

Test/Inspection Report

Project Information

URL

<https://github.com/Bhupesh-V/30-seconds-of-cpp>

License

30 Seconds of Cpp (aka 30C++) contains simple and brief examples of C++ STL features, each with a description and code snippet. The idea is that other popular sources of C++ information are overwhelming to beginning/self-teaching learners. The main utilization of the 30C++ collection is a Chrome extension, that shows a random example.

This project uses an MIT license, available here: <https://github.com/Bhupesh-V/30-seconds-of-cpp/blob/master/LICENSE>.

Description

I will be doing a code inspection for this project. There are 110 functions in the <algorithm> library and of them, 46 of them have been implemented and have code snippets. My goal is to go through these completed functions and check the accuracy of the snippets.

The project doesn't have a testing suite, likely due to the fragmented nature of 30Cpp. While I do not think I am far enough in the program to write such a suite (I haven't taken CS 361/2), I still think it would benefit from a thorough look over. Because it's purpose to help learners new to C++, it is especially important for them to correct. I will use the project's Contributing Outline as a guide for this investigation.

I will summarize my findings in a report and establish grading scale for the snippets. If there are examples that technically work but could be improved (reduce ambiguity, simplify for beginners, etc.) I will flag them as well. If a significant number of these issues are discovered, I will display the snippet library in graphical format.

Summary of Work Product

Introduction

For this assignment I chose to analyze the quality of the code in the OSS project, 30 Seconds of C++. Due to the simple nature of the project and its relatively young age, it lacks a testing suite for the code snippets it displays for each function. For these reasons, I decided to manually comb through the 46 completed functions in the <Algorithm> library. The primary purpose of this investigation was to determine which of the snippets followed the Contributing Guidelines (summarized below) and if they were suitable for learners new to C++.

Methods

To examine these code snippets, I created an eight-element scorecard with which to grade them. During the course of this assignment, the author of the project requested all contributors add the Time Complexity for 'relevant functions', so this a criterion of my scorecard. An example below, full collection of scorecards at end of document.

Function name	accumulate
Description Quality	Describes function behavior for default and overloaded versions. Attempts to differentiate between the two but the run-on sentence adds confusion.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators for accumulate's first two parameters.
Snippet Quality	Follows contributing standards. Demonstrates both uses of accumulate function: sum (default) and passed binary operation function Short and easy to follow.
Snippet LOC (without comments/headers)	8
Snippet Runs	Snippet works and correctly demonstrates the function's purpose. Shows use of both versions of the function.
Author	Harsh Raj

Description Quality was evaluated in terms of its simplicity, brevity, and completeness. Each snippet was graded on how well it followed the project's Contributing Guidelines and noted in the Snippet Quality section of the scorecard. Finally, the ability of the snippet to run (and properly) is assessed. The presence of other functions (which may need to be explained to a new user), Lines of Code, and the author name are included, as well.

Results & Figures

Through the analysis of these code snippets, there were seven different trends or relationships that surfaced. While my original focus was the adherence of the Contributing Guidelines, I believe that all of these observations are worth mentioning and may warrant further discussion by the project leader.

Author Contributions

Of the 46 implemented snippets, there were 16 different authors (Figure 1). It is interesting to note that an author only writing one submission accounts for over a quarter of the total snippets written. Therefore it is of little surprise that this library is dominated by a single author, Thamara Andrade, who wrote 25 snippets (54%). The runners-up for most contributions only wrote 4, 3, and 2 snippets.

