

第6章 循环结构





例:求 $s=1^2+2^2+\dots$,当s大于或等于1240为止

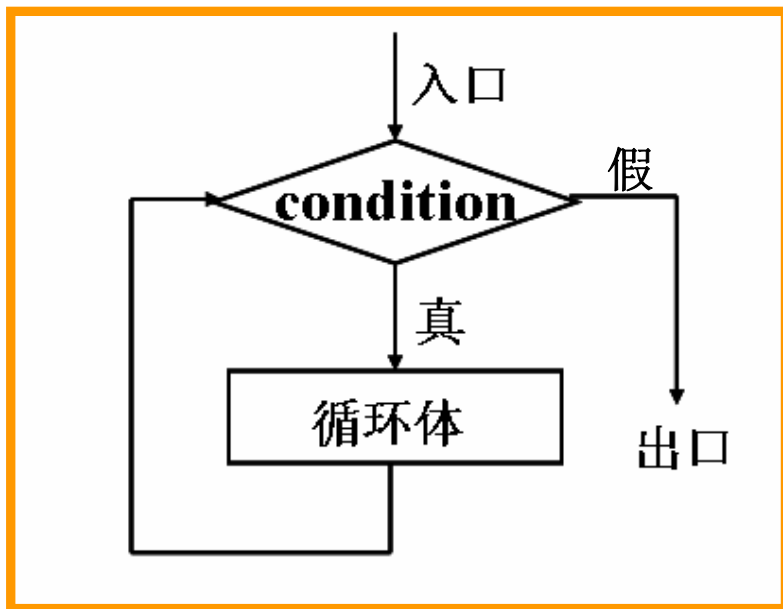
```
main( )
{ int s=0,p=1;
  loop: s+=p*p;
        p++; } 重复序列
        if(s>=1240) goto pr;
        goto loop;
pr: printf(“s=%d”,s);
}
```



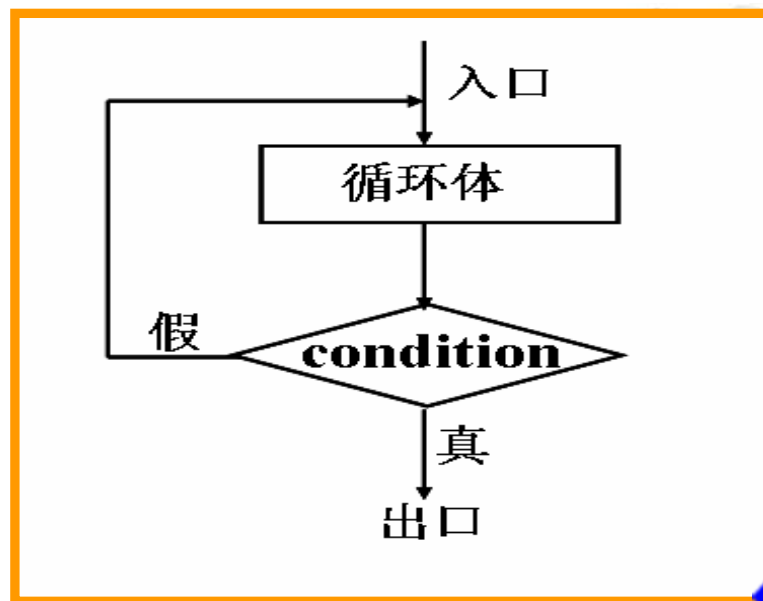
§ 1 循环的概念

循环结构也称重复结构,是指在一定条件下重复执行一组语句序列程序结构。其由二部分组成: 循环体——重复执行的语句序列; 循环控制机构——指明循环的条件

❖ 当型循环 —最少循环0次
当条件成立时, 不断循环



❖ 直到型循环 —最少循环1次
不断循环, 直到条件成立为止





循环构造技术

1. 用if和goto 语句构造循环
2. 用while 语句构造循环
3. 用for 语句构造循环
4. 用do-while 语句构造循环



§ 2 用if和goto构成循环

- **goto语句：** goto 标号标识符；
- **功能：** 无条件转向标号所指定的语句处执行

例：编程求100!之值

```
main( )  
{ int i=1; long t=1;  
  loop: if(i<=100)  
    { t*=i;  
      i++;  
      goto loop;  
    }  
  printf("100!=%ld",t);  
}
```

```
main( )  
{ int i=1; long t=1;  
  loop: t*=i;  
        i++;  
        if(i<=100) goto loop;  
  printf("100!=%ld",t);  
}
```



