/\*

NAME: KAMLESH SINGH BISHT (SECTION I)

UNIVERSITY ROLL NO.: 2016802

ROOTS OF A QUADRATIC EQUATION

\*/

#include<stdio.h>

#include<math.h>

int main()

{

int a,b,c;

printf("NAME : KAMLESH SINGH BISHT");

printf("\nSECTION I");

printf("\nBTech CSE");

printf("\nUNIVERSITY ROLL NO. : 2016802");

printf("\nENTER THE COEFFICIENTS OF THE QUADRATIC EQUATION (ax^2+bx+c=0) as a,b,c: \n");

scanf("%d",&a);

if(a==0)

{

printf("\nENTER THE COEFFICIENT OF x^2 AGAIN !!!");

exit(0);

}

scanf("%d %d",&b,&c);

float D=b\*b -4\*a\*c;

float alpha,beta;

alpha=(-b+sqrt(D))/(2\*a);

beta=(-b-sqrt(D))/(2\*a);

if(D<0)

{

float real=-b/2\*a;

float img=sqrt(-D)/2\*a;

printf("\nIMAGINARY ROOTS : %.2f+%.2fi , %.2f-%.2fi",real,img,real,img);

}

else if(D>0)

{

printf("\nREAL ROOTS : %.2f , %.2f",alpha,beta);

}

else

{

printf("\nEQUAL ROOTS : %.2f , %.2f",alpha,beta);

}

return 0;

}

/\*

NAME: KAMLESH SINGH BISHT (SECTION I)

UNIVERSITY ROLL NO.: 2016802

BISECTION METHOD

\*/

#include <stdio.h>

#define EPSILON 0.01

double func ( double x)

{

return x\*x\*x -2\*x- 5;

}

void bisection(double a, double b)

{

int count=1;

int itr;

if (func(a) \* func(b) >= 0)

{

printf("YOU HAVE ENTERED WRONG INTERVAL!!!\n");

return;

}

printf (" \n ENTER NUMBER OF ITERATIONS: ");

scanf (" %d", &itr);

printf("\nENTERED INTERVAL :(%.2lf , %.2lf)\n",a,b);

printf("\nNUMBER OF ITERATIONS : %d\n",itr);

printf("\nERROR ALLOWED : %lf\n",EPSILON);

double c = a;

while (count<=itr)

{

c = (a+b)/2;

if (func(c) == 0.0)

break;

printf (" \n The root after %d iterations is %lf. \n", count, c);

if (func(c)\*func(a) < 0)

b = c;

else

a = c;

count++;

}

if ((b-a) >= EPSILON)

{

printf("After %d iterations, root = %lf\n", itr, c);

}

}

int main ()

{

double a,b;

printf("NAME : KAMLESH SINGH BISHT");

printf("\nSECTION I");

printf("\nBTech CSE");

printf("\nUNIVERSITY ROLL NO. : 2016802");

printf ( " \n ENTER THE INTERVAL FOR THE EQUATION (x^3-2\*x -5): \n");

scanf (" %lf %lf", &a,&b);

bisection(a,b);

return 0;

}

/\*

NAME: KAMLESH SINGH BISHT (SECTION I)

UNIVERSITY ROLL NO.: 2016802

REGULA FALSI METHOD

\*/

#include<stdio.h>

#include<math.h>

#define EPSILON 0.00001

double func(double x)

{

return x\*x\*x - 2\*x -5;

}

void regula\_falsi(double a, double b)

{

int count=1;

int it;

if (func(a) \* func(b) > 0)

{

printf("You have not assumed right a and b\n");

return;

}

printf (" \n ENTER NUMBER OF ITERATIONS YOU WANT TO PERFORM : ");

scanf (" %d", &it);

printf("\nENTERED INTERVAL:(%.2lf , %.2lf)\n",a,b);

printf("\nNUMBER OF ITERATIONS : %d\n",it);

printf("\nERROR ALLOWED : %lf\n",EPSILON);

if(func(a)\*func(b)==0)