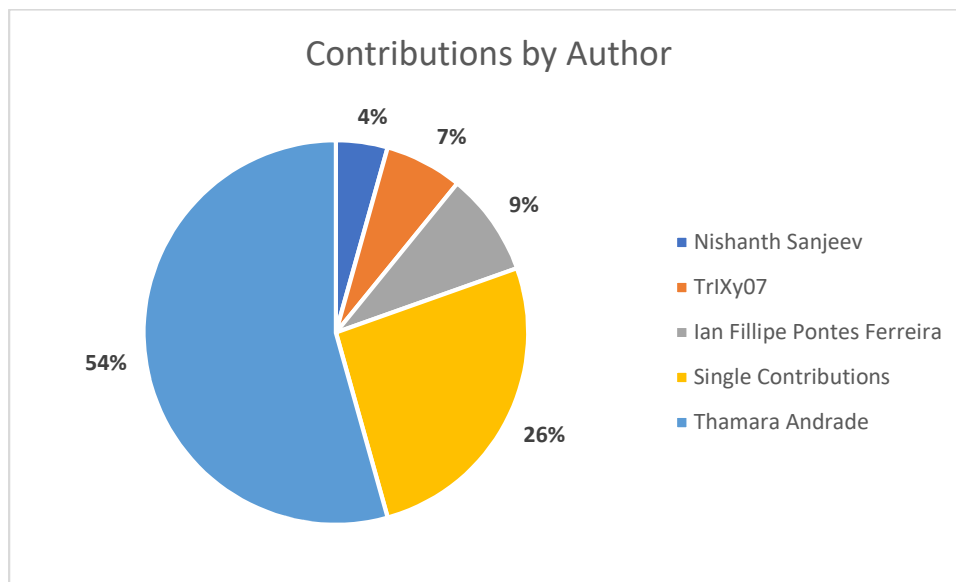


Figure 1

During my evaluation, I also noticed a wide range in the LOC used in these snippets. The shortest snippet had just 5 lines of code, while the longest had 38 lines. Figure 2 displays the average lines used per author. Because 12 of the 16 authors only submitted one snippet, each author is represented by their average snippet LOC. We see that the 46-snippet average is 12.2 LOC, and that the top contributor Thamara has an average of 12.6 lines.

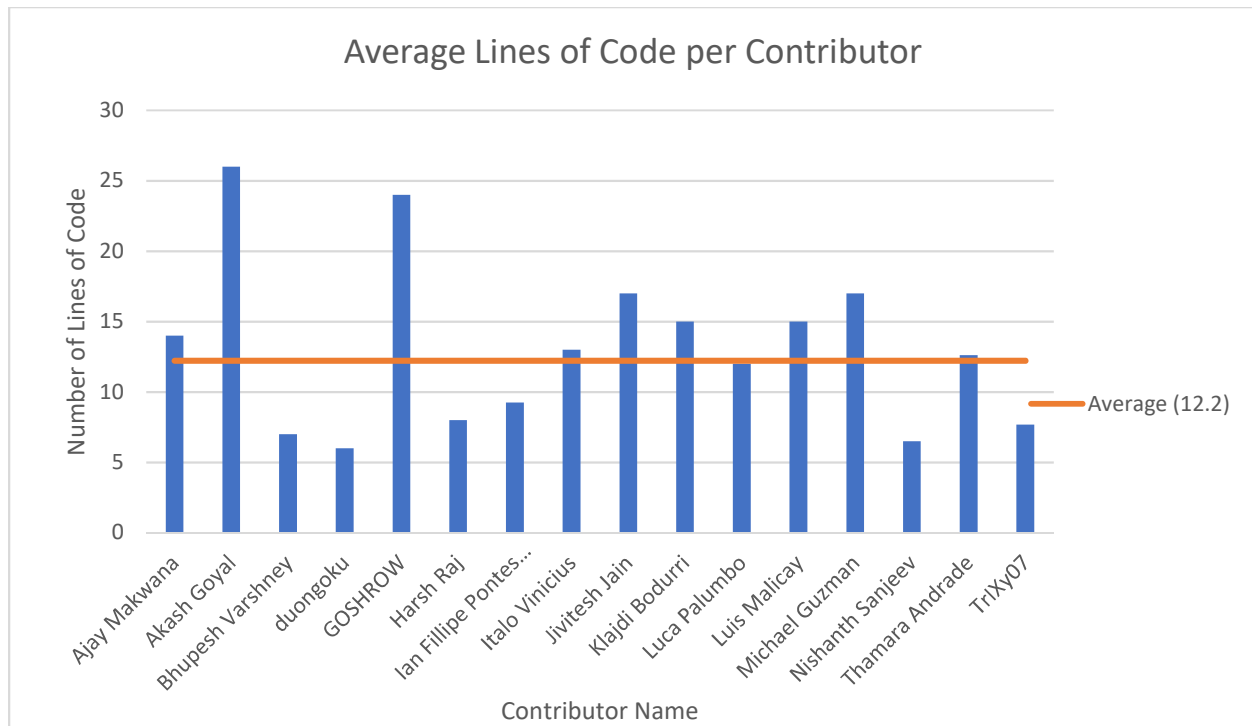


Figure 2

These details on the contributors can grant insight to those overseeing progress on the project. At this point in time, only roughly half of the library's functions have been written. If the goal is for the snippets to be brief, the project manager could choose to tighten the Contributing Guidelines with a LOC cap.

Function Description Quality

Each snippet is required to have a short, yet complete, description in its file header. During the evaluation of this criteria, it was discovered that many of the functions had incomplete descriptions (Figure 3). While some had minor issues such as a typo, two main deficiencies were observed: Sorting and Overloading.

