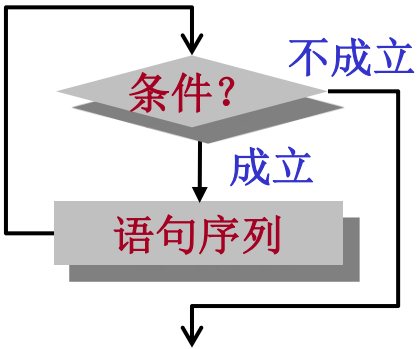
已知条件循环

夏秦

语句格式

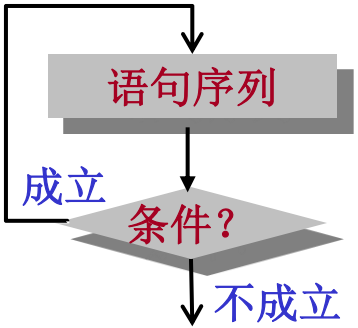
```
while （表达式）
{
   语句序列
}
```

循环体




```
do
{
   语句序列
} while （表达式）;
```

循环体



例：求e值

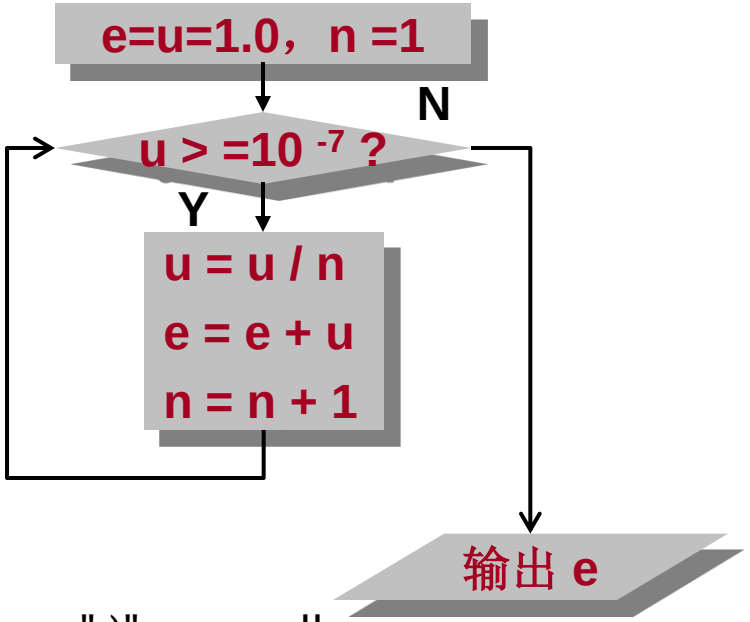
编写程序，使用下列级数近似计算e值，直到最后一个通项$<10^{-7}$为止。

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \ldots + \frac{1}{n!} + \ldots$$

$$u_i = \frac{1}{i!} = \frac{1}{(i-1)!} / i = u_{i-1} / i$$

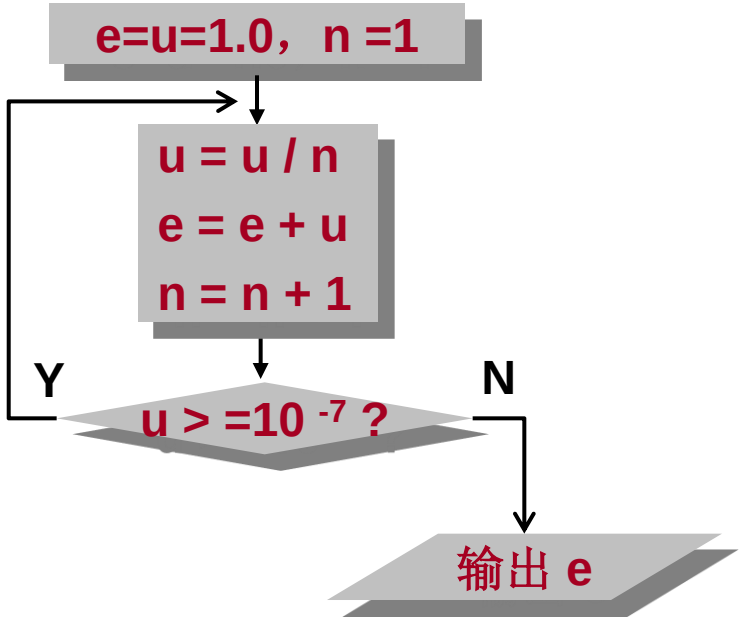
求e值的程序1

```
#include <iostream>
using namespace std;
int main()
{
        double e=1.0,u = 1.0;
        int n = 1;
        while(u >= 1.0E-7)
        {
                u = u/n;
                e = e+u;
                n = n+1;
        }
        cout << "e = " << e << " ( n = " << n << " )" << endl;
        return 0;
}
```



求e值的程序2

```
#include <iostream>
using namespace std;
int main()
{   double e=1.0,u = 1.0;
    int n= 1;
    do
    {       u = u/n;
            e = e+u;
            n = n+1;
    }while(u>=1.0E-7);
    cout << "e = " << e << " ( n = " << n << " )" << endl;
    return 0;
}
```



本例学到

- 1.while语句

  while(u >= 1.0E-7)

  {…}

- 2.do-while语句

  do

  {…

  }while(u >= 1.0E-7)

- 3.通项

  u = u/n;

- 4.累加和

  e = e+u;

例：计算实数n次方根

编写程序，能够根据输入的实数x和n，计算x的n次方根。

具体要求：

- 输入0 0时，程序结束
- 当(x<0且n<=0)或(x<=0且1/n不为整数)时，显示“输入错误”并允许用户继续输入
- 否则计算并显示x的n次方根并允许用户继续输入

计算实数n次方根的程序

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
{   double x,n;
    while(1)
    {       cin>>x>>n;
            if(x==0&&n==0)
                    {cout<<"Program terminated"<<endl;          break;}
            else
                    if((x<0&&n<=0)||(x<0&&1/n!=int(1/n)))
                    {cout<<"error reinput"<<endl;          continue;}
            cout<<x<<"\t"<<n<<"th root"<<pow(x,1.0/n)<<endl;}
    return 0; }
```

（1）break能够跳出所在位置最近的一层循环

（2）continue能够跳过后续语句，开始新一轮的循环

本例学到

- 1.While语句

  while(1)

  {…}
- 2.break语句
- 3.continue语句
- 4.分支语句

  if((x<0&&n<=0)||(x<0&&1/n!=int(1/n)))
- 5.pow函数

  pow(x,1.0/n)

问题

- 在求e值的程序中，能否将通项u的类型由双精度改为整型，初值由1.0改为1？为什么？
- 请总结break语句在多路分支和循环语句中的作用。
- 在什么情况下，可以使用死循环？