{

printf("ROOTS OF THE EQUATION ARE : %lf , %lf",a,b);

}

else

{

double c,prev;

while(count<=iterate)

{

prev=c;

c = (a\*func(b) - b\*func(a))/ (func(b) - func(a));

if (func(c)==0)

break;

printf (" \n The root after %d iterations is %lf. \n", count, c);

if (func(c)\*func(a) < 0)

b = c;

else

a = c;

count++;

}

if (fabs(func(c)-func(prev))>EPSILON)

{

printf("\nThe value of root is :%lf ",c);

}

}

}

int main ()

{

double a,b;

printf("NAME : KAMLESH SINGH BISHT");

printf("\nSECTION I");

printf("\nBTech CSE");

printf("\nUNIVERSITY ROLL NO. : 2016802");

printf ( " \n ENTER THE INTERVAL FOR THE EQUATION (x^3-2x-5): \n");

scanf (" %lf %lf", &a,&b);

regula\_falsi(a,b);

return 0;

}

/\*

NAME: KAMLESH SINGH BISHT (SECTION I)

UNIVERSITY ROLL NO.: 2016802

NEWTON RAPHSON METHOD

\*/

#include<stdio.h>

#include<math.h>

#define EPSILON 0.001

double func(double x)

{

return 3\*x\*x\*x - 2\*x\*x -1;

}

double derv(double x)

{

return 9\*x\*x - 4\*x;

}

void newton\_raphson(double x,int it)

{

int count=1;

double h=func(x)/derv(x);

while(count<=it)

{

h=func(x)/derv(x);

x=x-h;

printf (" \n%d iterations: %lf \n", count, x);

count++;

}

if(fabs(h)>=EPSILON)

{

printf("After %d iterations, root = %lf\n", it, x);

}

}

int main()

{

double a,b,x;

int it;

printf("NAME : KAMLESH SINGH BISHT");

printf("\nSECTION I");

printf("\nBTech CSE");

printf("\nUNIVERSITY ROLL NO. : 2016802");

printf ( " \n ENTER THE INTERVAL FOR THE EQUATION (x^3 -2^x -1): \n");

scanf (" %lf %lf", &a,&b);

if(func(a)\*func(b)<0)

{

if(abs(func(a)-0)>abs(func(b)-0))

{

x=a;

}

x=b;

}

else

{

printf("YOU HAVE ENTERED WRONG INTERVAL\n");

return 0;

}

printf (" \n INPUT THE NUMBER OF ITERATIONS YOU WANT TO PERFORM : ");

scanf (" %d", &it);

printf("\nTHE ENTERED INTERVAL IS :(%.2lf , %.2lf)\n",a,b);

printf("\n NUMBER OF ITERATIONS : %d\n",it);

printf("\nERROR ALLOWED : %lf\n",EPSILON);

newton\_raphson(x,it);

return 0;

}

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE COEFFICIENTS OF THE QUADRATIC EQUATION (ax^2+bx+c=0) as a,b,c:

2

3

4

IMAGINARY ROOTS : -2.00+4.80i , -2.00-4.80i

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE COEFFICIENTS OF THE QUADRATIC EQUATION (ax^2+bx+c=0) as a,b,c:

1

2

-3

REAL ROOTS : 1.00 , -3.00

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE COEFFICIENTS OF THE QUADRATIC EQUATION (ax^2+bx+c=0) as a,b,c:

0

ENTER THE COEFFICIENT OF x^2 AGAIN !!!

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE INTERVAL FOR THE EQUATION (x^3-2x-5):

2

1

ENTER NUMBER OF ITERATIONS: 10

ENTERED INTERVAL :(2.00 , 1.00)

NUMBER OF ITERATIONS : 9

ERROR ALLOWED : 0.010000

The root after 1 iterations is 1.500000.

The root after 2 iterations is 1.250000.

The root after 3 iterations is 1.375000.

The root after 4 iterations is 1.312500.

The root after 5 iterations is 1.343750.

The root after 6 iterations is 1.328125.