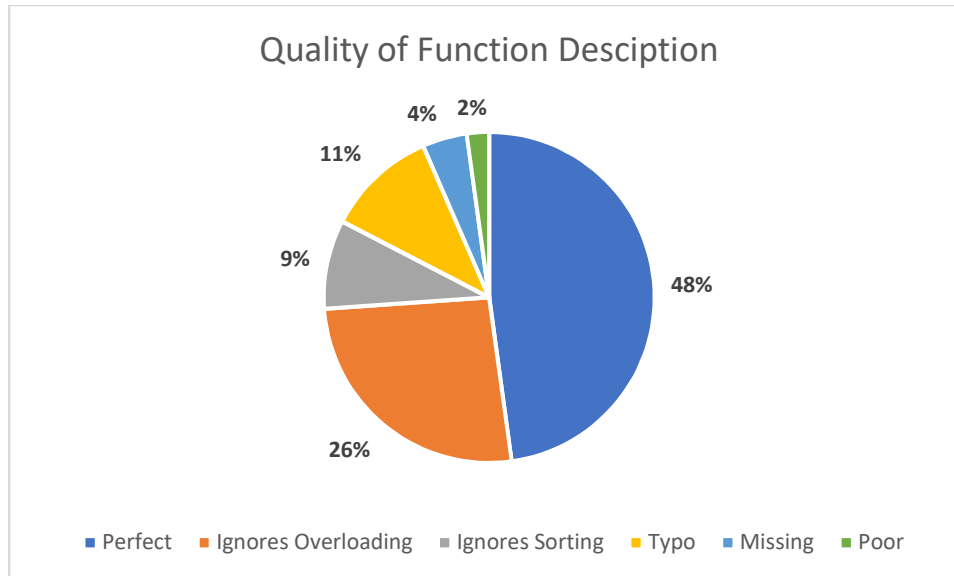


Figure 3

Most of the functions that required a list to be sorted prior to use (ie `equal_range()`) did not state this in the description, however it was present in the code's inline comments. The more egregious omission concerned overloading. Nearly a third of the functions (18 snippets) had the option to overload, to use a custom comparator or other binary operator instead of the default one (ie `min_element`). Figure 4 shows that only 17% of the overload-able functions mentioned this in the description.

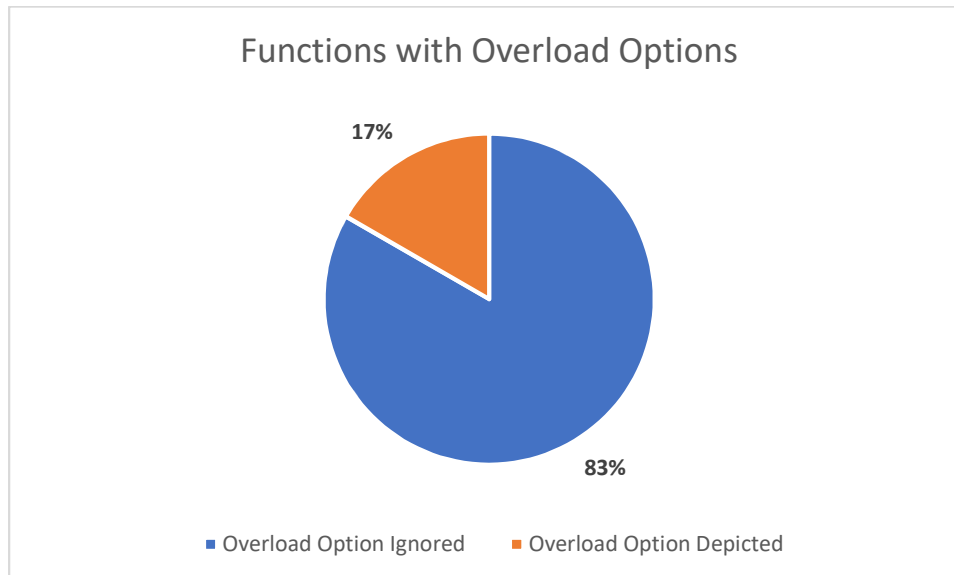


Figure 4

This information may or may not be of concern for the project manager. They could determine that mentioning the overloaded function are not in line with their design philosophy. Perhaps the concept is too advanced for the intended audience – those new to C++. On the other hand, this information could at least catalyze a change for the snippets to be consistent; either including details on overloaded functions or not.

Contributing Guidelines

By and large, the Contributing Guidelines are well followed in this project. The description comments made in the previous section are a matter personal design choice, and overall have little impact on the actual style guidelines. Over 80% of the code snippets followed them perfectly (Figure 5). The main offenders were missing the file header, using namespace instead of `std::` statements, or use of next-line braces.