§ 3 用while构造‘当型’循环

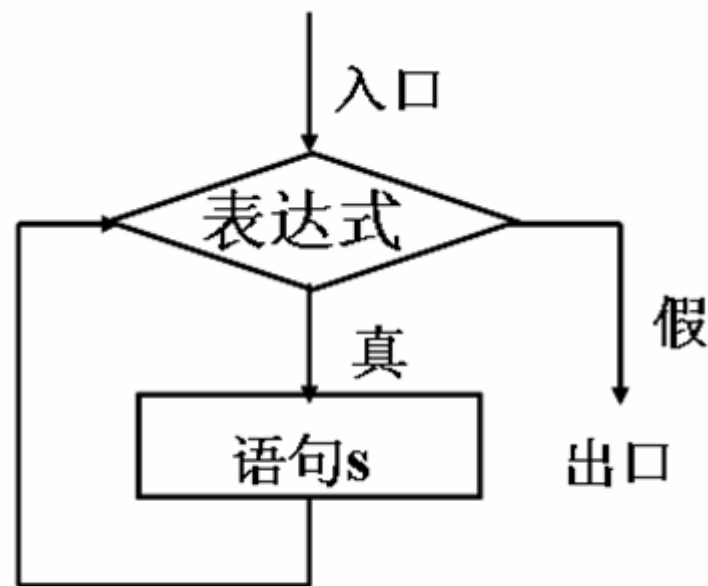
while(表达式) 语句s

```
eg1:while(product%2==0)  
    product/=2;
```

不断消去product的偶数因子，使其最终成为一奇数

```
eg2:while(1)    循环体
```

永真循环(死循环，除非循环体中设置循环的出口)



while的控制流程



§ 3 用while构造‘当型’循环

例1：在while(!a)中，其中“!a”与_____等价。

- A. $a==0$ B. $a==1$ C. $a!=1$ D. $a!=0$

A

例2：设有程序段

```
int k=0;
```

```
while(k=0) k=k-1;
```

则下面描述中正确的是_____。

- A. while循环执行10次 B. 循环是无限循环
C. 循环体执行0次 D. 循环体执行1次

C



§ 3 用while构造‘当型’循环

例3：设有程序段

```
int x=0,s=0;  
while(!x!=0) s+=++x;  
printf(“%d”,s); 则 B。
```

- A. 运行程序段后输出0 B. 运行程序段后输出1
C. 程序段中控制表达式非法 D. 程序段执行无限次

例4：下面程序段的运行结果是 **A**。

```
a=1; b=2; c=2;  
while(a<b<c) { t=a; a=b; b=t; c--;}  
printf(“%d,%d,%d”,a,b,c);
```

- A. 1,2,0 B. 2,1,0 C. 1,2,1 D. 2,1,1



§ 3 用while构造‘当型’循环

例5：下面程序的功能是从键盘输入的一行字符中统计出大写字母的个数m和小写字母的个数n,并输出m和n中的大者。请填空。

```
#include <stdio.h>
main( )
{ int m=0,n=0;
  char c;
  while( (1) !='\n')
  { if(c>='A'&&c<='Z') m++;
    if(c>='a'&&c<='z') n++;
  }
  printf(“%d\n”,m<n? (2) );
}
```

(c=getchar())

n:m



§ 3 用while构造‘当型’循环

例6：对键盘输入的一行字符分类统计，列出数字字符、空白格字符以及其它字符出现的次数，并统计该行长度（总字符数）。

```
#include <stdio.h>
main( )
{char c; int ndigit=0,nwhite=0,nother=0,length;
  while( (c=getchar())!='\n')
    switch(c)
    { case '0':case '1':case '2':case '3':case '4':case '5':
      case '6':case '7':case '8':case '9': ndigit++; break;
      case '\40': case '\t': nwhite++; break;
      default: nother++; }
  length=ndigit+nwhite+nother;
  printf("%d,%d,%d,%d\n",ndigit,nwhite,nother,length); }
```



