

第 4 章选择结构程序设计

4-4

方法 1

```
#include <stdio.h>

int main()
{
    int a,b,c;
    printf("请输入三个整数:");
    scanf("%d,%d,%d",&a,&b,&c);
    if (a<b)
        if (b<c)
            printf("max=%d\n",c);
        else
            printf("max=%d\n",b);
    else if (a<c)
        printf("max=%d\n",c);
    else
        printf("max=%d\n",a);
    return 0;
}
```

方法 2

```
#include <stdio.h>

int main()
{ int a,b,c,temp,max;
    printf("请输入三个整数:");
    scanf("%d,%d,%d",&a,&b,&c);
    temp=(a>b)?a:b;           /*将 a 和 b 中的大者存入 temp 中*/
    max=(temp>c)?temp:c;      /*将 a 和 b 中的大者与 c 比较,取最大者*/
    printf("三个整数的最大数是%d\n",max);
    return 0;
}
```

4-5

方法 1

```
#include <stdio.h>
#include <math.h>
#define M 1000

int main()
{
    int i,k;
    printf("请输入一个小于%d 的正整数 i:",M);
    scanf("%d",&i);
```

```

if (i>=M || i<=0)
{printf("输入的数不符合要求，请重新输入一个小于%d 的正整数 i:",M);
scanf("%d",&i);
}
k=sqrt(i);
printf("%d 的平方根的整数部分是: %d\n",i,k);
return 0;
}

```

方法 2

```

#include <stdio.h>
#include <math.h>
#define M 1000
int main()
{
    int i,k;
    printf("请输入一个小于%d 的整数 i:",M);
    scanf("%d",&i);
    while (i>=M || i<=0)
    {printf("输入的数不符合要求，请重新输入一个小于%d 的整数 i:",M);
    scanf("%d",&i);
    }
    k=sqrt(i);
    printf("%d 的平方根的整数部分是: %d\n",i,k);
    return 0;
}

```

4-8

```

#include <stdio.h>
int main()
{ float score;
  char grade;
  printf("请输入学生成绩:");
  scanf("%f",&score);
  while (score>100||score<0)
  {printf("\n 输入有误,请重输");
  scanf("%f",&score);
  }
  switch((int)(score/10))
  { case 10:
    case 9: grade='A';break;
    case 8: grade='B';break;
    case 7: grade='C';break;

```

```

        case 6: grade='D';break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0: grade='E';
    }
    printf("成绩是 %5.1f,相应的等级是%c\n ",score,grade);
    return 0;
}

```

4-9

```

#include <stdio.h>
#include <math.h>
int main()
{
    int num,indiv,ten,hundred,thousand,ten_thousand,place;    //分别代表个位,十位,百位,千
    位,万位和位数
    printf("请输入一个整数(0-99999):");
    scanf("%d",&num);
    if (num>9999)
        place=5;
    else if (num>999)
        place=4;
    else if (num>99)
        place=3;
    else if (num>9)
        place=2;
    else place=1;
    printf("位数:%d\n",place);
    printf("每位数字为:");
    ten_thousand=num/10000;
    thousand=(int)(num-ten_thousand*10000)/1000;
    hundred=(int)(num-ten_thousand*10000-thousand*1000)/100;
    ten=(int)(num-ten_thousand*10000-thousand*1000-hundred*100)/10;
    indiv=(int)(num-ten_thousand*10000-thousand*1000-hundred*100-ten*10);
    switch(place)
    { case 5:printf("%d,%d,%d,%d,%d",ten_thousand,thousand,hundred,ten,indiv);
        printf("\n 反序数字为:");
        printf("%d%d%d%d%d\n",indiv,ten,hundred,thousand,ten_thousand);
        break;
        case 4:printf("%d,%d,%d,%d",thousand,hundred,ten,indiv);

```

```

        printf("\n 反序数字为:");
        printf("%d%d%d%d\n",indiv,ten,hundred,thousand);
        break;
    case 3:printf("%d,%d,%d",hundred,ten,indiv);
        printf("\n 反序数字为:");
        printf("%d%d%d\n",indiv,ten,hundred);
        break;
    case 2:printf("%d,%d",ten,indiv);
        printf("\n 反序数字为:");
        printf("%d%d\n",indiv,ten);
        break;
    case 1:printf("%d",indiv);
        printf("\n 反序数字为:");
        printf("%d\n",indiv);
        break;
    }
    return 0;
}

```

4-11

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

```
int main()
```

```

{int  t,a,b,c,d;
    printf("请输入四个数:");
    scanf("%d,%d,%d,%d",&a,&b,&c,&d);
    printf("a=%d,b=%d,c=%d,d=%d\n",a,b,c,d);
    if (a>b)
        { t=a;a=b;b=t;}
    if (a>c)
        { t=a;a=c;c=t;}
    if (a>d)
        { t=a;a=d;d=t;}
    if (b>c)
        { t=b;b=c;c=t;}
    if (b>d)
        { t=b;b=d;d=t;}
    if (c>d)
        { t=c;c=d;d=t;}
    printf("排序结果如下: \n");
    printf("%d  %d  %d  %d  \n" ,a,b,c,d);
    return 0;
}