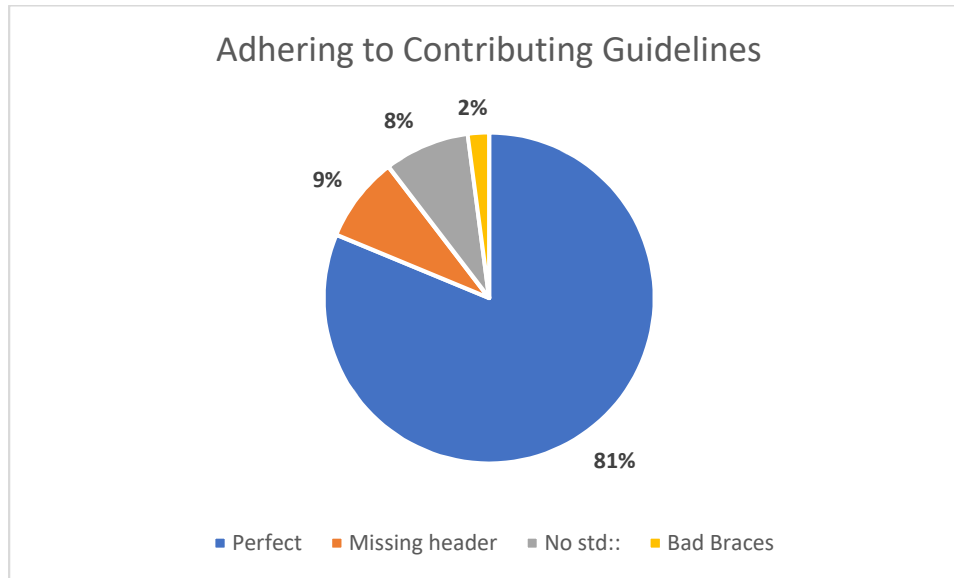


Figure 5

Snippet Quality

To test the quality of the snippets, I opened them in Visual Studio and attempt to run them. A surprising number of them, 12%, were unable to run as written (Figure 6). Five of the snippets were missing crucial `#include` statements to run (algorithm, iostream, etc), although this is an easy fix. The other snippet that would not run attempted to define an array of non-constant length. This could be solved by dynamic allocation (which is likely too advanced for a beginner to comprehend) or simply hardcoding an array guaranteed to be large enough.

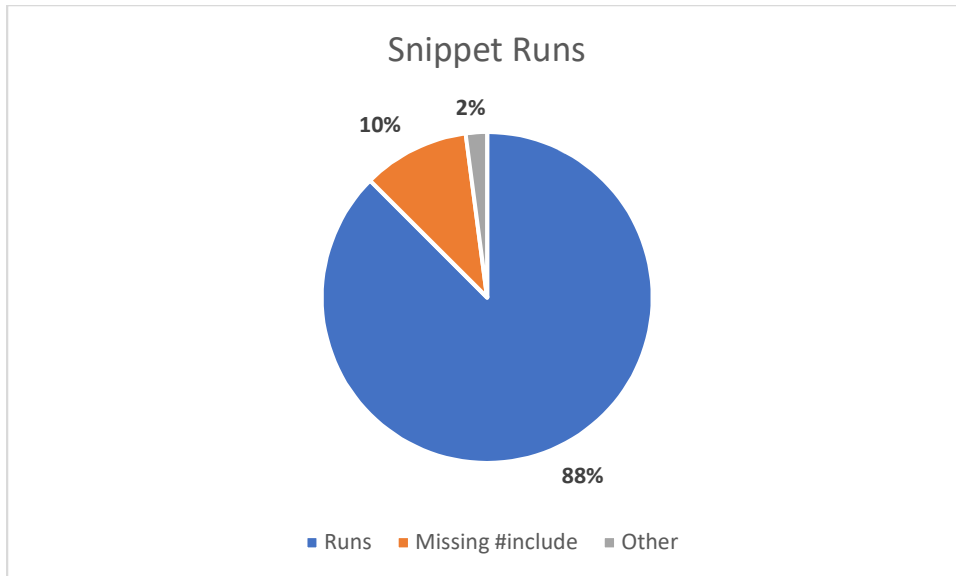


Figure 6

Finally is the inclusion of besides the example function being included in the snippet. Figure 7 shows that 62% of the snippets use other library functions, and another 21% use the build in vector functions, like `vector.begin`. Because these other functions are undefined in the code snippet, a new learner could easily be confused when attempted to follow the code. The Contributing Guidelines could be amended to minimize such references, should the project overseers so choose.