The root after 7 iterations is 1.335938.

The root after 8 iterations is 1.332031.

The root after 9 iterations is 1.330078.

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE INTERVAL FOR THE EQUATION (x^3-2x-5):

3

6

YOU HAVE ENTERED WRONG INTERVAL!!!

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE INTERVAL FOR THE EQUATION (x^3-2x-5):

2

3

ENTER NUMBER OF ITERATIONS YOU WANT TO PERFORM : 10

ENTERED INTERVAL:(2.00 , 3.00)

NUMBER OF ITERATIONS : 10

ERROR ALLOWED : 0.000010

The root after 1 iterations is 2.058824.

The root after 2 iterations is 2.081264.

The root after 3 iterations is 2.089639.

The root after 4 iterations is 2.092740.

The root after 5 iterations is 2.093884.

The root after 6 iterations is 2.094305.

The root after 7 iterations is 2.094461.

The root after 8 iterations is 2.094518.

The root after 9 iterations is 2.094539.

The root after 10 iterations is 2.094547.

The value of root is :2.094547

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE INTERVAL FOR THE EQUATION (x^3-2x-5):

1

1.5

You have not assumed right a and b

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE INTERVAL FOR THE EQUATION (x^3 -2^x -1):

2

5

YOU HAVE ENTERED WRONG INTERVAL

**Output:**

NAME : KAMLESH SINGH BISHT

SECTION I

BTech CSE

UNIVERSITY ROLL NO. : 2016802

ENTER THE INTERVAL FOR THE EQUATION (x^3 -2^x -1):

-1

2

INPUT THE NUMBER OF ITERATIONS YOU WANT TO PERFORM : 10

THE ENTERED INTERVAL IS :(-1.00 , 2.00)

NUMBER OF ITERATIONS : 10

ERROR ALLOWED : 0.001000

1 iterations: 1.464286

2 iterations: 1.156950

3 iterations: 1.026369

4 iterations: 1.000926

5 iterations: 1.000001

6 iterations: 1.000000

7 iterations: 1.000000

8 iterations: 1.000000

9 iterations: 1.000000

10 iterations: 1.000000

/\*

NAME: KAMLESH SINGH BISHT (SECTION I)

UNIVERSITY ROLL NO.: 2016802

GAUSS ELIMINATION METHOD

\*/

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int main()

{

int i, j, k, n;

float a[20][20], x[20];

double s, p;

printf("Name - Kamlesh Singh Bisht\n");

printf("Sec-I\n");

printf("University roll no- 2016802\n");

printf("Class roll no-39\n");

printf("Program-B.Tech(CSE)\n\n");

printf("Enter the number of equations :");

scanf("%d", &n);

printf("\nEnter the co-efficients of the equations :\n\n");

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

printf("a[%d][%d] = ", i, j);

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

}

printf("b[%d] = ", i + 1);

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

}

printf("\n\n");

printf("Equation 1: (%.0f)x+(%.0f)y+(%.0f)z=%.0f\n", a[0][0], a[0][1], a[0][2], a[0][3]);

printf("Equation 2: (%.0f)x+(%.0f)y+(%.0f)z=%.0f\n", a[1][0], a[1][1], a[1][2], a[1][3]);

printf("Equation 3: (%.0f)x+(%.0f)y+(%.0f)z=%.0f\n", a[2][0], a[2][1], a[2][2], a[2][3]);

printf("\n");

printf("Augmented Matrix: \n");

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

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

}

printf("%f \n",a[i][n]);

}

for (k = 0; k <= n - 1; k++)

{

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

{

p = a[i][k] / a[k][k];

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

{

a[i][j] = a[i][j] - (p \* a[k][j]);

}

}

}

printf("\n");

printf("Final Matrix is:-\n");

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

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

}

printf("%f \n",a[i][n]);

}

x[n - 1] = a[n - 1][n] / a[n - 1][n - 1];

for (i = n - 2; i >= 0; i--)

{

s = 0;

