```
function I=AdaptiveSimpsonC(f,a,b,eps) %Cursive
format long %type AdaptiveSimpsonC('x*x*sin(x)',1,pi) for test
if(nargin==3)
    eps=1.0e-8;     %缺省精度为0.00000001
end;
I=double(quadSimpson(f,a,b,eps));
function q=quadSimpson(f,a,b,eps)
flag=0;
SumI=0;
pt(1,1)=a;
pt(1,2)=b;
while flag==0
 flag=1;
 n=1;
 for i=1 : size(pt,1)
 a=pt(i,1);
 b=pt(i,2);
 QA=IntSimpson(f,a,b);
 QLeft=IntSimpson(f,a,(a+b)/2);
 QRight=IntSimpson(f,(a+b)/2,b);
    if(abs(QLeft+QRight-QA)<=15*eps*(b-a))
        Ipart=double(QLeft+QRight);
        SumI=SumI+Ipart;
    else
        flag=0;
        pttemp(n,1)=a;
        pttemp(n,2)=(a+b)/2;
        pttemp(n+1,1)=(a+b)/2;
        pttemp(n+1,2)=b;
        n=n+2;
    end
 end
pt=pttemp;
pttemp=0;
end
q=SumI;
function I = IntSimpson(f,a,b)
format long
       I=((b-a)/6)*(subs(sym(f),findsym(sym(f)),a)+...
           4*subs(sym(f),findsym(sym(f)),(a+b)/2)+...
           subs(sym(f),findsym(sym(f)),b));
```

```
function I=AdaptiveSimpson(f,a,b,eps) %Recursive
format long  %type AdaptiveSimpson('x*x*sin(x)',1,pi) for test
if(nargin==3)
    eps=1.0e-8;     %缺省精度为0.00000001
end;
I=double(quadSimpson(f,a,b,eps));

function q=quadSimpson(f,a,b,eps)
QA=IntSimpson(f,a,b);
QLeft=IntSimpson(f,a,(a+b)/2);
QRight=IntSimpson(f,(a+b)/2,b);
if(abs(QLeft+QRight-QA)<=15*eps*(b-a))
    q=QLeft+QRight;
else
    q=quadSimpson(f,a,(a+b)/2,eps)+quadSimpson(f,(a+b)/2,b,eps);
end
function I = IntSimpson(f,a,b)
format long
       I=((b-a)/6)*(subs(sym(f),findsym(sym(f)),a)+...
           4*subs(sym(f),findsym(sym(f)),(a+b)/2)+...
           subs(sym(f),findsym(sym(f)),b));
```