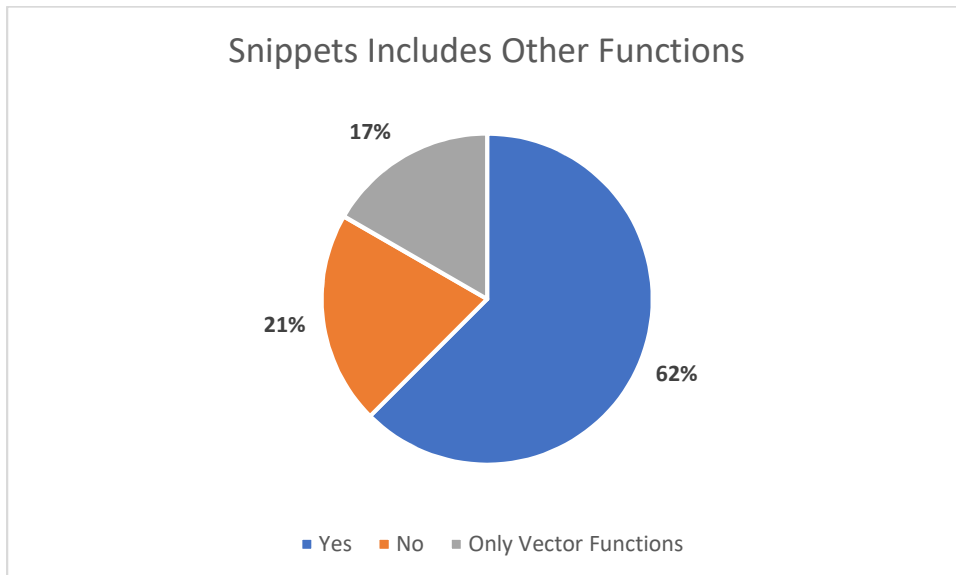


Figure 7

Conclusion

On the whole, this young project follows the Contributing Guidelines fairly well, but is by no means perfect. Many of the relationships reported above could be used to make decisions for the entire project

(not just the <algorithm> library examined here). It is my suggestion that whichever way the project manager aims for this project to go, to explicitly outline them in the guideline documents. This is especially important for heavy contributors, who can be easily be responsible for large portions of the codebase.

A final point worth mentioning is the 47th code snippet in this investigation. One of the snippets, `is_sorted`, is fundamentally flawed and was not included in the above calculations and figures. The entire contents of the file (header, description, code) is for an entirely different function, `is_sorted_until`. While these two share a similar name, and a mix up is what likely occurred, they operate completely differently. At the conclusion of this report, I will be submitting this in a bug report to the project.

Referenced Material

Contributing Guidelines

<https://github.com/Bhupesh-V/30-seconds-of-cpp/blob/master/CONTRIBUTING.md>

Style Guide

Follow this style guide to add sample programs:

1. Naming Style: For example if you are demonstrating `erase` in `list` name your file `erase.md` and save it in the `list` directory.
2. For adding Code Snippets in the `snippets` directory. Add the following comment header in every program. The comment header should always be at the top of program.

```
/*
  Author : this must be your name ;)
  Date  : Date format dd/mm/yyyy
  Time  : Time format hh:mm
  Description : description should be small & one liner.
*/
```

3. Add Opening braces on the same line.

```
int main()
{
    // ✗
}

int main(){
    // ✓.
```



```
}
```

4. Indentation : Use **1 Tab** or **4 Spaces**. Be consistent with whatever you choose Use only one indenting format for the whole program.

5. Add appropriate comments wherever necessary to explain the code.

Programs with NO Comments at all will not be merged.

6. Expression should be readable, Use 1 space between different TOKENS.

```
galaxy=stars+asteroids    // ✗  
galaxy = stars + asteroids    // ✓.
```

7. Always add braces in a for/while loop, even if it's a one-liner.

```
for(int i=0;i<45;i++)  
    cout<<i<<" ";        // ✗  
  
for(int i=0;i<45;i++){  
    cout<<i<<" ";        // ✓.  
}
```

8. Always use pre-increment(++i) while looping instead of post-increment(i++).

9. When submitting markdown files of methods in different directories, name your file exactly as the function/method name. For example push_back.md, swap.md etc.

