```c
/*
Project 3
Student Name      : Mete Can Gazi
Student No             : 141024020
*/


#include <stdlib.h>
#include <complex.h>
#include <math.h>


double complex func(double complex x){
        return (x*x*x*x) - 3 * (x*x*x) + (x*x) + (x) + 1 ; //Math library has some problem about using of pow so that I wrote
equations directly.
}

void muller(double complex p0 , double complex p1 , double complex p2 , double tolerance){
        //Setting Values
        double complex h1 = p1 - p0;
        double complex h2 = p2 - p1;
        double complex ro_1 = (func(p1) - func(p0)) / h1;
        double complex ro_2 = (func(p2) - func(p1)) / h2;
        double complex d = (ro_2 - ro_1)/(h2 + h1);

        //Variables
        int i=3;
        double complex b;
        double complex D;
        double complex p;
        double complex E;
        double complex h;

        //Information about equations

        printf("p0 = %f , p1 = %f , p2 = %f , tol. = %f\n\n" , p0 , p1 , p2 , tolerance);
        printf("No:\t        \tPi(x + yi)\t            \tf(pi)(x + yi)\n");
        printf("___                 _____          _____\n");

        while(i<200){
                b = ro_2 + (h2*d);
                D = csqrt(b*b -4 * func(p2) * d);
```

```c
        if( fabs(b - D) < fabs(b + D) ){
                E = b + D;
        }else{
                E = b - D;
        }

        h = (-2*func(p2))/E;
        p = p2 + h;

        printf(" %d.\t\t%f , %f\t\t%f , %f\n",i,creal(p),cimag(p),creal(func(p)),cimag(func(p)));

        if( (pow(creal(h),2) + pow(cimag(h),2)) < (tolerance*tolerance)){
                break;
        }

        p0 = p1;
        p1 = p2;
        p2 = p;
        h1 = p1 - p0;
        h2 = p2 - p1;
        ro_1=(func(p1)-func(p0))/h1;
        ro_2=(func(p2)-func(p1))/h2;
        d=(ro_2-ro_1)/(h2+h1);
        i = i + 1;
    }
}

int main(){
        double tolerance = 0.00001;
        muller(0.5 , -0.5 , 0 , tolerance);
        printf("\n\n");
        muller(0.5 , 1 , 1.5 , tolerance);
        printf("\n\n");
        muller(1.5 , 2 , 2.5 , tolerance);
        return 0;
}
```