§ 4 用do-while构造‘直到型’循环

do 语句**s while**(表达式);

例1(考题): 设整型变量x的当前值为3,执行以下循环后,输出结果是__。

```
do printf(“%d\t”,x-=2);  
while(!(--x));
```

1

-2

例2: 以下程序段_____。

```
x=-1;
```

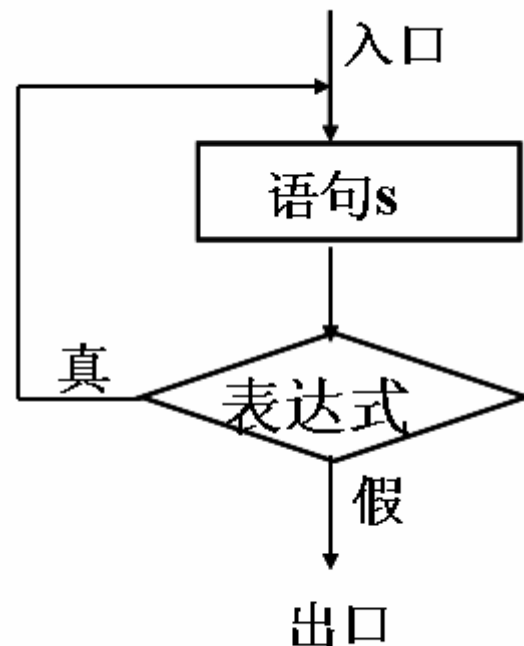
```
do x=x*x; while(! x);
```

A.死循环

B.循环执行2次

C. 循环1次

D.有语法错误



do-while的控制流程

C



§ 4 用do-while构造‘直到型’循环

例3：下面程序的功能是计算正整数2345的各位数字平方和，请选择填空。

```
main( )  
{ int n=2345,sum=0;  
do { sum=sum+ (1) ;  
      n= (2) ;  
}while(n);  
printf("sum=%d",sum);  
}
```

B

C

(1) A. $n\%10$ B. $(n\%10)*(n\%10)$ C. $n/10$ D. $(n/10)*(n/10)$

(2) A. $n/1000$ B. $n/100$ C. $n/10$ D. $n\%10$



§ 4 用do-while构造‘直到型’循环

例4：下面程序的功能是把316表示为两个加数的和，使两个加数分别能被13和11整除。请选择填空。

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

```
main( )
```

```
{ int i=0,j,k;
```

```
  do { i++; k=316-13*i;}
```

```
  while(__(1)__);
```

B

```
  j=k/11;
```

```
  printf(“316=13*%d+11*%d”,i,j);
```

```
}
```

(1) A. k/11

B. k%11

C. k/11==0

D. k/11==0



§ 4 用do-while构造‘直到型’循环

例5:求 $\text{sum}=1-2+3-4+\dots+99-100$ 之值

```
#include <stdio.h>
main( )
{ int sum,sign,i;
  sum=0,sign=1;
  i=1;
  do { sum=sum+sign*i;
      sign=-sign;
      i++;
    } while(i<=100);
  printf("sum=%d\n",sum);
}
```

```
#include <stdio.h>
main( )
{ int sum,sign,i;
  sum=0,sign=1;
  i=1;
  while(i<=100)
  { sum=sum+sign*i;
    sign=-sign;
    i++;
  }
  printf("sum=%d\n",sum);
}
```



