```
function term = series_sum(N,acc,chd)

partial_sums = partialsums(N);
display_partialsums(partial_sums);
term = noterm(acc,chd);
displayresults(partial_sums,term,acc);

end


function app = partialsums(N)
n=1;
sum(1) = 1/((2*n-1)^2*(2*n+1)^2);
app(1) = sqrt(sum(1)*16+8);
for i=2:N
    sum(i) = sum(i-1) + 1/((2*i-1)^2*(2*i+1)^2);
    app(i) = sqrt(sum(i)*16+8);
end
end


function display_partialsums(ps)
k=1:1:length(ps);
plot(k,ps,'*')
xlabel('N')
ylabel('Partial Sums')
title('Partial Sums against N')
end

function t = noterm(acc,chd)
i=1;
sum(i) = 1/((2*i-1)^2*(2*i+1)^2);
app(1) = sqrt(sum(1)*16+8);
tv = chop(pi,6);
apprx = chop(app(1),chd);

while abs(tv-apprx) > acc
    i = i+1;
    sum(i) = sum(i-1) + 1/((2*i-1)^2*(2*i+1)^2);
    app(i) = sqrt(sum(i)*16+8);
    apprx = chop(app(i),chd);
end
t = i;
end


function displayresults(ps,term,acc)
p = 1:1:length(ps);
table = [p' ps'];
disp('********************************')
disp('--------------------------------')
disp('     N       Partial Sums       ')
disp('--------------------------------')
disp(table)
disp('--------------------------------')
```

```
fprintf('\n')
fprintf('No of terms needed to get %5.6f accuracy = %d\n',acc,term)
disp('********************************')
end
```