LAB 1
BISECTION METHOD

OBJECTIVE

To find the real root of $x e^{x} - 1$

TOOLS

gcc (code::blocks)

ALGORITHM/FLOWCHART:

1. Read x1, x2, e
2. e is the absolute error
3. Compute: f1 = f(x1) and f2 = f(x2)
4. If (f1*f2) > 0, then display initial guesses are wrong and goto (11).
5. x = (x1 + x2)/2
6. If (| (x1 – x2)/x | < e), then display x and goto (11).
7. Else, f = f(x)
8. If ((f*f1) > 0), then x1 = x and f1 = f.
9. Else, x2 = x and f2 = f. Bishal Sharma
10. Goto (5).
11. Stop

CODE:

```
#include<stdio.h>
#include<math.h>
float f(float x){
            return(x*exp(x)-1);
}
 int main()
{
            float x1,x2,xm=0,flg,eps=0.0001,a,b,c;
            int count=0;
            do{
                  printf("Enter Initial guess");
                  scanf("%f %f", &x1,&x2);
                  if(f(x1)*f(x2)>0)
                  printf("\n Invalid Guess.... Type the guess again...");
            }
            while(f(x1)*f(x2)>0);
            printf("\n ITR    x1        x2         xm           F(x1)       F(x2)      F(xm)");
            do{
            flg=xm;
            xm=(x1+x2)/2.00;
            count++;
            a=f(x1);
            b=f(x2);
            c=f(xm);
            printf("\n %d      %f    %f    %f     %f     %f    %f",count,x1,x2,xm,a,b,c);
            if(f(xm)*f(x1)<0)
                  x2=xm;
```

```
        else
                x1=xm;
        }
        while(fabs(xm-flg)>eps);
        printf("\n Result = %12.4f   count=%d",xm,count);
        return 0;
}
```

OBSERATION AND RESULT

```
Enter Initial guess0
1

 ITR    x1          x2          xm           F(x1)        F(x2)       F(xm)
 1      0.000000    1.000000    0.500000    -1.000000    1.718282    -0.175639
 2      0.500000    1.000000    0.750000    -0.175639    1.718282    0.587750
 3      0.500000    0.750000    0.625000    -0.175639    0.587750    0.167654
 4      0.500000    0.625000    0.562500    -0.175639    0.167654    -0.012782
 5      0.562500    0.625000    0.593750    -0.012782    0.167654    0.075142
 6      0.562500    0.593750    0.578125    -0.012782    0.075142    0.030619
 7      0.562500    0.578125    0.570312    -0.012782    0.030619    0.008780
 8      0.562500    0.570312    0.566406    -0.012782    0.008780    -0.002035
 9      0.566406    0.570312    0.568359    -0.002035    0.008780    0.003364
 10      0.566406    0.568359    0.567383    -0.002035    0.003364    0.000662
 11      0.566406    0.567383    0.566895    -0.002035    0.000662    -0.00068
7
 12      0.566895    0.567383    0.567139    -0.000687    0.000662    -0.00001
3
 13      0.567139    0.567383    0.567261    -0.000013    0.000662    0.000325
 14      0.567139    0.567261    0.567200    -0.000013    0.000325    0.000156
 Result =       0.5672   count=14
Process returned 0 (0x0)   execution time : 6.920 s
Press ENTER to continue.
```

CONCLUSION

hence we can use bisection method to find the real root of $x e^{x} - 1$

LAB 2
FALSE POSITION METHOD

OBJECTIVE

To find the real root of $x - cosx$

TOOLS

gcc (code::blocks)

ALGORITHM/FLOWCHART:

1 Start
2 Read values of x0, x1 and e
3 Computer function values f(x0) and f(x1)
4 Check whether the product of f(x0) and f(x1) is negative or not.\
    If it is positive take another initial guesses.
    If it is negative then goto step 5.
5 Determine: x = [x0*f(x1) – x1*f(x0)] / (f(x1) – f(x0))
6 Check whether the product of f(x1) and f(x) is negative or not.
    If it is negative, then assign x0 = x;
    If it is positive, assign x1 = x;
7 Check whether the value of f(x) is greater than 0.00001 or not.
    If yes, goto step 5.
    If no, goto step 8.
    *Here the value 0.00001 is the desired degree of accuracy, and hence the
10 stopping criteria.*
8 Display the root as x.
9 Stop