§ 5 用for构造‘当型’循环

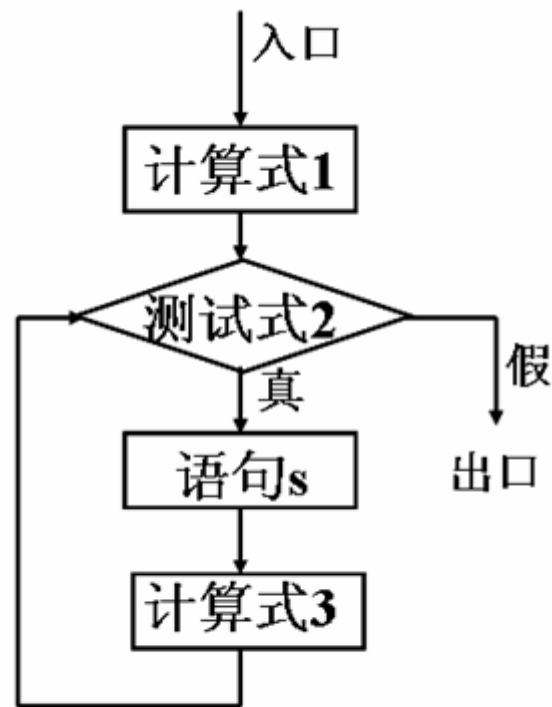
for([表达式1];[表达式2];[表达式3]) 语句s

● 表达式1~3均可缺省，但尾随；不能缺省

● 表达式1、3均可可是逗号表达式，从而对多个变量初始化及迭代

● 表达式2表示继续循环的条件，可以是任何表达式

eg：编程求一行输入字符的ASCII值之和。



for循环控制流程



§ 5 用for构造‘当型’循环

例1：编程求n,m,s之值： $n = \sum_{k=1}^{100} k$, $m = \sum_{k=1}^{100} k^2$, $s = \sum_{k=1}^{100} \frac{1}{k}$

```
#include<stdio.h>
main( )
{ int k,n=0,m=0;
  float s=0.0;
  for(k=1;k<=100;k++)
  { n=n+k;
    m=m+k*k;
    s=s+1./k;
  }
  printf("n=%d,m=%d,s=%.2f\n",n,m,s);
}
```

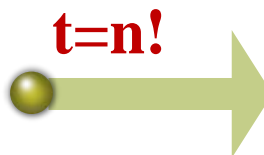




§ 5 用for构造‘当型’循环

例2：编程求 $t=10!$ 之值

```
#include <stdio.h>
main( )
{ int i,t=1;
  for(i=1;i<=10;i++)
    t*=i;
  printf("10!=%d\n",t);
}
```



```
#include <stdio.h>
main( )
{ int i,n;
  unsigned long t=1;
  scanf("%d",&n);
  for(i=1;i<=n;i++)
    t*=i;
  printf("%d!=%lu\n",n,t);
}
```



§ 5 用for构造‘当型’循环

例3：编程求s之值

$$s = \sum_{i=1}^{10} i!$$

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

```
main( )
```

```
{ int i,t=1,s=0;
```

```
  for(i=1;i<=10;i++)
```

```
    { t*=i; /* 将t中值由(i-1)!变为i! */
```

```
      s+=t;
```

```
    }
```

```
    printf("s=%d\n",s);
```

```
}
```



§ 5 用for构造‘当型’循环

例4：编程打印Fibonacci数列前40项

```
#include <stdio.h>

main( )
{ int i;
  long f1,f2;
  f1=1,f2=1;      /*第一组数*/
  for(i=1;i<=20;i++)
  { printf("%15ld%15ld",f1,f2); /*打印第i组数*/
    if(i%2==0) printf("\n");    /*控制一行4个数*/
    f1=f1+f2;
    f2=f2+f1;                  /*求第i+1组数*/
  }
}
```



§ 6 break和continue语句

一、间断语句

break;

退出switch语句的语句体

从包含break的循环中转移出来,即终止整个循环

总是退出最内层结构

```
main( )  
{ int i;  
  for(i=1; i<= 10; i++)  
    printf("%3d",i);  
}
```

1 2 3 4 5 6 7 8 9 10

```
main( )  
{ int i;  
  for(i=1; i<= 10; i++)  
    {printf("%3d",i);if(i==5) break;}  
}
```

1 2 3 4 5



§ 6 break和continue语句

一、间断语句

break;