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

{

s += (a[i][j] \* x[j]);

x[i] = (a[i][n] - s) / a[i][i];

}

}

printf("\n");

printf("\nThe result is :\n");

for (i = 0; i < n ; i++)

{

printf("\nx[%d] = %.2f", i + 1, x[i]);

}

return 0;

}

**Output:**

Name - Kamlesh Singh Bisht

Sec-I

University roll no- 2016802

Class roll no-39

Program-B.Tech(CSE)

Enter the number of equations :3

Enter the co-efficients of the equations :

a[0][0] = 1

a[0][1] = 1

a[0][2] = 19

b[1] = 9

a[1][0] = 2

a[1][1] = 3

a[1][2] = 4

b[2] = 13

a[2][0] = 3

a[2][1] = 4

a[2][2] = 5

b[3] = 40

Equation 1: (1)x+(1)y+(19)z=9

Equation 2: (2)x+(-3)y+(4)z=13

Equation 3: (3)x+(4)y+(5)z=40

Augmented Matrix:

1.000000 1.000000 19.000000 9.000000

2.000000 -3.000000 4.000000 13.000000

3.000000 4.000000 5.000000 40.000000

Final Matrix is:-

1.000000 1.000000 19.000000 9.000000

0.000000 -5.000000 -34.000000 -5.000000

0.000000 0.000000 -58.799999 12.000000

The result is :

x[1] = 10.49

x[2] = 2.39

x[3] = -0.20

/\*

NAME: KAMLESH SINGH BISHT (SECTION I)

UNIVERSITY ROLL NO.: 2016802

GAUSS JORDAN METHOD

\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

float a[3][4], t;

int i , j, k;

printf("Name - Kamlesh Singh Bisht\n");

printf("Sec-I\n");

printf("University roll no- 2016802\n");

printf("Class roll no-39\n");

printf("Program-B.Tech(CSE)\n\n");

printf("Enter the number of equations :");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 4; j++)

{

printf("a[%d][%d] : ", i, j);

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

}

}

printf("Equation 1: (%.0f)x+(%.0f)y+(%.0f)z=%.0f\n", a[0][0], a[0][1], a[0][2], a[0][3]);

printf("Equation 2: (%.0f)x+(%.0f)y+(%.0f)z=%.0f\n", a[1][0], a[1][1], a[1][2], a[1][3]);

printf("Equation 3: (%.0f)x+(%.0f)y+(%.0f)z=%.0f\n", a[2][0], a[2][1], a[2][2], a[2][3]);

printf("\n\n");

printf("Augmented Matrix: \n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 4; j++)

{

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

}

printf(" \n");

}

printf("\n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

if (i != j)

{

t = a[j][i] / a[i][i];

for (k = 0; k < 4; k++)

a[j][k] = a[j][k] - (a[i][k] \* t);

}

}

}

printf("Final Matrix form : \n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 4; j++)

{

printf("\t %.0f", a[i][j]);

}

printf("\n");

}

printf("\n\nSolution is :\n ");

for (i = 0; i < 3; i++)

{

printf("%f ", a[i][3] / a[i][i]);

}

}

**Output:**

Name - Kamlesh Singh Bisht

Sec-I

University roll no- 2016802

Class roll no-39

Program-B.Tech(CSE)

Enter the number of equations :a[0][0] : 2

a[0][1] : 1

a[0][2] : 1

a[0][3] : 5

a[1][0] : 3

a[1][2] : 2

a[1][3] : 15

a[2][0] : 2

a[2][1] : 1

a[2][2] : 4

a[2][3] : 8

Equation 1: (2)x+(1)y+(1)z=5

Equation 2: (3)x+(5)y+(2)z=15

Equation 3: (2)x+(1)y+(4)z=8

Augmented Matrix:

2.000000 1.000000 1.000000 5.000000

3.000000 5.000000 2.000000 15.000000

2.000000 1.000000 4.000000 8.000000

Final Matrix form :

2 0 0 2

0 4 0 7

0 0 3 3

Solution is:

1.000000

2.000000

1.000000