Project 1

Section 1 - Problem 1
Section 2 - Problem 1

Group 8 – Java:
Peter Boukhalil
Luong Dang
Kimberly Nivon
Julia Beatriz Ramos
Nhi Vu

Section 1 - Problem 1

Section 1

In all of these problems use a tolerance $\epsilon = 10^{-6}$.

1. In studies of solar-energy collection by focusing a field of plane mirrors on a central collector, one researcher obtained this equation for the geometrical concentration factor C:

$$C = \frac{\pi \left(h/\cos A\right)^2 F}{0.5\pi D^2 \left(1 + \sin A - 0.5\cos A\right)}$$

where A is the rim angle of the field, F is the fractional coverage of the field with mirrors, D is the diameter of the collector, and h is the height of the collector. Find A if $h = 300$, $C = 1200$, $F = 0.8$, and $D = 14$.

Section 1 - Problem 1

Let substitute *h=300,* C=120, *F=0.8,* and *D=14* to the equation for the geometrical concentration factor C:

$$\frac{\pi\left(\frac{300}{cosA}\right)^2(0.8)}{0.5\pi(14)^2(1+sinA-0.5cosA)} = 1200$$

$$\frac{72000/(cosA)^2}{98(1+sinA-0.5cosA)} = 1200$$

$$\frac{1}{(1+sinA-0.5cosA)(cosA)^2} = \frac{49}{30}$$

$$\frac{49}{30}(1+sinA-0.5cosA)(cosA)^2 - 1 = 0$$

Then we get a new nonlinear equation with an unknown variable A, to find A, we can use the bisection method, which:

A function is: $\frac{49}{30}(1+sinA-0.5cosA)(cosA)^2 - 1 = 0$

Tolerance: 10^{-6}

Interval for A: [0, 2π] (since A is an angle)

Section 1 - Problem 1 Solution

```java
class Section1Problem1 {

    static double f(double x) { // to be used for bisection method
        return (49.0/30.0)*Math.pow(Math.cos(x),2)*(1+Math.sin(x)-0.5*Math.cos(x))-1;
    }

    static void bisection(double a, double b, double TOL) { // bisection to find the value
        double x = a;
        while (!(Math.abs(b - a) < TOL)) { // for bisection to find a
            x = (a + b) / 2.0;
            if (f(x) * f(a) < 0) b = x;
            else a = x;
        }
        System.out.print("The value of A is : " + x); //prints out message
    }

    public static void main(String[] args) {
        bisection( a: 0,  b: 2*Math.PI, Math.pow(10, -6)); // calls bisection between 0 2pi with tolerance to 10 and -6
    }
}
```

Section 1 - Problem 1 Solution Output

```
"F:\Program Files\Java\jdk1.8.0_211\bin\java.exe" ...
The value of A is : 2.139366156021898
Process finished with exit code 0
```

Section 2 - Problem 1

Section 2

In all of these problems use a tolerance $\epsilon = 10^{-6}$.

1. The Chebyshev polynomials, $T_i(x)$, are a special class of functions. They satisfy the two-term recurrence relation

$$T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x)$$

with $T_0(x) = 1$ and $T_1(x) = x$.

a. Using the recurrence relation, determine the formula for $T_6(x)$.

b. Locate all the roots of $T_6(x)$

Section 2 - Problem 1 A Solution

```
    //For part A
    static String printFunction(int i) { // recursive function used
        if (i == 0) return "1"; // when it becomes 0 it returns 1
        if (i == 1) return "x"; // when it becomes 1, it returns x
        //T6(x)=
2x(2x(2xT3(x)-T2(x))-(2xT2(x)-T1(x)))-(2x(2xT2(x)-T1(x))-(2xT1(x)-T0(x)))
        return "(2x*"+printFunction(i-1)+"-"+printFunction(i-2)+")"; // if doesn't fall
within if conditions
        // above, will return
    }
```

Section 2 - Problem 1 A Solution Output

```
=========== PART A ===========
Function: (2x*(2x*(2x*(2x*(2x*x-1)-x)-(2x*x-1))-(2x*(2x*x-1)-x))-(2x*(2x*(2x*x-1)-x)-(2x*x-1)))
Simplified: 32x^6 - 48x^4 + 18x^2 -1
```

Section 2 - Problem 1 B Solution

```java
//For part B
static double func(int i, double x) { // recursive function used, does the same thing but with real values
    if (i == 0) return 1; // when it becomes 0, it will return 1
    if (i == 1) return x; // when it becomes 1, it will return x
    //T6(x)= 2x(2x(2xT3(x)-T2(x))-(2xT2(x)-T1(x)))-(2x(2xT2(x)-T1(x))-(2xT1(x)-T0(x)))
    return 2 * x * func( i: i-1,x) - func( i: i-2,x); // if doesn't fall within if conditions above, will return
}
//For part B
static double derivFunc(double x) { // derivative for newtons
    return 192 * Math.pow(x, 5) - 192 * Math.pow(x, 3) + 36 * x;
}
//For part B
static void newton(double x, double TOL) { // newtons methods - preferred method to find root
    double c = func( i: 6,x) / derivFunc(x);
    while (Math.abs(c) >= TOL) {
        c = func( i: 6,x) / derivFunc(x);
        x = x - c;
    }
    System.out.println("A root of T_6(x) is: " + x); // prints out the root
}
```