退出switch语句的语句体

从包含break的循环中转移出来,即终止整个循环

总是退出最内层结构

例1(考题): break语句 C。

A.用在能出现语句的任何地方

B.只能用在循环体中

C.只能用在循环体内或switch语句中

D.可用作函数内的任一语句



§ 6 break和continue语句

例2：编程判自然数n是否是质数

【解法1】

```
#include <stdio.h>
main( )
{ int n, i;
  scanf("%d",&n);
  for(i=2; i<=n-1; i++)
    if(n%i==0) break;
  if(i>=n) printf("%d is\n",n);
  else printf("%d is not\n",n);
}
```

【解法2】

```
#include <stdio.h>
#include <math.h>
main( )
{ int n, i,k;
  scanf("%d",&n);
  k=sqrt((double)n);
  for(i=2; i<=k; i++)
    if(n%i==0) break;
  if(i>=k+1) printf("%d is\n",n);
  else printf("%d is not\n",n);
}
```




§ 6 break和continue语句

二、继续语句

continue;

结束本次循环，开始下一次循环

```
#include <stdio.h>
main( )
{ int i;
  for(i=1; i<= 10; i++)
    printf("%3d",i);
}
```

1 2 3 4 5 6 7 8 9 10

```
#include <stdio.h>
main( )
{ int i;
  for(i=1; i<= 10; i++)
  { if(i%2!=0) continue;
    printf("%3d",i);
  }
}
```

2 4 6 8 10



§ 6 break和continue语句

二、继续语句

例：将100~300之间能被3整除的数,以每行10个的形式输出出来

```
#include <stdio.h>
main( )
{ int n, count=0;
  for(n=100; n<=300; n++)
  { if( n%3!=0) continue;
    printf("%5d",n);
    count++;
    if( count%10==0) printf("\n");
  }
}
```



§ 6 break和continue语句

三、break和continue的异同

**break 是结束整个循环；
continue是结束本次循环**

**break既可用于循环也可用于
switch中；continue只能用于
循环**

**break和continue仅适用于
while、do-while、for构成的
循环，不适用if-goto循环**



§ 7 循环嵌套技术

例1：编程打印乘法表第一行

```
#include <stdio.h>

main( )
{ int m,a,b;
  a=1;
  for(b=1;b<=9;b++)
  { m=a*b;
    printf("%d*%d=%-5d", a,b,m);
  }
}
```

例2：编程打印整个乘法表

```
#include <stdio.h>

main( )
{ int m,a,b;
  for(a=1;a<=9;a++)
  { for(b=1;b<=9;b++)
    { m=a*b;
      printf("%d*%d=%-5d",
              a,b,m);
    } /*打印第a行*/
    printf("\n");
  }
}
```



