Assignment 6

Name – Abhishek Agrahari

Roll Number – 190123066

Question 1-

Code

```
clc;
clear;
f = @(x) sin(2*x+2)/(x + 1);

integral_val = 5/9*f(-sqrt(3/5)) + 8/9*f(0) + 5/9*f(sqrt(3/5));
fprintf('Required value of the integral is %.15f\n', integral_val);
```

Ouput

```
Required value of the integral is 1.758022025436357
```

Observations

- Replacing $x = 2.t + 2$ gives the integral as $\int_{-1}^{1} \frac{\sin(2.t+2)}{t+1} dt$.
- Now taking $f(x) = \frac{\sin(2.t+2)}{t+1}$ and using the gaussian quadrature rule given in the question gives required value of the integral.

Question 2-

Code

```
clc;
clear;
W1 = 0.8535533903;
W2 = 0.1464466092;
f = @(x) x^3;
x1 = 0.5857864376;
x2 = 3.414213562;
integral_val = W1*f(x1)+W2*f(x2);
fprintf('Required value of the integral is %.15f\n', integral_val);
```

Ouput

```
Required value of the integral is 5.999999989779173
```

Observations

- Using the formula $\int_0^\infty e^{-x}.f(x)dx = W_1.f(x_1) + W_2.f(x_2)$ value of the integral can be found.
- Here $f(x) = x^3, W_1 = 0.8535533903,\ W_2 = 0.1464466092, x_1 = 0.5857864376$ and $x_2 = 3.414213562$.
- Value of W1 and W2 can be found as –
  - $W_1 = \int_0^\infty \frac{x-(2+\sqrt{2})}{(2-\sqrt{2})-(2+\sqrt{2})}.e^{-x}.dx = \int_0^\infty \frac{x-(2+\sqrt{2})}{-2\sqrt{2}}.e^{-x}.dx = 0.8535533903$
  - $W_2 = \int_0^\infty \frac{x-(2-\sqrt{2})}{(2+\sqrt{2})-(2-\sqrt{2})}.e^{-x}.dx = \int_0^\infty \frac{x-(2-\sqrt{2})}{2\sqrt{2}}.e^{-x}.dx = 0.1464466092$

Question 3-

Code

```
clc;
clear;
f = @(x,y) (x*y-y^2)/(x^2);
act_val = @(x) x/(0.5 + log(x));
err = @(a,b) abs(a-b);
euler_method = @(x,y,h) y + h*f(x,y);

y_euler=2;
y_rk2 = 2;
y_rk4 = 2;
h = 0.1;
fprintf('\t\t\t\t\t\t\t\t Table 1: With h = 0.1 \nx    \t    EM   \t    Error(EM)  \t\t   RK2 
\t   Error(RK2)  \t      RK4    \t   Error(RK4)\n');
for x = 1:h:2
    if x == 1.2 || x== 1.4 || x == 1.6 || x == 1.8 || x== 2
        fprintf('%.1f  %.12f  %.12f  %.12f  %.12f  %.12f  %.12f\n', x, y_euler, err(y_euler, 
act_val(x)), y_rk2, err(y_rk2, act_val(x)), y_rk4, err(y_rk4, act_val(x)));
    end
    y_euler = euler_method(x,y_euler,h);

    k1_rk2 = h*f(x,y_rk2);
    k2_rk2 = h*f(x + h, y_rk2 + k1_rk2);
    y_rk2 = y_rk2 + (k1_rk2 + k2_rk2)/2;

    k1_rk4 = h*f(x,y_rk4);
    k2_rk4 = h*f(x + h/2, y_rk4 + k1_rk4/2);
    k3_rk4 = h*f(x + h/2 , y_rk4 + k2_rk4/2);
    k4_rk4 = h*f(x + h, y_rk4 + k3_rk4);
    y_rk4 = y_rk4 + (k1_rk4 + 2*k2_rk4 + 2*k3_rk4 + k4_rk4)/6;
end

y_euler=2;
y_rk2 = 2;
y_rk4 = 2;
h = 0.01;
fprintf('\n\t\t\t\t\t\t\t\t\t Table 2: With h = 0.01 \nx    \t    EM   \t    Error(EM)  \t\t 
RK2   \t   Error(RK2)  \t      RK4    \t   Error(RK4)\n');
for x = 1:h:2
    if x == 1.2 || x== 1.4 || x == 1.6 || x == 1.8 || x== 2
```

```
        fprintf('%.1f  %.12f  %.12f  %.12f  %.12f  %.12f  %.12f\n', x, y_euler, err(y_euler,
act_val(x)), y_rk2, err(y_rk2, act_val(x)), y_rk4, err(y_rk4, act_val(x)));
    end

    y_euler = euler_method(x,y_euler,h);

    k1_rk2 = h*f(x,y_rk2);
    k2_rk2 = h*f(x + h, y_rk2 + k1_rk2);
    y_rk2 = y_rk2 + (k1_rk2 + k2_rk2)/2;

    k1_rk4 = h*f(x,y_rk4);
    k2_rk4 = h*f(x + h/2, y_rk4 + k1_rk4/2);
    k3_rk4 = h*f(x + h/2 , y_rk4 + k2_rk4/2);
    k4_rk4 = h*f(x + h, y_rk4 + k3_rk4);
    y_rk4 = y_rk4 + (k1_rk4 + 2*k2_rk4 + 2*k3_rk4 + k4_rk4)/6;
end
```

Ouput

Table 1: With h = 0.1

x	EM	Error(EM)	RK2	Error(RK2)	RK4	Error(RK4)
1.2	1.695867768595	0.062833811403	1.758692713155	0.000008866843	1.758691652629	0.000009927369
1.4	1.604772056962	0.068923660250	1.673399315464	0.000296401749	1.673686920372	0.000008796840
1.6	1.581794814460	0.067683550115	1.649039289116	0.000439075459	1.649470662745	0.000007701830
1.8	1.589332450766	0.065403774707	1.654231050648	0.000505174825	1.654729268050	0.000006957423
2.0	1.612974833418	0.063264303367	1.675703425197	0.000535711588	1.676232685524	0.000006451262

Table 2: With h = 0.01

x	EM	Error(EM)	RK2	Error(RK2)	RK4	Error(RK4)
1.2	1.753400705109	0.005300874889	1.758705453452	0.000003873454	1.758701579444	0.000000000554
1.4	1.667608674194	0.006087043019	1.673696627471	0.000000910259	1.673695716701	0.000000000511
1.6	1.643378260993	0.006100103582	1.649477496471	0.000000868104	1.649478364121	0.000000000454
1.8	1.648779663189	0.005956562284	1.654734406166	0.000001819307	1.654736225061	0.000000000412
2.0	1.670443662019	0.005795474766	1.676236797789	0.000002338996	1.676239136403	0.000000000383

Observations

- Error decreases as we go from Euler method to RK2 (second-order Runge Kutta method) and RK2 to RK4 (fourth-order Runge Kutta method).
  - This observation is in accordance with the order of their errors as given below –
    - Euler Method : $O(h^2)$
    - Second-order Runge Kutta Method : $O(h^3)$
    - Fourth-order Runge Kutta Method : $O(h^5)$
- Error decreases with decrease in value of h.
  - This is because error is proportional to some positive power of h.