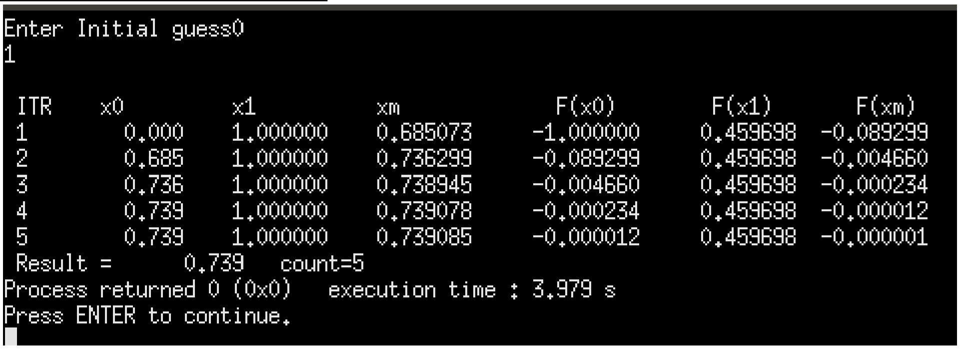
CODE:

```
#include<stdio.h>
#include<math.h>
float f(float x) {
            return(x-cos(x));
}
int main()
{
            float x0,x1,xm=0,flg,eps=0.0001,a,b,c;
            int count=0;
             do{
                  printf("Enter Initial guess");
                  scanf("%f %f", &x0,&x1);
                  if(f(x0)*f(x1)>0)
                  printf("\n Invalid Guess.... Type the guess again...");
             }
             while(f(x0)*f(x1)>0);
             printf("\n ITR    x0        x1         xm            F(x0)       F(x1)       F(xm)");
             do{
                   count++;
                   flg=xm;
                   xm=x0-(f(x0)*(x1-x0))/(f(x1)-f(x0));
```

```
            a=f(x0);
            b=f(x1);
            c=f(xm);
            printf("\n %d %12.3f    %f    %f     %f     %f  %f",count,x0,x1,xm,a,b,c);
           if(f(xm)*f(x0)<0)
                        x1=xm;
          else
                         x0=xm;

      }
      while(fabs(flg-xm)>eps);
      printf("\n Result = %10.3f   count=%d",xm,count);
      return 0;
}
```

OBSERATION AND RESULT



```
Enter Initial guess0
1

 ITR    x0         x1          xm            F(x0)        F(x1)      F(xm)
 1       0.000    1.000000    0.685073     -1.000000     0.459698  -0.089299
 2       0.685    1.000000    0.736299     -0.089299     0.459698  -0.004660
 3       0.736    1.000000    0.738945     -0.004660     0.459698  -0.000234
 4       0.739    1.000000    0.739078     -0.000234     0.459698  -0.000012
 5       0.739    1.000000    0.739085     -0.000012     0.459698  -0.000001
 Result =      0.739   count=5
Process returned 0 (0x0)   execution time : 3.979 s
Press ENTER to continue.
```

CONCLUSION

hence we can use false position method to find the real root of $x^3+x^2-3x-3=0$

LAB 3
FIXED POINT METHOD

OBJECTIVE

To find the real root of $\frac{(x^2+2)}{3}$

TOOLS

gcc (code::blocks)

ALGORITHM/FLOWCHART:

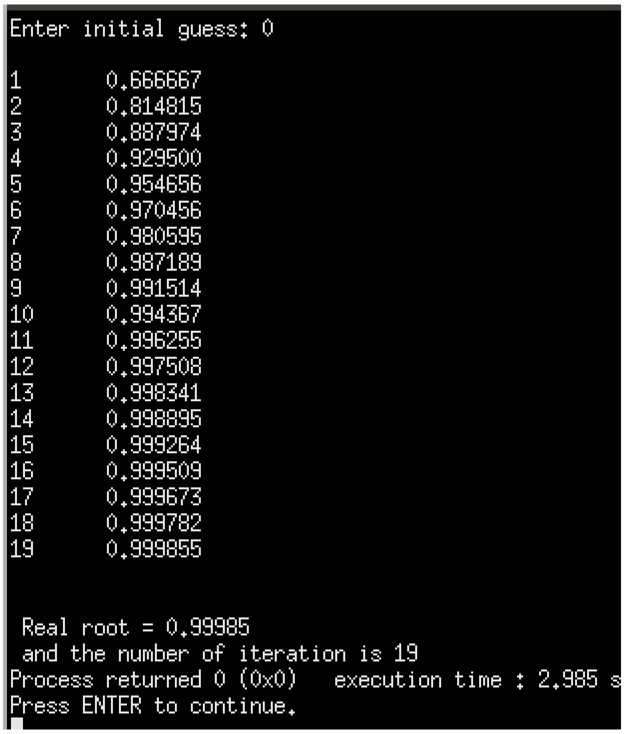
1 Start
2 Read values of x0 and e.
Here x0 is the initial approximation e is the absolute error or the desired degree of accuracy, also the stopping criteria
3 Calculate x1 = g(x0)
4 If |x1 – x0| <= e, goto step 6.
5 Else, assign x0 = x1 and goto step 3.
6 Display x1 as the root.
7 Stop

