R. K. GROUP OF COLLEGE

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

MATH BSC III year

INDEX

S. No.	Name of Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
1.	C- प्रोग्राम लिखकर रैखिक समीकरणों का विभाजन विधि द्वारा हल करना।				
2.	C- प्रोग्राम लिखकर रैखिक समीकरणों का मिथ्या विधि द्वारा हल करना				
3.	C- प्रोग्राम लिखकर न्युटन रेफसन विधि द्वारा हल करना।				
4.	C- प्रोग्राम लिखकर रंगे कुट्टा विधि द्वारा हल करना।				
5.	C- प्रोग्राम लिखकर ट्रेपिजोइडल विधि द्वारा हल करना।				
6	C- प्रोग्राम लिखकर गॉस उन्मूलन विधि द्वारा हल करना।				

Q=1. C- प्रोग्राम को लिखकर द्विभाजन ~~वाली~~ 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 करो;

Ans :→ हम जानते है कि यदि फलन $f(x)$ दो वास्तविक मानो x_0, x के मध्य सतत हों। तथा $f(x_0)$ और $f(x)$ विपरित चिन्ह के हो अर्थात $f(x_0) \cdot f(x_1) < 0$ हो तो x_0 तथा x 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 भागो में विभाजित किया। यदि $f(x_2) = 0$ है तब x_2, $f(x) = 0$ का सही मूल है। परन्तु यदि $f(x_2) \neq 0$ 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 आगे भी जिस अन्तराल में मूल बिंदु हो उसको दो समान भागो में विभाजित करते जाते है। जब तक कि मूल का वांछित शुद्धता तक मान ज्ञात नहीं हो जाए।

द्विभाजन विधी 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 */
void main()
{
    float x, x0, x1, x2, y0, y1, y2, e;
clrscr();
Printf("Input two Initial stading 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
```

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

    x2 = (x0 + x1) /2 ;
    y2 = f(x2)
    if = (y0 * y2 > 0)
      x0 - x2 ;
  else
      x1 = x2 ;

}  printf ("solution convergers to a rsatin");
printf (') n Root of the given
  equation is %. 8.3f") x.2);

  getch ('1');
}
```

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- प्रोग्राम के द्वारा समीकरण $x^2+x-1=0$ का एक मूल जो अन्तराल (0,1) में स्थित है, मिथ्या स्थिति विधी की सहायता से दशमलव के तीन स्थानों तक शुद्ध ज्ञात कीजिए?

Ans मिथ्या स्थिति विधि का पुनरावृत्ति सूत्र निम्न है:—

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

C- प्रोग्राम 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 %f %d" &x0. &x1) &max-iteration
                                  , &e);
```

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

Printf ("/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)
{ printf(" / solution is converge/n");

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

max iteration = i;
                else
          if (f2 * f(0) < 0)
        {
            x1 = x2 ;
            f1 = f2;
        }
            else
```

```
{
    x0 = x2 j
    f0 = f2 j
        }
        }

    getch ();
```

Input :-

Input Two initial Roots maximum iterations and tolerance value :-

0 J 10

Output :-

Solution is converge

Roots of the given equation is = 0.618

Q-3. C- प्रोग्राम के द्वारा समीकरण $x^3-3x-5=0$ का मूल अन्तराल (2,3) में स्थित है। न्युटन रेफसन विधि द्वारा दशमलव के चार स्थानों तक शुद्ध ज्ञात 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, fo, y1, x, e;
    long int. j monitor;
        clrscr();
printf(") Input initial root max. iterations
  and tolerance value : (n");
```

```
Scanf("%Lf %Ld %f %f", &maxiter, &e);
    for (i=j; i < monitor; i++)

{
        f0 = f(x0);
        f1 = df(x0);
        x1 = x0 - (f0/f1);
    if ((fabs(x1-x0)/x1) < e)

printf("\n Solution convergers to a roots(x)
printf("\n Number of iteration = %Ld\n");
printf("\n Root of the given equation is a
                8.4 if", x1);
        i = moniter;

    else
        {
            x0 = xj;
        }

    getch();
```

Input :- Input initial root, max itriations and tolerance value;

2 10 0.00001

Teacher's Signature.....

PAGE NO. .....

Output :→

Solution Converges to a root

number of Iterations = 4

Root of the given equation is = 2.2790

Q-4 C- प्रोग्राम को लिखते हुए रंगे कुट्टा विधि (चतुर्थ कोटि) का प्रयोग कर $x = 1.5$ पर y का सन्निकटन मान ज्ञात कीजिए यदि -

$\frac{dy}{dx} = xy^2 - \frac{y}{x}$ जबकि $x = 1.42$, $y = 1$ एवं पद लम्बाई 0.1 है।

Ans यहाँ

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

$x_0 = 1$, $y_0 = 1$, तथा $h = 0.1$

रंगे कुट्टा विधि (चतुर्थ कोटि) का सूत्र :-

$$k_1 = hf(x_n y_n)$$

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

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

$$k_4 = hf(x_n + h/2, y_n + k_3/2)$$

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

$$y_{n+1} = y_n + k \qquad n = 0, 1, 2, 3, \ldots$$

रंगे कुट्टा 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 */

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 (" in/n step 'x' y/n ");
  for (i = 1) i <- h; i++)

k1 = h * f(x, y); * function call */
k2 = h * f(x+h/2:0, y+k1/2.0); / * function
                                call */
k3 = h * f(x+h/2-0, y+k2/2.0) * 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 (" %.3d %.10.3f %.10.3fh: x.y);
printf ("/n Value of yat x = %.f'h/h ; x, y)

    getch ();

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

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~~ * *

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 at X = 1.5000 is 1.3334

PRIYANSHU PUBLICATION

Teacher's Signature.....

Q=5 C- प्रोग्राम को लिखते हुये समाकलन $\int_0^6 \frac{dx}{1+x^2}$ का मान ट्रेपिलजोइडल नियम द्वारा ज्ञात कीजिए।

Ans ट्रेपिलजोइडल नियम का सूत्र :-

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

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 xo, xn, h, sum, result;
    int i, n;
    clrscr();
printf(" Enter the law and upper limits
        of Integral ");
```

```
Scanf("%f%f" &x0, &xn);
Scanf("%f" &h);
    n = int(xn - x0)/h;
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
  trapezoidel rule is %f\n result
    getch();
}
  /* function definition f() */
      float f(float x)
{
  return (1.0/(1.0 + x*y); integer.
```

PAGE NO. .....

DATE .....

Input :-

Enter the lower and upper limites

integral 0.0.6.0

enter the segment with : 1.0

Output :->

Value of integral using trapezoida rule is 1.4107799.

PAGE NO. Experiment- (6)

DATE .....

Q=6 गॉस उन्मूलन विधि का प्रयोग करके रैखिक समीकरणों की निम्नलिखित प्रणाली के लिए C- प्रोग्राम लिखें?

$$2x + 8y + 2z = 14$$
$$x + 6y - 2 = 13$$
$$2x - y + 2z = 5$$

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 angmented (coefficient of equ.)
                                    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++)
```

PRIYANSHU

Teacher's Signature.....

```
{
    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] = %7.3f\n", i, x[i]);

getch();
```

Input :-

Enter the Number of equation : 3

The Number of Equation are : 3

Now Enter augmented (coefficients of equ.) 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