1)...

```
function integral = trapezoidal(f, a, b, n)
    h = (b-a)/n;
    result = 0.5*f(a) + 0.5*f(b);
    for i = 1:(n-1)
        result = result + f(a + i*h);
    end
    integral = h*result;
end
```

```
function application()
   v = @(t) 1-t-4*(t^3)+2*(t^5);
   n = input('n: ');
   numerical = trapezoidal(v, 0, 4, n);

   % Compare with exact result
   V = @(t) t-(t^2)/2-(t^4)+(t^6)/3;

   exact = V(4) - V(0);

   error = ((exact - numerical)/exact)*100;

   fprintf('n = %d\n', n);
    fprintf('numerical = %.16f\n',numerical);
     fprintf('error percent: %f\n',  error);
end
```

```
>> application
n: 2
n = 2
numerical = 1852.0000000000000000
error percent: -67.551267
>> application
n: 3
n = 3
numerical = 1447.7201646090534000
error percent: -30.975889
>> application
n: 4
n = 4
numerical = 1300.0000000000000000
error percent: -17.611580
```

c)

```
function I = simpson(f, a, b, n)
h = (b - a) / n;
ff1 = 0; ff2 = 0;
for i = 1:2:n-1;
    x  = (a + i* h);
    ff1 = ff1 + f(x);
end
for i = 2:2:n-2;
    x = (a + i* h);
     ff2 = ff2 + f(x);
end
I = (h/3)*(f(a) + 4*ff1 +2* ff2 + f(b));
end
```

```
function application2()
  v = @(t) 1-t-4*(t^3)+2*(t^5);
  n = input('n: ');
  numerical = simpson(v, 0, 4, n);

  % Compare with exact result
  V = @(t) t-(t^2)/2-(t^4)+(t^6)/3;

  exact = V(4) - V(0);

  error = ((exact - numerical)/exact)*100;

  fprintf('n = %d\n', n);
   fprintf('numerical = %.16f\n',numerical);
    fprintf('error percent: %f\n',  error);
end
```

```
>> application2
n: 2
n = 2
numerical = 1276.000000000000000
error percent: -15.440290
>> application2
n: 3
n = 3
numerical = 793.090077320530300
error percent: 28.248787
>> application2
n: 4
n = 4
numerical = 1116.000000000000000
error percent: -0.965018
fx >>
```

```
function I = SimpThreeEight(f, a, b, n)
h = (b-a)/n;
S =feval(f,a);
for i = 1:3:n-2
   x(i) = a + h*i;
   S = S + 3*feval(f, x(i));
end
for i = 2:3:n-1
   x(i) = a + h*i;
   S = S + 3*feval(f, x(i));
end
for i = 3 : 3: n-3
x(i) = a + h*i;
S = S + 2*feval(f, x(i));
end
S = S + feval(f, b);
I = 3*h*S/8;
```

```
function application2()
  v = @(t) 1-t-4*(t^3)+2*(t^5);
  n = input('n: ');
  numerical = SimpThreeEight(v, 0, 4, n);

  % Compare with exact result
  V = @(t) t-(t^2)/2-(t^4)+(t^6)/3;

  exact = V(4) - V(0);

  error = ((exact - numerical)/exact)*100;

  fprintf('n = %d\n', n);
   fprintf('numerical = %.16f\n',numerical);
    fprintf('error percent: %f\n',  error);
end
```

```
>> application2
n: 2
n = 2
numerical = 1342.5000000000000000
error percent: -21.456574
>> application2
n: 3
n = 3
numerical = 1181.1851851851850000
error percent: -6.862351
>> application2
n: 4
n = 4
numerical = 703.8750000000000000
error percent: 36.320115
```

2) i)

```
%  Composite Trapezoidal rule
function application3()
  m=[1,1.5,2];
  n=[2,2.5,3];
fprintf('segments\t values\n');
  for i=0:2
   u=i+2;
   v = @(t) (t^(m(i+1)-1))*((1-t)^(n(i+1)-1));
   numerical = trapezoidal(v, 0, 1, u);
   fprintf('  %d\t       %.16f\n',u,numerical);
  end
end
```

```
>>
>> application3
segments      values
  2         0.5000000000000000
  3         0.1571348402636772
  4         0.0781250000000000
>>
```

ii)

```
function application4()
  m=[1,1.5,2];
  n=[2,2.5,3];
  fprintf('segments\t values\n');
  for i=0:2
   u=i+2;
   v = @(t) (t^(m(i+1)-1))*((1-t)^(n(i+1)-1));
   numerical = simpson(v, 0, 1, u);
   fprintf('  %d\t       %.16f\n',u,numerical);
  end
end
```

```
>> application4
segments      values
  2         0.5000000000000000
  3         0.1396754135677131
  4         0.0833333333333333
>>
```