10. Always use prefix std:: for functions and types from the std namespace, either on the markdown files and on the snippets files.

```
using namespace std;  
vector<int> ...;    // ✗  
cout << ...;    // ✗  
  
std::vector<int> ...;    // ✓  
std::cout << ...;    // ✓
```

Scorecard Collection

Function name	accumulate
Description Quality	Describes function behavior for default and overloaded versions. Attempts to differentiate between the two but the run-on sentence adds confusion.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators for accumulate's first two parameters.
Snippet Quality	Follows contributing standards. Demonstrates both uses of accumulate function: sum (default) and passed binary operation function Short and easy to follow.
Snippet LOC (without comments/headers)	8
Snippet Runs	Snippet works and correctly demonstrates the function's purpose. Shows use of both versions of the function.
Author	Harsh Raj

Function name	adjacent_difference
Description Quality	Description is concise but accurate. Gives general idea of function use, leaves specific details for inline comments in snippet
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. The <functional> function multiples() as an operation for adjacent_difference's overload option (to override the default difference operation).
Snippet Quality	Follows contributing standards. Demonstrates both uses of adjacent_difference function: sum (three parameters) and passed binary operation function (four parameters). Includes an example for each a pre-built and a custom-made operation. Easy to follow.
Snippet LOC (without comments/headers)	24
Snippet Runs	Snippet works and correctly demonstrates the function's purpose. Shows use of both versions of the function.
Author	GOSHRW

Function name	copy
---------------	------

Description Quality	Description is concise but accurate. Completely covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. The <iterator> function back_inserter() as an operation to add values to the destination container.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Short and easy to follow.
Snippet LOC (without comments/headers)	9
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Thamara Andrade

Function name	copy_backward
Description Quality	Description is concise but accurate. Completely covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators for copy_backward's parameters.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Short and easy to follow.
Snippet LOC (without comments/headers)	9
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Thamara Andrade

Function name	copy_if
Description Quality	Description is concise but accurate. Completely covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. The <iterator> function back_inserter() as an operation to add values to the destination container. This is in addition to <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Uses custom function as binary operator. Short and easy to follow.
Snippet LOC (without comments/headers)	13

Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Thamara Andrade

Function name	copy_n
Description Quality	Description is concise but accurate. Completely covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. The <iterator> function back_inserter() as an operation to add values to the destination container. This is in addition to <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Short and easy to follow.
Snippet LOC (without comments/headers)	9
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Thamara Andrade

Function name	count
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators for count's parameters.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Short and easy to follow.
Snippet LOC (without comments/headers)	7
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Bhupesh Varshney

Function name	equal
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators for equal's parameters.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Fairly easy to follow. The if statements are a bit busy and could be difficult

	for someone new to C++ to easily grasp, especially with the lack of parameter description: <code>if(std::equal(v1.begin(),v1.end(),v2.begin()))</code> . Could better explain the parameters of the function (start of first vector, end of first vector, start of second vector).
Snippet LOC (without comments/headers)	17
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Michael Guzman

Function name	<code>equal_range</code>
Description Quality	Description is concise but accurate. Covers all aspects of function utility. Requirement for list to be pre-sorted missing from description, but present in the snippet's inline comments.
Time Complexity (if applicable)	Absent. At most $O(\log_2(\text{first-last}))$
Includes other Functions	Yes. Uses the <code><algorithm></code> function <code>sort()</code> to prepare the list for <code>equal_range()</code> to examine. This is in addition to the <code><vector></code> functions <code>begin</code> and <code>end</code> as the iterator arguments.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Fairly easy to follow. Uses 'auto' as a placeholder type specifier, which could be confusing to someone new to C++.
Snippet LOC (without comments/headers)	8
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Thamara Andrade

Function name	<code>fill</code>
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <code><vector></code> functions <code>begin</code> and <code>end</code> as the iterators for <code>count</code> 's parameters.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow. Minor typos
Snippet LOC (without comments/headers)	13
Snippet Runs	Snippet works and correctly demonstrates the function's purpose of filling a list with a value.
Author	Italo Vinicius

Function name	fill_n
Description Quality	Description does not at all describe the function. Merely states "Show an example of fill_n".
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and size as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	15
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Klajdi Bodurri

Function name	is_sorted
Description Quality	Contents of this file are actually for the function is_sorted_until <i>not</i> the is_sorted function. Will be submitting this as a bug.
Time Complexity (if applicable)	
Includes other Functions	
Snippet Quality	
Snippet LOC (without comments/headers)	
Snippet Runs	
Author	Abhay Singh Yadav

Function name	lexicographical_compare
Description Quality	Description missing
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Does not follow contributing standards: Missing file header. Demonstrates full use of the function (both default and passed in comparators). Easy to follow and understand.
Snippet LOC (without comments/headers)	17
Snippet Runs	Snippet does not work. Missing #include <cctype> to use tolower() function.
Author	Jivitesh Jain