CODE:

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

float f(float x){
	return ((pow(x,2)+2)/3);
}
int main()
{
	float x0,x1=0, flg, eps=0.0001;
	int count=0;
	 printf("Enter initial guess: ");
	 scanf("%f",&x0);
	do {
		 flg=x1;
		 x1=f(x0);
		count++;
		printf("\n%d\t%f",count,x1);
		x0=x1;
	}
	while(fabs(flg-x1)>eps);
	printf("\n\n\n Real root = %0.5f \n and the number of iteration is %d",x1,count);
  return 0;
  }
```

OBSERATION AND RESULT



```
Enter initial guess: 0

1       0.666667
2       0.814815
3       0.887974
4       0.929500
5       0.954656
6       0.970456
7       0.980595
8       0.987189
9       0.991514
10      0.994367
11      0.996255
12      0.997508
13      0.998341
14      0.998895
15      0.999264
16      0.999509
17      0.999673
18      0.999782
19      0.999855


 Real root = 0.99985
 and the number of iteration is 19
Process returned 0 (0x0)   execution time : 2.985 s
Press ENTER to continue.
```

CONCLUSION

hence we can use fixed point method to find the real root of $\frac{(x^2+2)}{3}$

LAB 4
NEWTON-RAPHSON METHOD

OBJECTIVE

To find the real root of $3^x + sinx - e^x$

TOOLS

gcc (code::blocks)

ALGORITHM/FLOWCHART:

1 Start
2 Read x, e, n, d
x is the initial guess e is the absolute error i.e the desired degree of accuracy n is for operating loop d is for checking slope
3 Do for i =1 to n in step of 2
4 f = f(x)
5 f1 = f'(x)
6 If (|f1| < d), then display too small slope and goto 11.
7 x1 = x – f/f1
8 If (|(x1 – x)/x1| < e), the display the root as x1 and goto 11.
9 x = x1 and end loop
10 Display method does not converge due to oscillation.
11 Stop

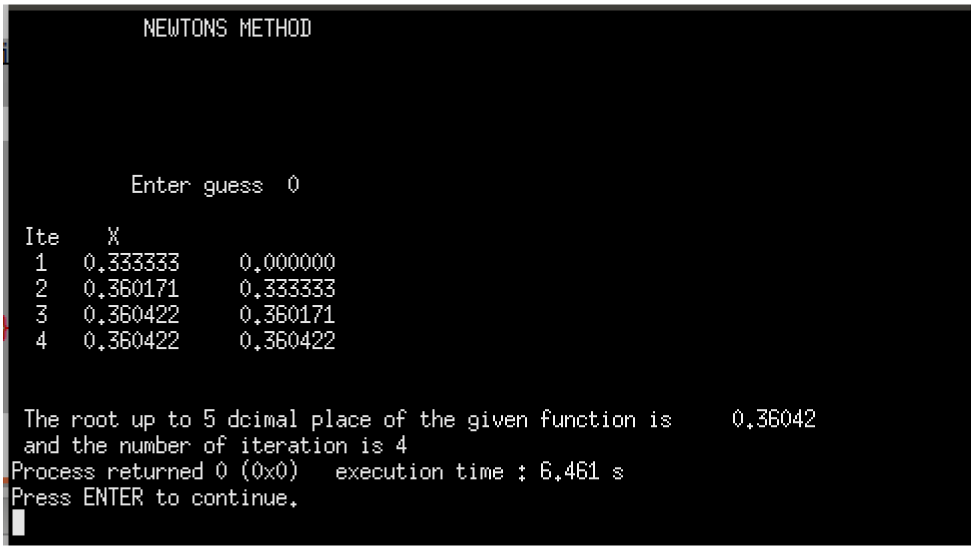
CODE:

