

CS 2 lab 3

Firstly I created the function which calculates the root of given function with given tolerance. I had some issues with implementing number of iterations as pointer, so I implemented it as global variable.

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

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

```
int i_iter;
```

```
double bisec( double xa, double xb, double (*pf)(double), double eps)
```

```
{
```

```
    int    i;
```

```
    double fa, fb, xc, fc;
```

```
    fa = pf(xa);
```

```
    fb = pf(xb);
```

```
    if ( fa * fb > 0.0)
```

```
    {
```

```
        i_iter = -1;
```

```
        return 0;
```

```
    }
```

```

for ( i = 1; i <= 1000; i++ )
{
    xc = ( xa + xb ) / 2.;
    fc = pf( xc );

    if( fa * fc < 0. )
    {
        xb = xc;
        fb = fc;
    }
    else
    {
        xa = xc;
        fa = fc;
    }

    if ( fabs(fc) < eps && fabs(xb-xa) < eps)
        break;
}
i_iter = i;
return xc;
}

```

```
double gx(double x){  
    return(cos(x)-x);  
}
```

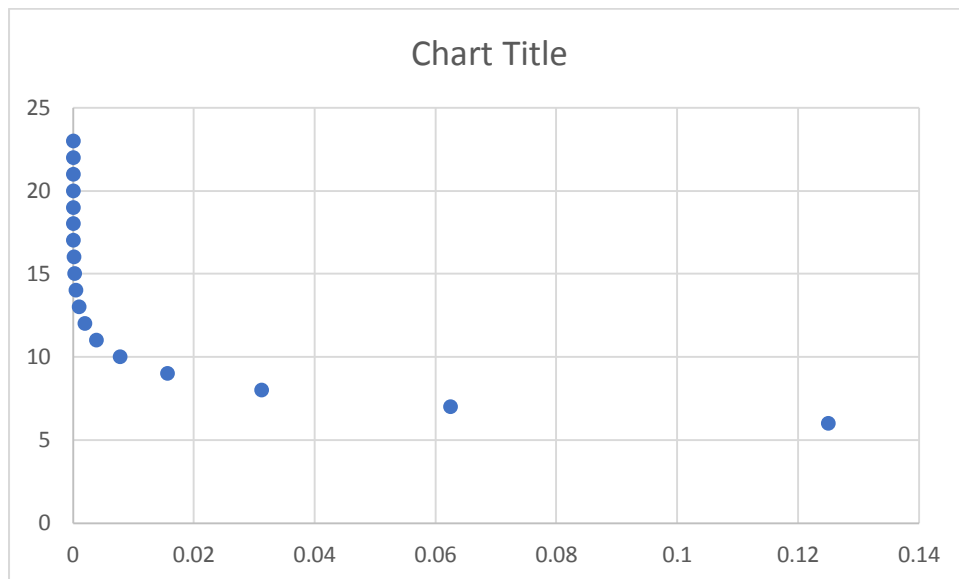
Question 1

I created the file to which I wrote the tolerable error and corresponding number of iterations.

```
FILE *fp=fopen("bisec_data.txt","w");  
    printf("Insert first boundary point: ");  
    scanf("%lf", &a);  
    printf("\nInsert second boundary point: ");  
    scanf("%lf", &b);  
    printf("\nInsert tolerable error: ");  
    scanf("%lf", &eps);  
    double solution = bisec(a, b, gx, eps);  
    printf("\nSolution is: %lf",solution);  
    printf("\nNumber of iterations: %d", i_iter);  
    int power = -3;  
    for(int i = 0; i < 18; i++){  
  
        eps = pow(2,power);  
        solution = bisec(a, b, gx, eps);  
        fprintf(fp,"2^%d\t%d\n", power, i_iter);  
        power = power + 1;  
    }
```

```
fclose(fp);
```

Question 2



Question 3

```
double secant( double xa, double xb, double (*pf)(double), double eps)
```

```
{  
  
    int    i;  
    double fa, fb, xc, fc;  
  
    fa = pf(xa);  
    fb = pf(xb);  
  
    if ( fa * fb > 0.0)  
    {  
        i_iter = -1;  
        return 0;  
    }  
  
    for ( i = 1; i <= 100000; i++ )  
    {  
        xc = xb-fb*(xb-xa)/(fb-fa);
```

```

        fc = pf( xc );

        if( fa * fc < 0. )
        {
            xb = xc;
            fb = fc;
        }
        else
        {
            xa = xc;
            fa = fc;
        }

        if ( fabs(fc) < eps)
            break;
    }
    i_iter = i;
    return xc;
}

double newton( double xa, double (*pf)(double), double(*pff)(double), double eps)
{
    int    i;
    double fa, fpa, xc, fc;

    fa = pf(xa);
    fpa = pff(xa);

    for ( i = 1; i <= 100000; i++ )

```

```

{
    xc = xa - fa/fpa;
    fc = pf( xc );

    xa = xc;

    if ( fabs(fc) < eps)
        break;
}
i_iter = i;
return xc;
}

```

Question 4

I created global variable w and modified g function. I created new file in which I saved data.

```

for(int i = 0; i < 146; i++){
    solution = bisec(a, b, gx, eps);
    fprintf(f,"%lf\t%lf\n", w, solution);
    w += 0.1;
}

```

Question 5

