2. Design a C program to implement R-K method.

Runge-Kutta method: To solve ordinary differential Equations.

To solve $\frac{dy}{dx} = f(x, y) = y' = \dot{y}$

given $y(x_0) = y_0$. y_1, y_2 $y_1 = y(x_1)$ & $y_2 = y(x_2)$

$x_1 = x_0 + h$ h – Step size – interval length.

$h = x_1 - x_0 = x_2 - x_1 = x_3 - x_2 \dots$

The fourth order R-K method algorithm:

$k_1 = h f(x, y)$

$k_2 = h f(x + h/2, y + k_1/2)$

$k_3 = h f(x + h/2, y + k_2/2)$

$k_4 = h f(x + h, y + k_3)$

$k = (k_1 + 2k_2 + 2k_3 + k_4)/6$.

$y(x + h) = y(x) + k$.

x_0 and y_0. To find $y_1 = y(x_1)$ where $x_1 = x_0 + h$.

$k_1 = h\, f(x_0, y_0)$

$k_2 = h\, f(x_0 + h/2\,;\ y_0 + k_1/2)$

$k_3 = h\, f(x_0 + h/2\,;\ y_0 + k_2/2)$

$k_4 = h\, f(x_0 + h,\ y_0 + k_3)$

$$K = (k_1 + 2k_2 + 2k_3 + k_4)/6.$$

$$y_1 = y_0 + K$$

Code :

1. Start
2. Define function $f(x, y)$
3. Read the values of (x, y) at the starting point or initial conditions of x & y i.e., (x_0, y_0). number of steps (n) and calculation point or end point of x. x_n

4. Calculate the step size $h = (x_n - x_0)/n$

5. Set $i = 0$.

6. Loop

$k_1 = h * f(x_0, y_0);$

$k_2 = h * f(x_0 + h/2, y_0 + k1/2);$

$k_3 = h * f(x_0 + h/2, y_0 + k_2/2);$

$k_4 = h * f(x_0 + h, y_0 + k_3);$

$k = (k_1 + 2 * k_2 + 2 * k_3 + k_4)/6;$

$y_n = y_0 + k;$

$i = i + 1$

$x_0 = x_0 + h;$

$y_0 = y_n$

While $i < n$

7. Display y_n as a Result.

8. Stop.

Find $y(1.1), y(1.2)$ if $dy/dx = x^3 + y/2$ using R-K method of 4th-order where $y(1) = 2$ through C. programming.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
/* define f(x,y)    (pow(x,3) + (y/2)) */
float x, y;
# define f(x,y) (pow(x,3)+(y/2))
int main ()
{ float x0, y0, h, xn, yn, k1, k2, k3, k4, k;
int i, n;
printf("Enter initial Condition \n");
printf(" x0 = ");
scanf("%f", &x0);
```

```
printf(" y0 = ");
scanf("%f", &y0);
printf("Enter the calculation point xn = ");
scanf("%f", &xn);
printf("Enter the number of steps: ");
scanf("%d", &n);
printf(" The Step-size: h = ");
scanf("%f", &h);
/* Runge kutta method */
for( i=0; i<n; i++)
{   k1 = h* f(x0, y0);
```

```
k2 = h * f((x0 + h/2), (y0 + k1/2));
k3 = h * f((x0 + h/2), (y0 + k2/2));
k4 = h * f((x0 + h), (y0 + k3));
k = (k1 + 2 * k2 + 2 * k3 + k4) / 6;

yn = y0 + k
printf("\n x  %d \t y %d \n", i i);
printf("%0.4f \t %0.4f \n", x0, y0);

x0 = x0 + h;
y0 = yn
} printf("\n x %d, \t y %d \n", n, n);
printf("%0.4f, \t %0.4f.", x0, y0);
getch()
?
```

Enter Initial condition

$x_0 = 1$

$y_0 = 2$

Enter Calculation point $x_n = 1.2$

Enter number of steps : 2

Enter step size 0.1

x_0	y_0
1.0000	2.0000
x_1	y_1
1.1000	2.2214
x_2	y_2
1.2000	2.4913

2) Find $y(0.7)$, $y(0.8)$ if $dy/dx = y - x^2$ using R-K method where $y(0.6) = 1.7379$ through C-programming

x_1 0.7000 y_1 1.8763

x_2 0.8000 y_2 2.0145

$$f(x, y) = y - x^2$$

$f(x,y)$ $\underline{y -}$ $x * x$.

3) obtain the values of y at $x = 0.1, 0.2$ using R-K method of 4th order for the differential Equation $y' = -y$. given $y(0) = 1$. $f(x,y)$ $-y$.

4) Evaluate $y(1.2)$ and $y(1.4)$ given $y' = \dfrac{2xy + e^x}{x^2 + x e^x}$

$y(1) = 0$

$f(x,y)$ ((2 * x * y + exp(x)) / (x * x + x * exp(x)))

$y(1.2) = 0.1402$; $y(1.4) = 0.2705$

5) Determine y at $x = 0.2(0.2)0.6$ by R-K method given $dy/dx = xy+1$, given $y(0) = 0$

$x_0 = 0$ $\qquad$ $y(x_0) = y_0 = 0 = 0$.

$x_1 = 0.2$ $\qquad$ $y_1 = y(0.2) = 2.243$

$x_2 = 0.4$ $\qquad$ $y_2 = y(0.4) = 2.589$

$x_3 = 0.6$ $\qquad$ $y_3 = y(0.6) = 3.072$.

$f(x,y)$ $\qquad$ $x*y+1$