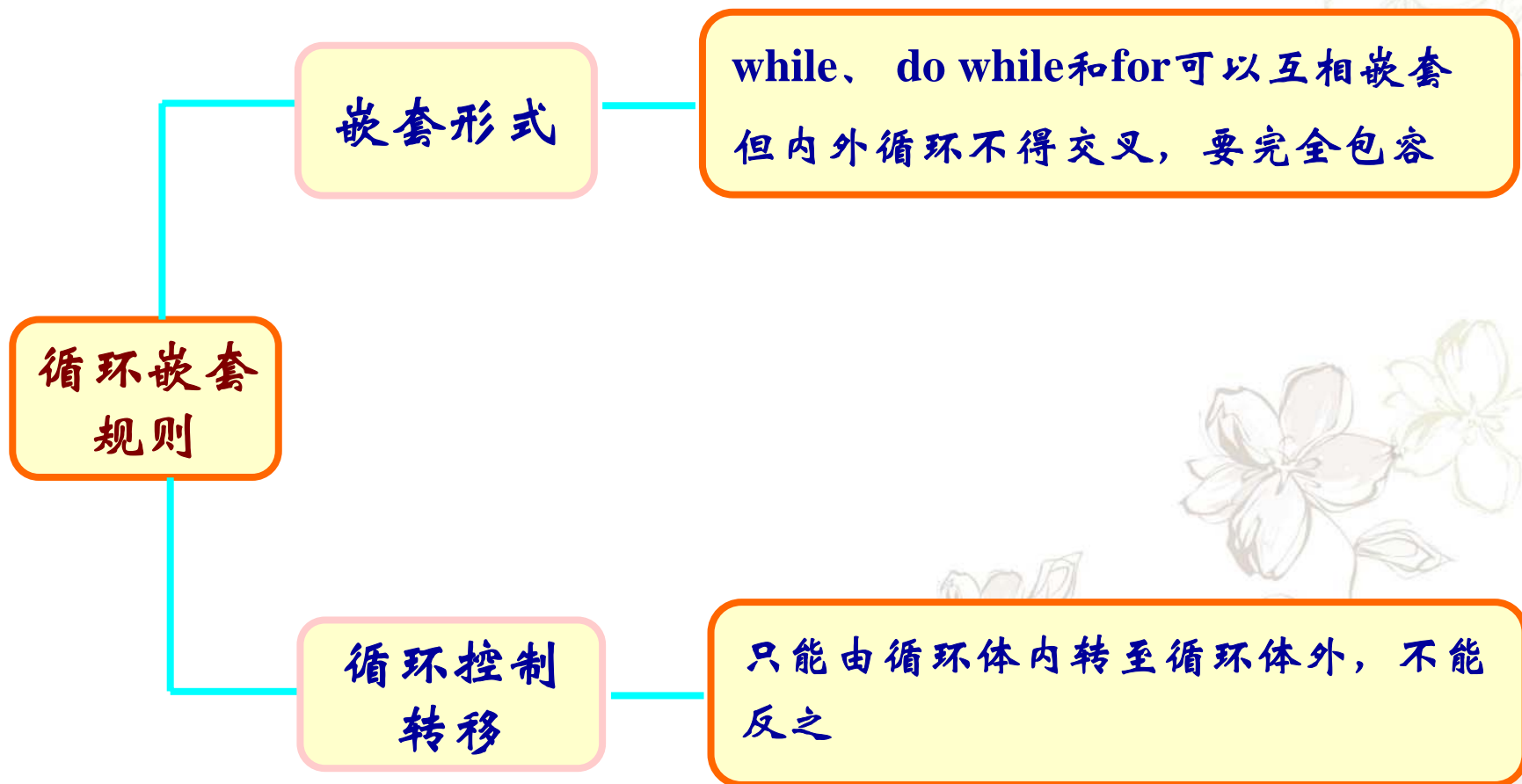
§ 7 循环嵌套技术

例3：编程打印2张乘法表

```
main( )  
{ int m,a,b,k;  
  for(k=1;k<=2;k++)  
  {for(a=1;a<=9;a++)  
    { for(b=1;b<=9;b++)  
      { m=a*b;  
        printf("%d*%d=%-5d", a,b,m);  
      }  
      printf("\n");  
    }  
    printf("\f");  
  } }
```



§ 7 循环嵌套技术





§ 8 举例

例1：C语言中while和do-while循环的主要区别是_____。

- A. do-while的循环体至少无条件执行一次
- B. while的循环控制条件比do-while的严格
- C. do-while允许从外部转到循环体内
- D. do-while的循环体不能是复合语句

A



§ 8 举例

例2：下列程序段**不是死循环**的是__C__。

A. `int i=100;`

```
while(1) { i=i%100+1;  
           if(i>100) break; }
```

B. `for(; ;);`

C. `int k=0;`

```
do{ ++k; } while(k>=0);
```

D. `int s=36;`

```
while(s); --s;
```

例3：以下**是死循环**的程序段是__C__。

A. `for(i=1; ;)`

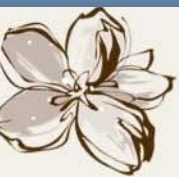
```
{ if(++i%2==0) continue;  
  if(++i%3==0) break; }
```

