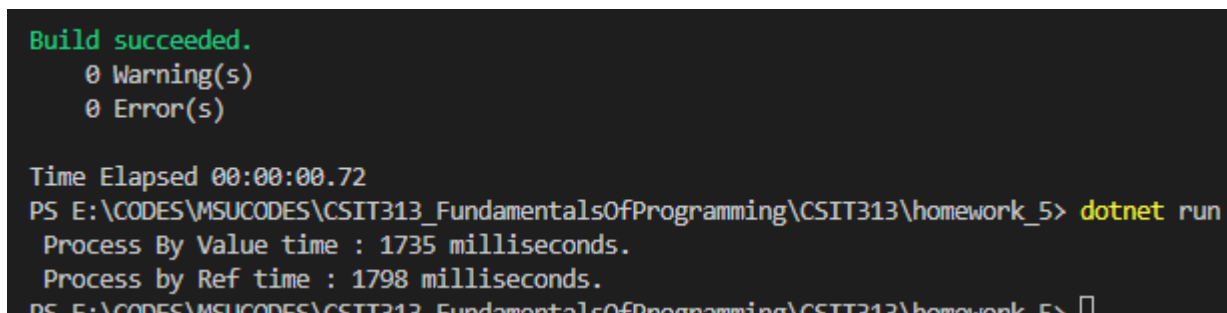


#1:

- (5 pts) Write a C# program that includes two methods that are called a large number of times. Both methods are passed a large array, one by value and one by reference. Compare the times required to call these two methods and explain the difference. Be sure to call them a sufficient number of times to illustrate a difference in the required time. Analyze the running result and reasons caused it. Put code into problem1

Compare the times required to call these two methods and explain the difference. Analyze the running result and reasons...

CODE OUTPUT :



```
Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:00.72
PS E:\CODES\MSUCODES\CSIT313_FundamentalsOfProgramming\CSIT313\homework_5> dotnet run
Process By Value time : 1735 milliseconds.
Process by Ref time : 1798 milliseconds.
PS E:\CODES\MSUCODES\CSIT313_FundamentalsOfProgramming\CSIT313\homework_5>
```

- So Pass by Value took 1.7 seconds and Pass by Reference took 1.8 seconds to run. They both ran 100,000 times and were both working with an array of 1 million integers. As you can see the time is very close in comparison of the runtimes, I believe this is because in C# arrays are reference types so the overhead isn't significant for pass by reference. If the data was primitive types the overhead for this runtime would most likely be more significant but arrays are easy to work with.

#2:

- (5 pts) Write a program, using the syntax of whatever language you like, that produces different behavior depending on whether pass-by-reference or pass-by-value-result is used in its parameter passing. Analyze the differences by looking at the related variable status/value. Put the code into problem2

Code Output:

```
Before Functions are Called :
-----
By Val : 1000, By Ref : 2000

-----
Inside Function PassByRef: Ref = 4000
-----

After PassByRef is Called:
-----
By Val : 1000, By Ref : 4000

-----
Inside Function PassByVal: Val = 2000
-----

After PassByVal is Called:
-----
By Val : 1000, By Ref : 4000
```

So as you can see PassByVal is initially set to 1000 and PassByRef is set to 2000. So when PassByRef is called, ByRef is set to 4000 inside the function as well as outside the function, this is because it directly modifies the original variable. When PassByVal is called ByVal is set to 2000 inside the function, but this change does not translate outside the function so ByVal remains as 1000. This is because a copy of the variable was passed to the function not the original copy(the parameter).

#3:

- #3 Show the stack with all activation record instances, including static and dynamic chains, when executable reaches position 1 in the following skeletal program. Assume bigsub is at level 1.

```
function bigsub()
{
    function a(flag)
    {
        function b()
        {
            // Calls a(false)
            a(false);
        }

        function c()
        {
```

```

        function d()
        {
            // Line 1
        }

        // Calls d()
        d();
    }

    if (flag)
        b(); // Calls b() when flag is true
    else
        c(); // Calls c() when flag is false
}

a(true); // Calls a() with true
}

```

The calling sequence for this program for execution to reach is:

bigsub calls a
 a calls b
 b calls a
 a calls c
 c calls d

Execution Flow Breakdown :

1. bigsub() is called.
2. bigsub() calls a(true) [Level 1].
3. a(true) calls b() [Level 2].
4. b() calls a(false) [Level 3].
5. a(false) calls c() [Level 4].
6. c() calls d() [Level 5]. (THIS is the point when we are at line 1)

Stack/Activation Breakdown :

+-----+	
Activation Record of d() (Level 5)	<-- Top of the stack
- Static Chain: Points to c()	

```

| - Dynamic Chain: Points to c() |
+-----+
| Activation Record of c() (Level 4) |
| - Static Chain: Points to a(false) |
| - Dynamic Chain: Points to a(false) |
+-----+
| Activation Record of a() (Level 3) |
| - Static Chain: Points to b() |
| - Dynamic Chain: Points to b() |
+-----+
| Activation Record of b() (Level 2) |
| - Static Chain: Points to a(true) |
| - Dynamic Chain: Points to a(true) |
+-----+
| Activation Record of a() (Level 1) |
| - Static Chain: Points to bigsub() |
| - Dynamic Chain: Points to bigsub() |
+-----+
| Activation Record of bigsub() (Level 0) |
| - Static Chain: Null |
| - Dynamic Chain: Null |
+-----+

```

#4:

#5:

Started this assignment too late in the day, on the day it was due, and ran out of time for #4 and #5 :(!!!!!