CS-2420-006 Program 6 Canvas Discussion Board (4-10-15)

[JaNell Driscoll](https://uvu.instructure.com/courses/332741/users/487536)

55 unread replies.99 replies.

Has anyone started this project?  I have wrote and tested all the sort functions. However I am having trouble understanding the assignment.  He wants us to read the file in and sort it using insertion sort, then shell sort, then quick sort but after the insertion sort the file is already sorted so the times will be off.  Are we suppose to read the file in for every sort we do or copy the array so we have 3 copies of the same array to sort 3 different ways?  If anyone has any suggestions or did this a better way please let me know. One more question are we suppose to just know that there will be 3 files to be read in or are we suppose to write it for an undisclosed amount of files to sort?

Thank You JaNell

 Unread     Collapse replies Expand replies

[Subscribed](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[ReplyReply to Main Discussion](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Russell Andlauer](https://uvu.instructure.com/courses/332741/users/1248709)

[**Russell Andlauer**](https://uvu.instructure.com/courses/332741/users/1248709)

Thursday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

I am just starting the program today so I have not looked over the instructions too intensely yet. Although, yesterday I revised the directions for the project to make it easier to read.   If anyone would like a copy of that, just send me a message on Canvas.  I spread out the instructions over 4 pages and fixed a few typos etc. I have a feeling that we are not supposed to modify the original file at all, so I think we should re-open the file each time we perform a sort in our programs. It does mention in the instructions that we will create 9 new text files to place the sorted integers in, one for each sort, for every data file.  I hope this helps clarify things a bit, but I may be misinterpretting the directions as well.

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Ryan Pendleton](https://uvu.instructure.com/courses/332741/users/549649)

[**Ryan Pendleton**](https://uvu.instructure.com/courses/332741/users/549649)

Yesterday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

Rather than reopening the file each time you do a sort, I'd just read the file once then copy the array/vector. This will reduce file I/O time.

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Russell Andlauer](https://uvu.instructure.com/courses/332741/users/1248709)

[**Russell Andlauer**](https://uvu.instructure.com/courses/332741/users/1248709)

Thursday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

For your second question, I would assume that we are supposed to write it for an undisclosed amount of files to sort.  The previous 5 projects specify not to hardcode the file names, which implies that you should be able to enter in as many data files as you want.

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Ryan Pendleton](https://uvu.instructure.com/courses/332741/users/549649)

[**Ryan Pendleton**](https://uvu.instructure.com/courses/332741/users/549649)

Thursday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

After I read the file, I just copy the array before each sort. I was wondering the same thing about how many files we need to read though..

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Russell Andlauer](https://uvu.instructure.com/courses/332741/users/1248709)

[**Russell Andlauer**](https://uvu.instructure.com/courses/332741/users/1248709)

Thursday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

It sounds like we may need to have a loop to wait for the user to say they don't want to do any more sorting.  The directions do say that you should create 9 text files.

*Edited by*[*Russell Andlauer*](https://uvu.instructure.com/courses/332741/users/1248709)*on Apr 9 at 2:05pm*

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Ryan Pendleton](https://uvu.instructure.com/courses/332741/users/549649)

[**Ryan Pendleton**](https://uvu.instructure.com/courses/332741/users/549649)

Yesterday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

That's what confuses me. If we're supposed to create 9 text files, then I'd assume we're only suppose to read 3 input files.

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Steph Gilchrist](https://uvu.instructure.com/courses/332741/users/1196692)

[**Steph Gilchrist**](https://uvu.instructure.com/courses/332741/users/1196692)

Yesterday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

Hey guys! Do any of you know if it affects the sort time if you use a vector vs. an array? Also what times are you getting for your sorts? My insertion sort for file3 takes forever!