B. `i=32767;`

```
do { if(i<0) break; }while(++i);
```

C. `for(i=1; ;) if(++i<10) continue;`

D. `i=1; while(i--);`



§ 8 举例


例4：编程求e之值： $e \approx 1 + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$

【解法1：一重循环】

```
#include <stdio.h>
main( )
{ int n, i, t;
  float e;
  e=0.0, t=1;
  scanf("%d", &n);
  for( i=1; i<=n; i++)
  { t*=i;
    e+=1./t;
  }
  printf("e=%.2f\n", e); }
```

【解法2：双重循环】

```
#include <stdio.h>
main( )
{ int n, t, i, j;
  float e=0.;
  scanf("%d", &n);
  for( i=1; i<=n; i++)
  { t=1;
    for(j=1; j<=i; j++) t*=j;
    e+=1./t;
  }
  printf("e=%.2f\n", e); }
```



$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{x^{4n-1}}{(4n-1)!} - \frac{x^{4n+1}}{(4n+1)!} + \dots$$

例6：编程求sin(x)之值，要求误差小于给定的eps

```
#include <math.h>
main( )
{ int i=1,j=0,k,t;
  double x,eps,s,y=0.0, y0;
  scanf("%lf%lf",&x,&eps);
  do { y0=y;
      t=s=1;
      for(k=1;k<=i;k++)
        { t*=k; s*=x;}
      if(j%2!=0) y=y-s/t; else y=y+s/t;
      i+=2,j++;
    } while(fabs(y-y0)>eps);
  printf("%d\tsin(%f)=%f\n",j,x,y); }
```



本章总结

相关语法

- if和goto技术以及特点
- while、do-while、for的构造技术及流程
- 无条件转移、间断以及继续技术
- 循环嵌套技术(要求到3重嵌套)

循环开发技术

经典算法

- 求累加和、累乘(阶乘、幂次等)的算法
- 求质数的算法
- 求最大公约数(辗转相除算法)和最小公倍数
- 数列生成技术
- 搜索一批数中的特定数(最大、最小、次大等)
- 数位截取技术
- 级数求解、求某级数的满足精度 ξ 的值
- 图形输出以及换行技术



经典算法专题讨论

1. 有大小杂乱无章的10个数，编程打印其中最大数max

2. 有大小杂乱无章的10个数，编程打印其中最大数和最小数

```
main( )
{ int i,num,max;
  scanf("%d",&num);
  max=num;
  for(i=2;i<=10;i++)
  { scanf("%d",&num);
    if(num>max) max=num;
  }
  printf("\n max=%d\n",max);
}
```

```
main( )
{ int i,num,max,min;
  scanf("%d",&num);
  max=min=num;
  for(i=2;i<=10;i++)
  { scanf("%d",&num);
    if(num>max) max=num;
    else if(num<min) min=num;
  }
  printf("max=%d\t min=%d\n",
        max,min);
}
```



经典算法专题讨论

3. 有大小杂乱无章的10个数，编程打印其中最大数和最小数，并打印它们的位置。

```
main( )
{ int i,num,max;
  scanf("%d",&num);
  max=num;
  for(i=2;i<=10;i++)
  { scanf("%d",&num);
    if(num>max) max=num;
  }
  printf("\n max=%d\n",max);
}
```

```
main( )
{ int i,num,max,min;
  scanf("%d",&num);
  max=min=num;
  for(i=2;i<=10;i++)
  { scanf("%d",&num);
    if(num>max) max=num;
    else if(num<min) min=num;
  }
  printf("max=%d\t min=%d\n",
        max,min);
}
```




经典算法专题讨论

3.有大小杂乱无章的10个数，编程打印其中最大数和最小数，并打印它们的位置。

```
main( )
{ int i,num,max,min,locmax,locmin;
  scanf("%d",&num);
  max=min=num; locmax=locmin=1;
  for(i=2;i<=10;i++)
  { scanf("%d",&num);
    if(num>max) { max=num; locmax=i;}
    else if(num<min) {min=num; locmin=i;}
  }
  printf("\n max=%d,location:%d\t",max,locmax);
  printf("min=%d,location:%d\n",min,locmin);
}
```

思考题：编程打印10个数中的最大数max和次大数secmax