LAB V

A. Write a C program that finds the roots of a quadratic polynomial $ax^2 + bx + c = 0$. The program accepts values of a, b and c from the terminal and then proceeds depending on the values of a, b and c.

1. If a is zero, then the program prints out the message "**It is not a quadratic equation: Enter nonzero a**". Once you enter a nonzero value of a, it proceeds to next step.

2. It next find the discriminant $d = b^2 - 4ac$.

3. If $d > 0$, the program finds the real distinct roots

$$r1 = \frac{-b + \sqrt{d}}{2a}, \quad r2 = \frac{-b - \sqrt{d}}{2a}.$$

   It prints out the roots with appropriate message.

4. If $d = 0$, the roots are real and equal

$$r1 = \frac{-b}{2a}, \quad r2 = \frac{-b}{2a}.$$

   It prints out the roots with appropriate message.

```
/* This prog. finds roots of ax^2+bx+c=0*/

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

int main()
{
  double a,b,c,d,r1,r2;

  printf("Enter a,b,c: ");
  scanf("%lf%lf%lf",&a,&b,&c);

  while(a==0)
  {
   printf("Enter a nonzero a: ");
   scanf("%lf",&a);
  }

  d=b*b-4.0*a*c;
  if(d>0)
  {
   r1 = (-b+sqrt(d))/(2*a);
   r2 = (-b-sqrt(d))/(2*a);
   printf("The roots are real and distinct: r1=%0.2lf  r2=%0.2lf\n",r1,r2);
  }
   else if(d==0.0)
   {
   r1 = -b/(2*a);
```

```
    r2 = r1;
    printf("The roots are real and equal: r1=%0.2lf  r2=%0.2lf\n",r1,r2);
   }
   else
   {
    r1 = -b/(2*a);
    r2 = sqrt(-d)/(2*a);
    printf("The roots are complex conjugate:\n");
    printf("r1=%0.2lf + i %0.2lf   r2=%0.2lf - i %0.2lf\n",r1,r2,r1,r2);
   }


    return 0;
}
```

5. If $d < 0$, the roots are complex conjugate with real part $r1 = -b/2a$ and imaginary part $r2 = \sqrt{-d}/2a$. It then prints out the roots with appropriate message.

B. The equation $f(x) := (1-x)\cos x - \sin x = 0$ has a root between $a = 0$ and $b = 1$ since $f(a)f(b) < 0$. The bisection method of finding the root proceeds as follows:

1. It finds the midpoint $r = (a+b)/2$.
2. If $f(r) = 0$, then r is the root. If $|b-a|$ is very small, then also we can take r as the root. In either of the cases, our job is done.
3. If $f(r) \neq 0$ and $f(a)f(r) < 0$, then the root lies between a and r. We assign r to b and go to step 1.
4. If $f(r) \neq 0$ and $f(b)f(r) < 0$, then the root lies between r and b. We assign r to a and go to step 1.
5. If the number of iterations is high, we may stop the process with appropriate message.

```
/* This prog. finds roots of ax^2+bx+c=0*/
/* Can also be done with
 * #deine f(x) (1-(x))*cos((x))-sin((x))
 * fa=f(a) fb=f(b) fr=f(r) etc*/

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

int main()
{
  int iter,ITMAX=1000;
  double a=0,b=1,r,eps=1.e-6;
  double fa,fb,fr;


  fa = (1.0-a)*cos(a)-sin(a);
  fb = (1.0-b)*cos(b)-sin(b);

  r=0.5*(a+b);
  fr = (1.0-r)*cos(r)-sin(r);
```

```
  iter = 0;

  while(fr != 0.0 && fabs(b-a)>eps && iter < ITMAX)
  {
   if(fr*fa < 0.0)
   {
    b=r;
    fb=fr;
   }
   else
   {
    a=r;
    fa=fr;
   }
   iter++;
   r=0.5*(a+b);
   fr = (1.0-r)*cos(r)-sin(r);

  }

   if(iter==ITMAX)
   {
    printf("The bisection method does not converge\n");
    return 0;
   }

   printf("The required root is %0.5lf\n",r);


   return 0;
}
```