Name		ID		Class ID	
Date		Time			

ROOT FINDING

Table of Contents

Name		ID		Class ID	
Date		Time			

1 (Root Finding) Hand Calculation

1.1 Find the Square Root of Six ($\sqrt{6}$) to 2 significant figures.

Express your answer as a range, i.e., a < $\sqrt{6}$ < b. where a and b are numbers. Show all steps.

Step		
1. Express as y=f(x)		
2. Solve for approximate solution		

1.2 Find the Cubic Root of Nine ($\sqrt[3]{9}$) to 3 significant figures.

Express your answer as a range, i.e., a< $\sqrt[3]{9}$ <b. Show all steps.

Step		
1. Express as y=f(x)		
2. Solve for approximate solution		

When you have completed the work, ask the TA to stamp. Then please help other students.	**TA Stamp**

Name		ID		Class ID	
Date		Time			

2 (Root Finding) Preparing to Code: UML Diagram

Based on your hand calculations, summarize the procedure using a UML Diagram.

When you have completed the UML diagram. Ask a TA to stamp.	**TA Stamp**

Name		ID		Class ID	
Date		Time			

3 (Root Finding) A First Attempt With Variables

3.1 Commented Code

Write a commented version of the code (not real C-code). When your code compiles without error, ask the TA to stamp.	**TA Stamp**

3.2 ANSI-C Code

Write the C-CODE for solving the problem. When your code compiles, linked and runs without error, ask the TA to stamp .	**TA Stamp**

3.3 Application : Cubic Root of Nine

Use your code to find the Cubic Root of Nine ($\sqrt[3]{9}$) to 3 significant figures. Express your answer as a range, i.e., a<x<b. Show all steps

Step	(x1,y1)	(x2,y2)	(x3,y3)
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
-	Final Answer: < $\sqrt[3]{9}$ <		

Once you have completed this worksheet and have TA to stamp, please help your fellow students to solve the problems with their code. This will not only help your friends but also help yourself in identifying coding errors	**TA Sign**

Name		ID		Class ID	
Date		Time			

4 (Root Finding) Using Pointers to get Input Data

4.1 Write the C-CODE to implement the input functions. (don't use joninlib.h)

Replace getFloat() with your own code. Your code should prompt the user for the next guess of the root .

4.2 Use your program to check the answer for $\sqrt[3]{9}$

This is to check that after changing your code the calculation remains correct.

	$< \sqrt[3]{9} <$	

When your code compiles, linked and runs without error, and you have filled in the table to check the answers haven't changed, ask the TA to stamp.	**TA Sign**

Name		ID		Class ID	
Date		Time			

5 (Root Finding) Automating: Using Repetition and Conditions

5.1 Work through the algorithm to find $\sqrt[3]{9}$

Your answer should be correct to 2 decimal places. To simplify the coding we observe that the maximum possible value for a root is 9 and the minimum possible value is 1.

L	Pseudo-Code	Sample Calculation Result (Line Number Executing)	
1	Set x1=1	**x1=1 (1)**	
2	Calculate y1=f(x1)		
3	Set x2=9		
4	Calculate y2=f(x2)		
5	x3=(x2+x1)/2		
6	y3=f(x3)		
7	If(y1 × y3<0)		
8	{x2 = x3; y2=y3;}		
9	else {x1 = x3; y1=y3;}		
10	Ans good enough? If no, goto 5		
11	HALT		

5.2 Modify your Code to automate the finding of the roots of an equation

Use either the `while()` or `do...while()` repetition structures as well as `if...else..`. to automate the finding of the roots of the equation.

5.3 Use your program to calculate: $\sqrt[3]{9}$ correct to 6 decimal places.

When your code compiles, linked and runs without error, and you have finished the calculation, ask the TA to stamp. If this is being done as a homework assignment, please copy and print your code on the back of this sheet of paper.	**TA Sign**

Name		ID		Class ID	
Date		Time			

6 (Root Finding) Using Arrays to Generalize Root Finder

6.1 Modify the function equ() to help solve $\sqrt[n]{p}$ for n < 4

To do this you will need to create an array of coefficients in the main program and pass the coefficients to function equ() which you will need to modify. Your new function prototype should look like:

```
double equ(double x, double a[]);
```

where `a[]` contains the coefficients of the root to be calculated, i.e.,

$$y = f(x) = a_3 x^3 + a_2 x^2 + a_1 x^1 - a_0 \rightarrow 0$$

For example, if we want to calculate $\sqrt[3]{9}$ then we set $a_0 = 9, a_1 = 0, a_2 = 0, a_3 = 1$ and pass the array `a[9,0,0,1]` to our function.

Note: You Need To Modify Both Function `main()` And `equ()`.

6.2 Modify the main function variables

Replace x1,x2,x3 and y1,y2,y3 with two arrays, one for x and one for y.

6.3 Apply your function to find roots and solutions to the equations.

Root	a[]	Answer
$\sqrt[2]{2}$	`2,0,1,0,0`	`1.414214`
$\sqrt[3]{9}$		
$\sqrt[3]{8}$		
$\sqrt[2]{302}$		
na	$y = f(x) = 3x^3 + 2x^2 + x^1 - 3 \rightarrow 0$	

When your code compiles, linked and runs without error, and you have finished the calculations ask the TA to stamp. If this is being done as a homework assignment, please copy and print your code on the back of this sheet of paper.	**TA Sign**

Name		ID		Class ID	
Date		Time			

7 (Root Finding) Using Records to Link X and Y

7.1 Replace the two arrays, X and Y with a single array of records

NOTE: ONLY MODIFY FUNCTION MAIN()

When your code compiles, linked and runs without error, ask the TA to stamp. If this is being done as a homework assignment, please copy and print your code on the back of this sheet of paper.	**TA Sign**

Name		ID		Class ID	
Date		Time			

8 Appendix

8.1 Equations Short

$$\sqrt{2}\ y = x^2 - 2 \rightarrow 0$$

8.2 Square Root of 2 example

$x = \sqrt{2}$

$x^2 = 2$

$y = x^2 - 2 \rightarrow 0$

8.3 Z-root of X

$$\sqrt[n]{p}$$

$$y = x^n - p$$

$$y = x^n - p \rightarrow 0$$