



R. K. GROUP OF COLLEGE

BEHIND KALWAR POLICE STATION, KALWAR, JAIPUR (RAJ.)

MATH BSC III year



S. No.	Name of Experiment.	Page No.	Date of Experiment	Date of Submission	Remarks
1.	C- ગ્રામ લેખક રેખા સમીકરણો કા ડિમાજન વિધિ દ્વારા દલ કરના				
2.	C- ગ્રામ લેખક રેખા સમીકરણો કા મિથ્યા વિધિ દ્વારા દલ કરના				
3.	C- ગ્રામ લેખક ન્યુન રેખાસન વિધિ દ્વારા દલ કરના				
4.	C- ગ્રામ લેખક રંગી કદા વિધિ દ્વારા દલ કરના				
5.	C- ગ્રામ લેખક ટ્રેપિજોઇડલ વિધિ દ્વારા દલ કરના				
6.	C- ગ્રામ લેખક ગામ ઉન્મૂલન વિધિ દ્વારા દલ કરના				

Q=1. C- जौग्राम को लिखकर द्विभाजन ~~सही~~ विधि का प्रयोग करते हुए समीकरण $x^3 - 2x - 5 = 0$ का मूल जो अंतराल $[2, 3]$ में स्थित है। दशमलव के तीन स्थान ज्ञात करें।

Ans → हम जानते हैं कि यदि फलन $f(x)$ दो वास्तविक मानों x_0, x_1 के मध्य सतत हो। तथा $f(x_0)$ और $f(x_1)$ विपरित चिन्ह के हो अर्थात् $f(x_0) \cdot f(x_1) < 0$ हो तो x_0 तथा x_1 के मध्य $f(x) = 0$ का एक मूल अवश्य होगा। यदि माना कि $f(x_0)$ का मान ऋण तथा $f(x_1)$ का मान धन चिन्ह है अब अन्तराल (x_0, x_1) को उनके मध्य स्थित बिंदु $x_2 = \frac{x_0 + x_1}{2}$ द्वारा दो समान भागों में विभाजित किया। यदि $f(x_2) = 0$ है तब $x_2, f(x) = 0$ का सही मूल है। परन्तु यदि $f(x_2) \neq 0$ है तो $f(x) = 0$ का मूल (x_0, x_2) या (x_2, x_1) में किसी एक में विद्यमान होगा। यदि $f(x_2)$ का मान धनात्मक है। तो मूल (x_0, x_2) में स्थित होगा क्योंकि $f(x_0) \cdot f(x_2) < 0$ होगा और यदि $f(x_2)$ का मान ऋणात्मक है तो मूल $[x_2, x_1]$ में स्थित होगा। क्योंकि $f(x_2) \cdot f(x_1) < 0$ होगा।

इसी प्रकार भागों भी जिस अन्तराल में मूल बिंदु हो उसको दो समान भागों में विभाजित करते जाते हैं। जब तक कि मूल का वांछित शुद्धता तक मान ज्ञात नहीं हो जाए।

विभाजन विधि द्वारा समीकरण $f(x)=0$ के मूल ज्ञात करने
का C- प्रोग्राम इस प्रकार है —

/* BISECTION METHOD */

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define f(x) x*x*x - 2.0*x - 5.0 /*
write given equation x/
void main ( )
```

```
{
```

```
float x, x0, x1, x2, y0, y1, y2, ε;
clrscr();
```

```
printf("Input two Initial starting roots/n");
```

```
scanf("%f %f", &x0, &x1);
```

```
y0 = f(x0);
```

```
y1 = f(x1);
```

```
if (y0 * y1 > 0.0)
```

```
{
```

```
printf("Initial values are unsuitable/n");
```

```
} else
```

Teacher's Signature.....


```
while ( fabs (x1 - x0) / x1 ) > e)
```

```
    x2 = (x0 + x1) / 2 ;
```

```
    y2 = f(x2)
```

```
    if (y0 * y2 > 0)
```

```
        x0 = x2 ;
```

```
    else
```

```
        x1 = x2 ;
```

```
} printf ("solution converges to a root");
```

```
printf ( '\n Root of the given  
equation is %8.3f\n' , x2 );
```

```
getch ( );
```

```
}
```

Input :-

Input Two Initial starting roots 2, 3

Input the tolerance value 0.0001

Output :

Solution converge to a root of
the given equation is -2.025

Q-2. C- जोगाम के द्वारा समीकरण $x^2 + x - 1 = 0$ का एक मूल जो अन्तराल $(0, 1)$ में स्थित है, मिथ्या स्थिति विधि की सहायता से दशमलव के तीन स्थानों तक शुद्ध बात कीजिए?

Ans. मिथ्या स्थिति विधि का पुनवृत्ति सूत्र निम्न है—

$$x_{n+1} = \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})} \quad n=1, 2, 3.$$

C- जोगाम इस प्रकार है -

/ * Root of an equation by Regula falsi method * /

