Find the Root

Description

Write an efficient algorithm to solve the following equation (i.e. find the value of ***x*** (if exists))

$$F(x) = p*e^{-x} + q*\sin(x) + r*\cos(x) + s*\tan(x) + t*x^{2} + u = 0$$

Given that:

1- $0 \le x \le 1$

2- The function is **decreasing** in the given interval

3- $0 \le p, r, u \le 20$ and $-20 \le q, s, t \le 0$

If x is exists, return it, else, return -1.

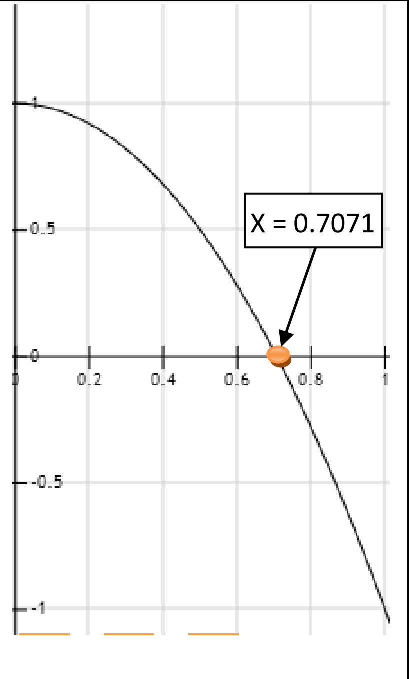
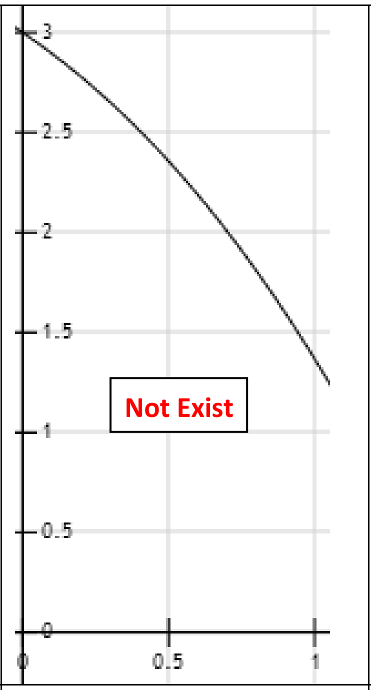
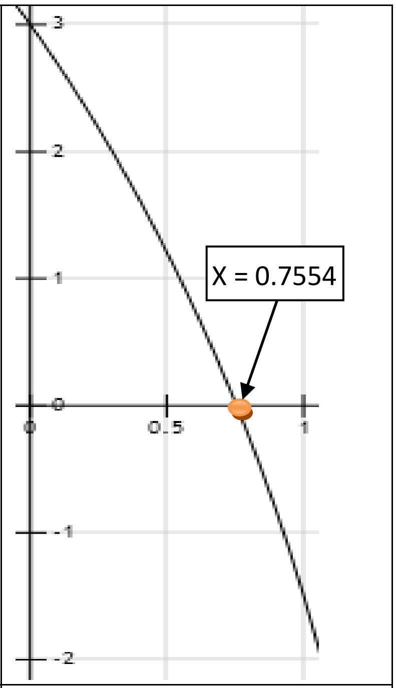
HINT:

Due to the floating point calculations of the above function, its value at the root may equal very small value but not necessarily equal an exact **ZERO**. So, you can consider that the root (x) is found if the function value **F(x)** is:

$$-\,eps < F(x) < eps$$

Where **eps** is a very small value and already defined in the code by 10^{-9}

Examples:

		
p=0, q=0, r=0, s=0, t=-2, u=1	p=1, q=0, r=0, s=0, t=-1, u=2	p=1, q=-1, r=1, s=-1, t=-1, u=1

Input: Already Implemented

Input consists of multiple test cases and terminated by an EOF. Each test case consists of 6 integers in a single line: p, q, r, s, t and u (where $0 <= p,r,u <= 20$ and $-20 <= q,s,t <= 0$).

Output: Already Implemented

For each set of input, there should be a line containing the value of x, correct up-to 4 decimal places, or -1 if not exists, whichever is applicable.

Given Function: Already Implemented

```
static double f(int p, int q, int r, int s, int t, int u, double x)
```

It calculates the above function **F(x)** at a given (`double x`) value with the given six parameters (`p, q, r, s, t` and `u`) and return the result.

Required Function: Implement it!

```
double findTheRoot(int p, int q, int r, int s, int t, int u)
```

It takes the six integers (p, q, r, s, t and u) and should **return the value of the root (x)** that satisfies **F(x)** = 0 (or return -1 if x is not exists).

Template

- C# template

Test Cases

#	Input	Output
1	0 0 0 0 -2 1	0.7071
	1 0 0 0 -1 2	-1
	1 -1 1 -1 -1 1	0.7554
2	16 -1 12 -2 -12 4	-1
	4 -9 10 -2 -4 8	-1
	4 -15 19 0 -5 6	-1
	10 -5 20 -2 -11 4	-1
	16 -4 18 -7 -2 1	-1
	17 0 6 -8 -4 7	-1
	20 -3 5 -6 0 2	-1
	8 -7 18 -3 -12 10	-1
3	1 -20 3 -20 -5 6	0.2347
	2 -20 3 -20 -5 6	0.2521
	3 -20 3 -20 -5 6	0.2689
	4 -20 3 -20 -5 6	0.2850
	5 -20 3 -20 -5 6	0.3005
	6 -20 3 -20 -5 6	0.3154
	3 -4 1 -3 -2 5	0.7863
	6 -11 8 -20 -3 1	0.3807
	4 -4 4 -4 -4 5	0.8016

	17 -6 2 -8 -1 3
	16 -1 12 -2 -12 4
	4 -9 10 -2 -4 8
	4 -15 19 0 -5 6
		0.7628
		-1
		-1
		-1

C# Help

If you need any help regarding the syntax of C#, **ask any TA**.

Creating 1D array

```
int [] array = new int [size]
```

Creating 2D array

```
int [,] array = new int [size1, size2]
```

Sorting single array

Sort the given array in ascending order

```
Array.Sort(items);
```

Sorting parallel arrays

Sort the first array "master" and re-order the 2nd array "slave" according to this sorting

```
Array.Sort(master, slave);
```