Function name	lower_bound
Description Quality	Description is concise but accurate. Covers all aspects of function utility. Requirement for list to be pre-sorted missing from description, but present in the snippet's inline comments.
Time Complexity (if applicable)	Absent. $\log_2 n + 1$

Includes other Functions	Only the <vector> functions begin and end as the passed arguments.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	6
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Thamara Andrade

Function name	make_heap
Description Quality	Description missing
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin, front, and end.
Snippet Quality	Does not follow contributing standards: Missing file header. Demonstrates full use of the function. Easy to follow and understand.
Snippet LOC (without comments/headers)	7
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	TriXy07

Function name	max
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	No.
Snippet Quality	Snippet works and correctly demonstrates the function's purpose.
Snippet LOC (without comments/headers)	5
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	Thamara Andrade

Function name	max_element
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Snippet works and correctly demonstrates the function's default purpose. Does not show overloaded function.

Snippet LOC (without comments/headers)	5
Snippet Runs	Snippet works and correctly demonstrates the function's purpose by returning the largest element.
Author	Thamara Andrade

Function name	merge
Description Quality	Description is concise but accurate. Covers all aspects of function utility. Requirement for list to be pre-sorted missing from description, but present in the snippet's inline comments.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. Uses the <algorithm> function sort() to prepare the list for merge() to examine. This is in addition to the <vector> functions begin, size, and end.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Easy to follow.
Snippet LOC (without comments/headers)	14
Snippet Runs	Snippet works and correctly demonstrates the correct use of the function by returning the merged list.
Author	Thamara Andrade

Function name	min
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	No.
Snippet Quality	Follows contributing standards. Demonstrates correct use of the function by returning the smaller of the two values. Extremely easy to follow.
Snippet LOC (without comments/headers)	5
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose. Does not show overloaded function.
Author	Thamara Andrade

Function name	min_element
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	N/A (Linear)

Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates both the default (three parameters) and passed binary operation function (four parameters). Short and easy to follow.
Snippet LOC (without comments/headers)	5
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by returning the smallest element in the list. Does not show overloaded function.
Author	Thamara Andrade

Function name	minimax_element
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the parameters.
Snippet Quality	Follows contributing standards. Demonstrates correct use of the default function by returning the smallest and largest values of the list. Extremely easy to follow.
Snippet LOC (without comments/headers)	9
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by returning the smallest element in the list. Does not show overloaded function.
Author	Ian Fillipe Pontes Ferreira

Function name	minmax
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	No.
Snippet Quality	Follows contributing standards. Demonstrates correct use of the default function by returning the smallest and largest values of the list. Extremely easy to follow.
Snippet LOC (without comments/headers)	9
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by returning the smallest element in the list. Does not show overloaded function.

Author	Thamara Andrade
--------	-----------------

Function name	minmax_element
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	6
Snippet Runs	Snippet works and correctly demonstrates the function's purpose of returning the smallest and largest elements in a list.
Author	Thamara Andrade

Function name	nth_element
Description Quality	Description is a run-on sentence and doesn't clearly convey the function's purpose or mention overloading.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	No.
Snippet Quality	Does not follow contributing standards: "using namespace std" and the opening brace of main() is not on the same line. Demonstrates full use of the function, although inline comments do not fully explain the function's execution.
Snippet LOC (without comments/headers)	9
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by placing the specified element in the proper position of the list. Does not show overloaded function.
Author	Ian Fillipe Pontes Ferreira

Function name	pop_heap
Description Quality	No Description.
Time Complexity (if applicable)	Absent. At most $2 \cdot \log(n)$
Includes other Functions	Yes, the function make_heap(). This is in addition to the <vector> functions begin, front, and end.
Snippet Quality	Does not follow contributing standards: is missing the author information header.

	Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	8
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by removing the top element in the heap.
Author	TriXy07

Function name	random_shuffle
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with random generator.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Does not follow contributing standards: 'using namespace std'. Demonstrates default use of the function. Does not show overloaded function. Very easy to follow.
Snippet LOC (without comments/headers)	14
Snippet Runs	Snippet does not work. Needs to #include <algorithm> and <time.h>. Does not show overloaded function.
Author	Ajay Makwana

Function name	remove
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin, erase, and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	9
Snippet Runs	Snippet works and correctly demonstrates the function's purpose of returning the list with selected elements removed.
Author	Thamara Andrade

Function name	remove_copy
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)

Includes other Functions	Yes. The <iterator> function back_inserter() as an operation to add values to the destination container. This is in addition to the <vector> functions begin and end.
Snippet Quality	Follows contributing standards. Short and very easy to follow.
Snippet LOC (without comments/headers)	17
Snippet Runs	Snippet works and correctly demonstrates the function's purpose of removing elements from a list that meet a specified condition.
Author	Thamara Andrade