```
# include <stdio.h>
# include <conio.h>
# include <math.h>
# define f(x) x*x+x-1 / * Write given equation * /
```

```
Void main ( )
```

```
{ Float x0, x1, x2, f0, f1, f2 e;
```

```
int max-iteration;
```

```
clrscr();
```

```
printf("Input two initial Roots maximum iteration and tolerance value:");
```

```
scanf("%f %f %d %f" &x0, &x1, &max-iteration, &e);
```



```

f(0) = f(x0);
f(1) = f(x1);
if (f0 * f1 > 0.0)

```

```

    print f ("Initial roots are unsuitable
    )

```

```

    else

```

```

    {
        for (i = 1; i <= max-iteration; i++)

```

```

    {
        x2 = (x0 * f1 - x1 * f0) / (f1 - f0);

```

```

        f2 = f(x2)

```

```

        if (fabs(f2) < e)

```

```

        {
            print f ("Solution is converge/n");

```

```

        print f ("Root of the given equation is
            = %.84f /n", x2);

```

```

        max iteration = i;

```

```

        else

```

```

        {
            if (f2 * f(0) < 0)

```

```

                x1 = x2;

```

```

                f1 = f2;

```

```

            }

```

```

            else

```


$$\begin{cases} x_0 = x_2 j \\ f_0 = f_2 j \\ \vdots \\ \vdots \end{cases}$$

getch ();

Input :-

Input Two initial Roots maximum iterations and tolerance value :-

0 j 10

Output :-

Solution is converge

Roots of the given equation is $x = 0.618$

Q3. C- प्रोग्राम के द्वारा समीकरण $x^3 - 3x - 5 = 0$ का मूल अन्तराल $(2, 3)$ में स्थित न्युटन रेफसन विधि द्वारा दशमलव के चार स्थानों तक शुद्ध बात करो।

Ans

न्युटन रेफसन विधि का पुनरावृत्ति सूत्र:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

जहाँ $f'(x_n)$, $f(x_n)$ के सापेक्ष प्रथम अवकलन है।

C- प्रोग्राम

```
/* Newton - Raphson method */
#include <stdio.h>
#include <conio.h>
#include <math.h>
# define f(x) x*x*x - 3*x - 5 /* given
                                equation */
# define f'(x) 3*x*x - 3 /* Diff. of equation */
```

```
void main()
```

```
{
    long float x0, x1, to y1, x1, e;
    long int j, maxitor;
    clrscr();
```

```
printf("\n Input initial root max. iterations
and tolerance value : (n");
```



```
scanf ("%f", &f) %f; %d %d; %s %s, &maxiter &e;
for (i = j; i < monitor; i++)
```


Output $\Rightarrow$

Solution Converges to a root
number of Iterations $= 4$

Root of the given equation is $= 2.2790$

Q=4 C- प्रोग्राम को लिखते हुए रूंगे कुहा विधि (चतुर्थ कोटि) का उपयोग कर $x=1.5$ पर y का सन्निकटन मान ज्ञात कीजिए यदि - एवं

$$\frac{dy}{dx} = xy^2 - \frac{y}{x} \quad \text{जबकि } x=1.42, y=1$$

पद लम्बाई 0.1 है।

यहाँ

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

$$x_0 = 1, y_0 = 1, \text{ तथा } h = 0.1$$

रूंगे कुहा विधि (चतुर्थ कोटि) का सूत्र :-

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + h/2, y_n + k_1/2)$$

$$k_3 = hf(x_n + h/2, y_n + k_2/2)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

$$K = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$y_{n+1} = y_n + K \quad h = 0, 1, 2, 3, \dots$$

रूंगे कुहा विधि का C- प्रोग्राम

* Runge - Kutta method of fourth order

include <stdio.h>

include <conio.h>


```
void main()
```

```
{
```

```
float f(float x, float y); /* function proto  
+ pe x/
```

```
float x0, y0, h, xn, k, k2, k3, k4, x, y;
```

```
int i, n;
```

```
clrscr();
```

```
print ("Enter the initial value of x and y;")
```

```
scanf ("%f %f", &x0, &y0);
```

```
print f("\n Enter x at which y is  
required: ")
```

```
scanf ("%f", &xn);
```

```
print f("\n Enter step size;");
```

```
n(int) = ((xn - x0) / h + 0.5);
```

```
x = x0;
```

```
y = y0;
```

```
print f("\n in step 'x' y/n");
```

```
for (i = 1) i <= n; i++)
```

```
k1 = h * f(x, y); /* function call */
```

```
k2 = h * f(x + h/2, y + k1/2); /* function  
call */
```

```
k3 = h * f(x + h/2, y + k2/2); /* function  
call */
```

```
k4 = h * f(x + h, y + k3); /* function call */
```



```

K = (K1 + 2.0 * (K2 * K3) + K4) / 16.0;
X = X + h;
Y = Y + K;
printf("%f 3d %f 10.3 f %f 10.3 f\n", X, Y);
printf("\n Value of y at x = %f\n", X, Y);