```

第 5 章循环结构程序设计

5-2

```
#include <stdio.h>
#include <math.h> // 程序中用到数学函数 fabs, 应包含头文件 math.h
int main()
{
    int sign=1,count=0; // sign 用来表示数值的符号,count 用来统计循环次数
    double pi=0.0,n=1.0,term=1.0; // pi 开始代表多项式的值,最后代表  $\pi$  的值, n 代表分母,
    term 代表当前项的值
    while(fabs(term)>=1e-8) // 检查当前项 term 的绝对值是否大于或等于 10 的(-8)次方
    {
        pi=pi+term; // 把当前项 term 累加到 pi 中
        n=n+2; // n+2 是下一项的分母
        sign=-sign; // sign 代表符号, 下一项的符号与上一项符号相反
        term=sign/n; // 求出下一项的值 term
        count++; // count 累加 1
    }
    pi=pi*4; // 多项式的和 pi 乘以 4, 才是  $\pi$  的近似值
    printf("pi=%10.8f\n",pi); // 输出  $\pi$  的近似值
    printf("count=%d\n",count); // 输出循环次数
    return 0;
}
```

5-3

方法 1

```
#include <stdio.h>
int main()
{
    int p,r,n,m,temp;
    printf("请输入两个正整数 n,m:");
    scanf("%d,%d",&n,&m);
    if (n<m)
    {
        temp=n;
        n=m;
        m=temp;
    }
    p=n*m;
    while(m!=0)
    {
        r=n%m;
        n=m;
    }
```

```

        m=r;
    }
    printf("它们的最大公约数为:%d\n",n);
    printf("它们的最小公倍数为:%d\n",p/n);
    return 0;
}

```

方法 2

```

#include <stdio.h>
main()
{ long m,n,i=1,j,s;
  scanf("%ld,%ld",&m,&n);
  for(;i<=m&&i<=n;i++)
  { if(m%i==0&&n%i==0) s=i;}
  if(m>=n) j=m;
  else j=n;
  for(;!(j%m==0&&j%n==0);j++);
  printf("s=%ld,j=%ld\n",s,j);
}

```

5-6

```

#include <stdio.h>
int main()
{ double s=0,t=1;
  int n;
  for (n=1;n<=20;n++)
  {
    t=t*n;
    s=s+t;
  }
  printf("1!+2!+...+20!=%22.15e\n",s);
  return 0;
}

```

5-8

```

#include <stdio.h>
int main()
{
  int i,j,k,n;
  printf("parcissus numbers are ");
  for (n=100;n<1000;n++)
  {
    i=n/100;
    j=n/10-i*10;

```

```

    k=n%10;
    if (n==i*i*i + j*j*j + k*k*k)
        printf("%d ",n);
    }
    printf("\n");
    return 0;
}

```

5-9

```

#include <stdio.h>
int main()
{
    int m,s,i;
    for (m=2;m<1000;m++)
    {
        s=0;
        for (i=1;i<m;i++)
            if ((m%i)==0) s=s+i;
        if(s==m)
        {
            printf("%d,its factors are ",m);
            for (i=1;i<m;i++)
                if (m%i==0) printf("%d ",i);
            printf("\n");
        }
    }
    return 0;
}

```

5-13

```

#include <stdio.h>
#include <math.h>
int main()
{
    float a,x0,x1;
    printf("enter a positive number:");
    scanf("%f",&a);
    x0=a/2;
    x1=(x0+a/x0)/2;
    do
    {
        x0=x1;
        x1=(x0+a/x0)/2;
    } while(fabs(x0-x1)>=1e-5);
    printf("The square root of %5.2f is %8.5f\n",a,x1);
    return 0;
}

```

```
}
```

5-14

```
#include "stdio.h"
#include "math.h"
int main()
{ double x1, x2;
  x2 = 1.5;
  do
  { x1 = x2;
    x2 = x1 - (2 * x1*x1*x1 - 4 * x1*x1 + 3 * x1 - 6) / (6 * x1*x1 - 8 * x1 + 3);
  } while (fabs(x2 - x1) > 1.0e-6);
  printf("%lf",x2);
  return 0;
}
```

5-16

```
#include <stdio.h>
int main()
{ int i,j,k;
  for (i=0;i<=3;i++)
  { for (j=0;j<=2-i;j++)
    printf(" ");
    for (k=0;k<=2*i;k++)
    printf("*");
    printf("\n");
  }
  for (i=0;i<=2;i++)
  { for (j=0;j<=i;j++)
    printf(" ");
    for (k=0;k<=4-2*i;k++)
    printf("*");
    printf("\n");
  }
  return 0;
}
```