LAB V

1.

```
/* This prog. counts the number of integers between 1 & 9 entered*/
#include <stdio.h>

int main()
{
int a,nev=0,nmul3=0;

printf("Enter integers between 1 & 9 both inclusive, outside range to stop\n");
printf("Enter integer :");
scanf("%d",&a);
while(a>=1 && a<=9)
{
  switch(a%2==0)
  {

  case 1: nev++;
  }

  switch(a%3==0)
  {

  case 1: nmul3++;
  }
printf("Enter integer :");
scanf("%d",&a);

}

printf("Total no of even integer entered is %d\n",nev);
printf("Total no of multiples of 3 entered is %d\n",nmul3);

return 0;
}
```

2.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
int cbracket(double,double);
double func(double);
double rootb(double,double,double,int);
int main()
{
double x,eps,a,b;
int flag,Nmax;

printf("Enter eps and Nmax :");
scanf("%lf%d",&eps,&Nmax);
printf("Enter a, b :");
scanf("%lf%lf",&a,&b);
```

```
flag=cbracket(a,b);
if(flag==0)
   {
    printf("Root must be bracketed\n");
    return 0;
   }

x=rootb(a,b,eps,Nmax);

printf("Root = %g\n",x);


return 0;
}

int cbracket(double a,double b)
{
if(func(a)*func(b)<0)
      {
       return 1;
      }
else
      {
      return 0;
      }
}


double func(double x)
{
return (1-x)*cos(x)-sin(x);
}

double rootb(double a,double b,double eps,int N)
{
double r;
int i=0;

r=0.5*(a+b);
if(func(r)==0.0)
  {
    return r;
  }
while(func(r)!=0.0 && fabs(b-a)>eps && i<N)
  {
    if(func(r)*func(a)<0)
         {
          b=r;
          }
     else
```

```
         {
          a=r;
         }
    i++;
    r = 0.5*(a+b);
  }

  if(i==N)
      {
       printf("Increase the max iteration\n");
       exit(1);
      }

return r;
}
```

3. /* This prog. finds gcd of two ints*/

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

int nrgcd(int,int);  //uses non recursive approach

int main()
{
  int a,b;

  printf("Enter two integers: ");
  scanf("%d%d",&a,&b);
  if(a==0 && b==0)
      printf("At least one number must be nonzero\n");
  else
      printf("GCD of %d and %d is %d\n",a,b,nrgcd(abs(a),abs(b)));

return 0;
}

int nrgcd(int a, int b)
{
int r;

if(b==0)
{
return a;
}
r=a%b;
while(r !=0)
{
        a=b;
        b=r;
        r=a%b;
}
return b;
}
```

4.

```
/* This prog. finds gcd of two ints*/

#include <stdio.h>
#include <stdlib.h>

int rgcd(int,int);  //uses recursive approach

int main()
{
  int a,b,r,p,q;

  printf("Enter two integers: ");
  scanf("%d%d",&a,&b);

  if(a==0 && b==0)
      printf("At least one number must be nonzero\n");
  else
      printf("GCD of %d and %d is %d\n",a,b,rgcd(abs(a),abs(b)));

return 0;
}

int rgcd(int a, int b)
{

if(b==0)
{
    return a;
}

return rgcd(b,a%b);

}
```

5.

```
/* This prog. finds lcm of two ints using |a*b|/gcd(a,b)*/

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

```
int gcd(int,int);
int lcm(int,int);

int main()
{
  int a,b,r;

  printf("Enter two integers: ");
  scanf("%d%d",&a,&b);

  r= lcm(a,b);

if(r==-1){
  printf("Both the number must be nonzero\n");
  }
  else
  {
  printf("LCM of %d and %d is %d\n",a,b,r);
  }


return 0;
}

int lcm(int a, int b)
{
int r;
if(a==0 || b==0)
   {
     return -1;
   }
else
   {
    return abs(a*b)/gcd(abs(a),abs(b));
   }
}

int gcd(int a, int b)
{
int r;

r=a%b;
while(r !=0)
{
        a=b;
        b=r;
        r=a%b;
}
return b;
}
```

6. /* This prog. finds sum of digits of a number*/

```
#include <stdio.h>

int rsumd(int);  //uses recursive approach

int main()
{
  int a,s;

  printf("Enter a non-negative integer: ");
  scanf("%d",&a);
  while(a<0)
  {
    printf("Number must be non-negative: enter again:");
    scanf("%d",&a);
  }

  s= rsumd(a);
  printf("Sum of digits of %d is %d\n",a,s);

return 0;
}

int rsumd(int a)
{
if(a==0)
  return 0;
else
  return a%10+rsumd(a/10);
}
```