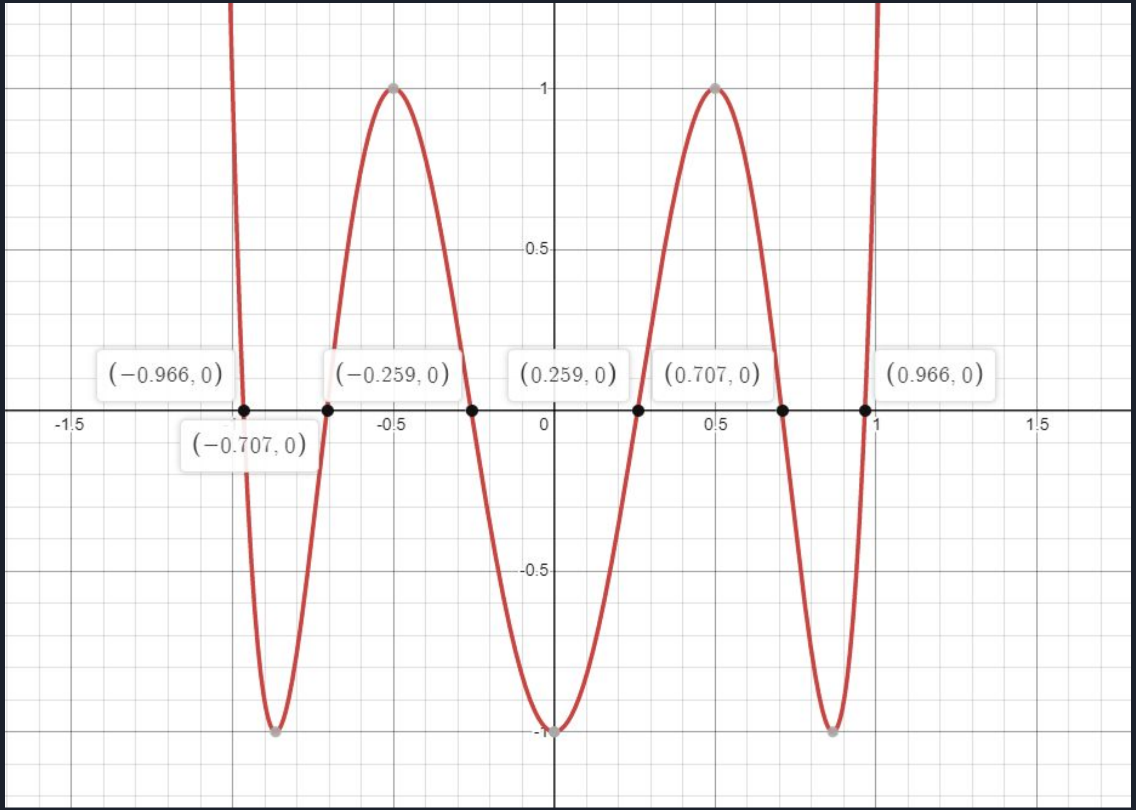
Section 2 - Problem 1 Solution

```
public static void main(String[] args) {
    System.out.println("=========== PART A ==========="); // to print out format - for part A
    System.out.println("Function: " + printFunction( i: 6)); // use of recurrence relation
    System.out.println("Simplified: 32x^6 - 48x^4 + 18x^2 -1\n"); //prints out simplified function at root 6

    // By ploting fuction, we find the roots near 1,-1,0.7,-0.7,0.3,-0.3 Initial
    // values assumed
    //For part B
    System.out.println("=========== PART B ==========="); // to print out format
    double TOL = Math.pow(10, -6); // defining what tol is
    // initial values needed for newtons
    newton( x: 1, TOL); // prints value at 1
    newton( x: -1, TOL); // prints value at -1
    newton( x: 0.7, TOL); // prints value at 0.7
    newton( x: -0.7, TOL); // prints value at -0.7
    newton( x: 0.3, TOL); // prints value at 0.3
    newton( x: -0.3, TOL); // prints value at -0.3
    //above corresponds to given graph in slides and report
}
for part b plots, found 6 roots and use newtons methods
```

Section 2 - Problem 1 Solution Output

```
=========== PART A ===========
Function: (2x*(2x*(2x*(2x*(2x*x-1)-x)-(2x*x-1))-(2x*(2x*x-1)-x))-(2x*(2x*(2x*x-1)-x)-(2x*x-1)))
Simplified: 32x^6 - 48x^4 + 18x^2 -1

=========== PART B ===========
A root of T_6(x) is: 0.9659258262910302
A root of T_6(x) is: -0.9659258262910302
A root of T_6(x) is: 0.7071067811865476
A root of T_6(x) is: -0.7071067811865476
A root of T_6(x) is: 0.25881904510252135
A root of T_6(x) is: -0.25881904510252135
```

Section 2 - Problem 1 B Solution - Graph



The End