getch();

```

```

/* function definition f() */
float f(float x, float y)

```

```

return (x * y * y - y/x) / /* R.H.S of the given
equation */

```

Input :-

Enter the initial value of x and y : 1, 1

Enter x at which y is required : 1.5 enter step size 0.1

Output :-

Step	X	Y
1	1.1	1.0101
2	1.2	1.0417
3	1.3	1.0985
4	1.4	1.1905
5	1.5	1.3333

Value of y at x = 1.5000 is 1.3334

Q-5 C- प्रोग्राम को लिखते हुये समाकलन का मान ट्रेपिजोइडल नियम द्वारा बता दीजिए।

$$\int_0^6 \frac{dx}{1+x^2}$$

Ans ट्रेपिजोइडल नियम का सूत्र :-

$$\int_{x_0}^{x_n} y \cdot dx = h \left[\frac{1}{2} (y_0 + y_n) + (y_1 + y_2 + y_3 + \dots + y_{n-1}) \right]$$

जहाँ

$$h = \frac{x_n - x_0}{n} \quad \text{या} \quad n = \frac{x_n - x_0}{h}$$

ट्रेपिजोइडल का C- प्रोग्राम :-

/ * Trapezoidal Rule */

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
    float f(float x); /* function prototype */
```

```
float x0, xn, h, sum, result;
```

```
int i, n;
```

```
clrscr();
```

```
printf("Enter the low and upper limits of integral");
```

```
};
```


Scan f (" %. f %. f " & x0, & xn);

Scan f (" %. f " & h);

h = int (xn - x0) / n;

Sum = (f(x0) + f(xn)) / 2.0; /* function

for () = 1; i < n; i++)

{

Sum = Sum + f(x0 + i * h); /* function
call */

}

result = Sum * h;

printf (" n/n value of integral using
Trapezoidal rule is %. f/n/n result

getch ();

}

/* function definition f() */
float f (float x)

{

return (1.0 / (1.0 + x * y)); integer.

Input :-

Enter the lower and upper limits

integral 0.0.6.0

enter the segment with : 1.0

Output :->

rule Value of integral using trapezoidal
 is 1.410799.

Q-6 गॉस उन्मूलन विधि का प्रयोग करते देखिये समीकरणों की निम्नलिखित प्रणाली के लिए C- प्रोग्राम लिखें?

$$2x + 8y + 2z = 14$$

$$x + 6y - z = 13$$

$$2x - y + 2z = 5$$

Ans: गॉस उन्मूलन विधि का C- प्रोग्राम -

/* C program: Gauss-Elimination Method */

#include <stdio.h>

#include <conio.h>

void main()

{

float a[10][10], x[10], temp;

int i, j, k, n;

clrscr();

printf("\n Enter the Number of equation :");

scanf("%d", &n);

printf("\n The Number of equation are : %d\n", n);

printf("\n Now Enter augmented (coefficient of equo) matrix\n");

for (i=1; i<=n; i++)

for (j=1; j<=n+1; j++)

scanf("%f", &a[i][j]);

/* calculate upper triangular matrix */

for (i=1; i<=n-1; i++)


```

{
    temp = 0.0;
    for (j = i+1; j <= n; j++)

```

```

{
    temp = a[j][i] / a[i][i];
    for (k = 1; k <= n+1; k++)
        a[j][k] = a[j][k] - (temp * a[i][k]);
}
}

```

```

}
/* print upper triangular matrix */
printf("\n Upper Triangular matrix is : \n");
for (i = 1; i <= n; i++)

```

```

{
    for (j = 1; j <= n+1; j++)
        printf("%7.2f", a[i][j]);
    printf("\n");
}

```

```

/* Evaluate unknowns by back substitution */
x[n] = a[n][n+1] / a[n][n];
for (i = n-1; i >= 1; i--)

```

```

{
    x[i] = a[i][n+1];
    for (j = i+1; j <= n; j++)
        x[i] = x[i] / a[i][i];
}
}

```



```
printf("\n Solution of the equation is : \n");  
for (i=1; i<=n; i++)  
    printf("x [%d] = %f \n", i, x[i]);  
  
getch();
```

Input :-

Enter the Number of equation : 3

The Number of Equation are : 3

Now Enter augmented (coefficients of eqns) matrix

2 8 2 14

1 6 -1 13

2 -1 2 5

Output :->

Upper Triangular Matrix is :

2.00 8.00 2.00 14.00

0.00 2.00 -2.00 6.00

0.00 0.00 -9.00 18.00

Solution of the equation is :

$x[1] = 5.000$

$x[2] = 1.000$

$x[3] = -2.000$