘输入 x 的值，计算 y 的值。 参考 2

11 编程：判定用户输入一个不大于 999 的正整数是否为一个回文数。回文数是指正读和反读都相同的数。

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
int a,x,y,z;
```

```
if(0<a&&a<=999)
```

```
{
```

```
x=a/100;
```

```
y=(a-100*x)/10;
```

```
z=a%10;
```

```
if(x==0)
```

```
{
```

```
if(y==z)
```

```
else
```

```
}
```

```
else if(x==z)
```

```
else
```

```
}
```

```
}
```

12 编程：从键盘输入一个两位数的正整数 a ，将其转换形成一个整数 b 并输出 b 。转换的方式是：将 a 中较大的数码作为 b 的十位，剩余的数码作为 b 的个位。例如，当输入 $a=49$ ，则结果 $b=94$ 。

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
int a,b,x,y;
```



```

if(0<a&&a<100)
{
x=a/10;
y=a%10;
if(x>=y)
{
b=a;
}
else
{
b=10*y+x;
}

}
else
}

```

13 编程：从键盘输入一个两位数的正整数 **a**，将其转换形成一个整数 **b** 并输出 **b**。转换的方式是：将 **a** 中较小的数码作为 **b** 的十位，剩余的数码作为 **b** 的个位。例如，当输入 **a=54**，则结果 **b=45**。 参考 12

14 初始化数组 **a** 中包含 10 个不同的整数，要求从键盘输入一个整数，若这个整数在数组中则从数组中删除这个整数并输出删除后的数组，否则输出 **no**。

```

#include<stdio.h>
main()
{
int a[10]={2,3,4,1,5,6,7,8,9,10},i,x,j,k;

```

```

for(i=0;i<10;i++)
{
if(a[i]==x)
{
for(j=i;j<10;j++)
a[j]=a[j+1];
for(k=0;k<9;k++)

break;
}
else if(i==9)

```

```
}  
}
```

15 初始化一字符串（字符串中的每个字符各不同），要求从键盘输入一个字符，删除字符串中和输入字符相同的字符，输出处理后的字符串。

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
char b[1];
```

```
int i,j,k;
```

```
gets(b);
```

```
for(i=0;i<7;i++)
```

```
{
```

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

```
{
```

```
for(j=i;j<5;j++)
```

```
{
```

```
a[j]=a[j+1];
```

```
}
```

```
for(k=0;k<5;k++)
```

```
break;
```

```
}
```

```
{
```

```
break;
```

```
}
```

```
}
```

```
}
```

16 计算一个 4*4 矩阵的主对角线的元素之和以及最大元素的值。要求初始化矩阵，输出计算结果并输出该矩阵。

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
int a[4][4]={ {2,123,1,5},{4,22,56,23},{5,7,89,23},{1,2,3,4}};
```

```

int i,sum=0,max,j;
for(i=0;i<4;i++)
sum=sum+a[i][i];
max=a[0][0];
for(i=0;i<4;i++)
{
for(j=0;j<4;j++)
if(max<a[i][j])
max=a[i][j];
}

}

```

17 计算一个 3*3 矩阵的所有元素之和以及所有元素中的最大元素的值。要求初始化矩阵，输出计算结果并输出该矩阵。 参考 16，懒得弄了

18 初始化一字符串（长度小于 20 个字符），取出第 3 个字符到第 7 个字符组成的子串，输出该子串。

```

#include<stdio.h>
main()
{

int i;
puts(a);
for(i=2;i<7;i++)

}

```

19 编程计算一个 a[3][4]数组的最小值所在位置，输出该位置以及 a[3][4]矩阵。要求初始化矩阵。 参考 16

20 初始化一矩阵 a[4][4],计算每行的最大元素值，要求输出原始矩阵以及每行的最大元素值。

```

#include<stdio.h>
main()
{
int a[4][4]={ {2,123,1,5},{4,22,56,23},{5,7,89,23},{1,2,3,4}},i,j,max;
for(i=0;i<4;i++)
{
max=a[i][0];
for(j=0;j<4;j++)
{

}

}
}

```



```

}
for(i=0;i<4;i++)
{
for(j=0;j<4;j++)
{
if(max<a[i][j])
max=a[i][j];
}

}

}

```

21 初始化一矩阵 **a[4][4]**,计算每列的最大元素值，要求输出原始矩阵以及每列的最大元素值。 参考 20

22 编程计算 **a[3][4]**数组的最大值所在位置，输出该位置以及 **a[3][4]**矩阵。要求初始化矩阵。 参考 20

23 编程实现：计算矩阵 **a[3][4]**每行元素的和。要求输出原始矩阵以及每行元素的和值。要求初始化矩阵。
参考 20

24 编程实现：计算矩阵 **a[3][4]**每列元素的和。要求输出原始矩阵以及每列元素的和值。要求初始化矩阵。
参考 20

25 一维数组中存放任意 10 个整数，从第四个数据开始直到最后一个数据，依次向右移动一个位置。输出原始数据和移动后的结果。要求初始化数组。 例如：原始数据为：6 10 2 9 8 7 17 18 23 45 输出结果为：
6 10 2 9 9 8 7 17 18 23 45

```

#include<stdio.h>
main()
{
int a[11]={6,10,2,9,8,7,17,18,23,45},i;

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

for(i=10;i>3;i--)
a[i]=a[i-1];

for(i=0;i<11;i++)

}

```


以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/838027051012006075>