```
#include<stdio.h>
#include<math.h>
float f(float x){
                return(3*x+sin(x)-exp(x));
}
float g(float y){
                return(3+cos(y)-exp(y));
}
int main()
{

        float x0,x1=0,eps=0.0000001,flg;
        count=0;
        printf("         NEWTONS METHOD");
        printf("\n\n\n\n\n\n         Enter guess  ");
        scanf("%f",&x0);
        printf("\n Ite    X");
        do{
                flg=x1;
                x1=x0-(f(x0)/g(x0));
                count++;
                printf("\n  %d   %f     %f",count,x1,x0);
                x0=x1;
        }
        while(fabs(flg-x1)>eps);
         printf("\n\n\n The root up to 5 decimal place of the given function is%12.5f \n and the
```

```
        number of iteration is %d",x1,count);
        return 0;
}
```

OBSERATION AND RESULT



CONCLUSION

hence we can use newton-raphson method to find the real root of $3^x + sinx - e^x$

LAB 5
SECANT METHOD

OBJECTIVE

To find the real root of x^3+x^2+x+7

TOOLS

gcc (code::blocks)

ALGORITHM/FLOWCHART:

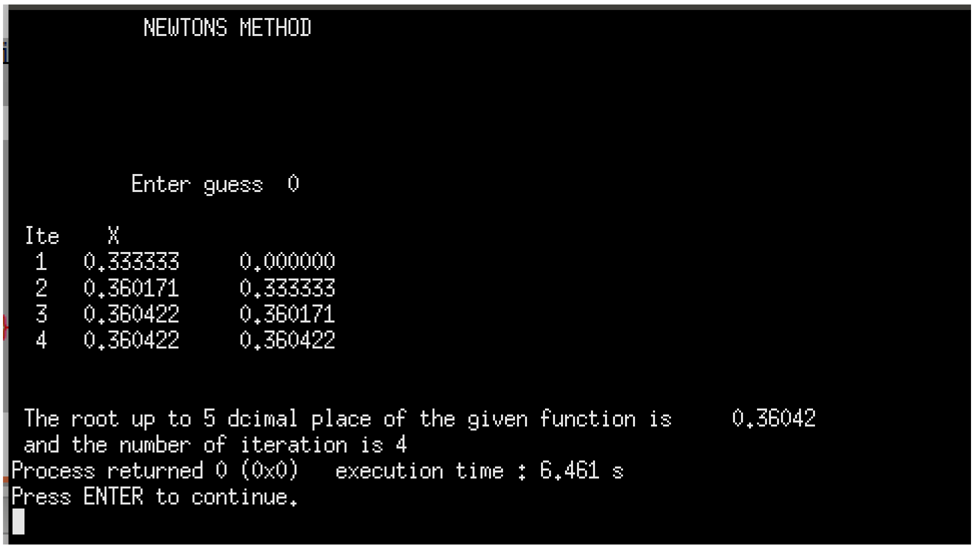
1 Start
2 Get values of x0, x1 and e
Here x0 and x1 are the two initial guesses e is the stopping criteria, absolute error or the desired degree of accuracy
3 Compute f(x0) and f(x1)
4 Compute x2 = |x0*f(x1) – x1*f(x0)|/ |f(x1) – f(x0)|
5 Test for accuracy of x2
If |(x2 – x1)/x2 |> e, assign x0 = x1 and x1 = x2 goto step 4 Else,goto step 6
6 Display the required root as x2.
7 Stop

CODE:

```
#include<stdio.h>
#include<math.h>
float f(float x){
        return(3*x+sin(x)-exp(x));
}
float g(float y){
        return(3+cos(y)-exp(y));
}
int main()
{

    float x0,x1=0,eps=0.0000001,flg;
    count=0;
    printf("        NEWTONS METHOD");
    printf("\n\n\n\n\n\n\n        Enter guess  ");
    scanf("%f",&x0);
    printf("\n Ite    X");
    do{
        flg=x1;
        x1=x0-(f(x0)/g(x0));
        count++;
        printf("\n  %d   %f     %f",count,x1,x0);
        x0=x1;
    }
    while(fabs(flg-x1)>eps);
     printf("\n\n\n The root up to 5 decimal place of the given function is%12.5f \n and the number of iteration is %d",x1,count);
    return 0;
}
```

OBSERATION AND RESULT



```
          NEWTONS METHOD

         Enter guess  0

 Ite    X
  1   0.333333     0.000000
  2   0.360171     0.333333
  3   0.360422     0.360171
  4   0.360422     0.360422

 The root up to 5 dcimal place of the given function is     0.36042
 and the number of iteration is 4
Process returned 0 (0x0)   execution time : 6.461 s
Press ENTER to continue.
```

CONCLUSION

hence we can use newton-raphson method to find the real root of $3^x + sinx - e^x$