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Ryan Pendleton](https://uvu.instructure.com/courses/332741/users/549649)

[**Ryan Pendleton**](https://uvu.instructure.com/courses/332741/users/549649)

Yesterday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

This is actually a pretty important question. From what I remember, none of the classes we've taken so far have talked about compiler configuration. There are two very important values you can configure in a compiler. The first is the optimization level, and the second is whether the build should be for debug or release.

When you're debugging a program, it's useful to be able to see what's going on with your code. Being able to step through your code line by line and see what happens helps you find bugs, reinforce concepts, and easily understand your code. Sometimes, code will even be written with the sole purpose of helping a developer debug a program. However, more often than not, compilers are smarter than the developers using them. Compilers can find unnecessary variables, dead code, and redundant conditions. Some compilers can even remove or rearrange code to make it more efficient without affecting the functionality of the program. When they do this though, it makes it harder to step through your code line by line. This is one of the reasons optimizations are turned off by default when debugging an application.

Up until now, I've actually had teachers that just tell students to ignore the difference between release and debug builds, but I think this is a terrible stance to take. One of the reasons is that it can give students the wrong idea. For example, anyone who implemented this insertion sort with both arrays and vectors could reasonably decide that vectors are horribly slow. That isn't the case though. Vectors actually use arrays internally, so fundamentally, there's no reason they should be much slower. Your question made me curious about how much slower an insertion sort really could be though, so I went ahead and made a version that used vectors. I ran the third file through my insertion sort, and here's what I found:

With optimization turned on:  
    Array sort: 1.000 seconds  
    Vector sort: 1.016 seconds

With optimization turned off:  
    Array sort: 5.417 seconds  
    Vector sort: 303.036 seconds

That's obviously a huge difference, and it's all because of optimization. I'm actually pretty surprised by this, but it's hard to argue with these numbers. There must be some overhead in the operator[] overload that's removed when compiling with optimizations turned on. You have three solutions:

* + - Turn on debugging. This is a quick solution and will solve the slowness, but you shouldn't be forced into debugging your program with optimizations enabled just to solve this problem.
    - Use arrays rather than vectors. This may be acceptable depending on the project or style guidelines for a project, but not always. And it may take a lot of work depending on how much code uses the vectors.
    - Use vectors as arrays. If you're already using vectors, this is the solution you'll probably see recommended by most developers, which I'll explain below.

Part of the C++11 specification has a requirement that all elements in a vector must be stored contiguously. This means if you get a pointer to the first element of a vector, then the next element will be directly after it. This allows you to treat vectors as an array using the following code. This will allow your code to run much faster.

std::vector<**int**> vector;

// vector style

**for** (**int** i = 0; i < vector.size(); i++)

{

**int** item = vector[i];

}

// array style

**int** \*items = &vector[0];

**for** (**int** i = 0; i < vector.size(); i++)

{

**int** item = items[i];

}

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Steph Gilchrist](https://uvu.instructure.com/courses/332741/users/1196692)

[**Steph Gilchrist**](https://uvu.instructure.com/courses/332741/users/1196692)

Yesterday

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

Thanks Ryan!! This is super helpful! I emailed Sanati and he said he wants it done with an array. Also, if anyone is wondering, he said no hard-coding on the file names and sort the data in ascending order.

*Edited by*[*Steph Gilchrist*](https://uvu.instructure.com/courses/332741/users/1196692)*on Apr 10 at 10:59pm*

[ReplyReply to Comment](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

[Russell Andlauer](https://uvu.instructure.com/about/1248709)

[**Russell Andlauer**](https://uvu.instructure.com/about/1248709)

1:45pm

[Manage Discussion Entry](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)

Wow, dude. Thank you. I would love to see that program you wrote after the break.  This is really helpful.  I am still in the early phases of the project, but this is also just good info to know.  Is turning optimization on/off simply a selection in one of the option menus in Visual Studio or is it something you must select when creating a new .cpp or .h file?  Thanks for all that work.

[Reply](https://uvu.instructure.com/courses/332741/discussion_topics/1281942)