Function name	remove_copy_if
Description Quality	Description is concise but accurate. Covers all aspects of function utility. Minor typo.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. The <iterator> function back_inserter() as an operation to add values to the destination container. This is in addition to the <vector> functions begin and end.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	20
Snippet Runs	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Author	Thamara Andrade

Function name	remove_if
Description Quality	Description is concise but accurate. Covers all aspects of function utility. Minor typo.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin, erase, and end as the iterators.
Snippet Quality	Follows contributing standards. Short and very easy to follow.
Snippet LOC (without comments/headers)	12
Snippet Runs	Snippet works and correctly demonstrates the function's purpose of removing elements from a list that meet a specified condition.
Author	Thamara Andrade

Function name	replace
---------------	---------

Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	12
Snippet Runs	Snippet works and correctly demonstrates the function's purpose by replacing all specified values with a new specific value.
Author	Thamara Andrade

Function name	replace_copy
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. The <iterator > function back_inserter() as an operation to add values to the destination container. This is in addition to the <vector> functions begin and end.
Snippet Quality	Follows contributing standards. Short and very easy to follow.
Snippet LOC (without comments/headers)	19
Snippet Runs	Snippet works and correctly demonstrates the function's purpose of removing elements form a list that meet a specified condition.
Author	Thamara Andrade

Function name	replace_copy_if
Description Quality	Description is concise but accurate. Covers all aspects of function utility. Minor typo.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. The <iterator > function back_inserter() as an operation to add values to the destination container. This is in addition to the <vector> functions begin and end.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	22
Snippet Runs	Snippet works and correctly demonstrates the function's purpose of removing elements form a list that meet a specified condition.

Author	Thamara Andrade
--------	-----------------

Function name	replace_if
Description Quality	Description is concise but accurate. Covers all aspects of function utility. Minor typo.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	15
Snippet Runs	Snippet works and correctly demonstrates the function's purpose of replacing values that satisfy a specified condition.
Author	Thamara Andrade

Function name	rotate
Description Quality	Description is concise and generally correct. Gives general idea of function use but would be more accurate to say the specified element position becomes the first, rather than imply it is the number of times being rotated.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin, size, and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Fairly easy to follow. Shows rotation in both directions.
Snippet LOC (without comments/headers)	26
Snippet Runs	Snippet works and correctly demonstrates the function's purpose by rotating the positions of a vector's element left and right.
Author	Akash Goyal

Function name	search
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates default use of the function. Does not show overloaded function. Very easy to follow.

Snippet LOC (without comments/headers)	12
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by returning the start position of a specific subarray in the list. Does not show overloaded function.
Author	Luca Palumbo

Function name	set_difference
Description Quality	Description is concise but accurate. Gives general idea of function use, leaves specific details for inline comments in snippet (such as the requirement that the input lists be sorted). Could also clarify that this is a directional operation ($A \cap B \neq B \cap A$)
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin, size, resize, and end.
Snippet Quality	Snippet works and correctly demonstrates the function's default purpose. Does not show overloaded function.
Snippet LOC (without comments/headers)	16
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by returning the items in the first list that are absent from the second. Does not show overloaded function.
Author	Thamara Andrade

Function name	set_intersection
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin, size, resize, and end.
Snippet Quality	Snippet works and correctly demonstrates the function's default purpose. Does not show overloaded function.
Snippet LOC (without comments/headers)	16
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by returning the items in the first list that are common between the second. Does not show overloaded function
Author	Thamara Andrade

Function name	set_union
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Only the <vector> functions begin, size, resize, and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates default use of the function. Fairly easy to follow.
Snippet LOC (without comments/headers)	16
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by returning the smallest element in the list. Does not show overloaded function.
Author	Thamara Andrade

Function name	shuffle
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes. Several functions used to seed clock and random number generation. Only the <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates full use of the function. Part of the snippet could be difficult for a person new to C++ to decipher. Specifically the line: "unsigned seed = std::chrono::system_clock::now().time_since_epoch().count();" . Other portions are much easier to read.
Snippet LOC (without comments/headers)	9
Snippet Runs	Snippet works and correctly demonstrates the function's purpose. Values appear random after multiple executions.
Author	Ian Fillipe Pontes Ferreira

Function name	sort
Description Quality	Description is concise but accurate. Covers default use of function, but not overloading with custom binary operator.
Time Complexity (if applicable)	Absent. Up to $n \cdot \log_2(n)$
Includes other Functions	Only the <vector> functions begin and end as the iterators.

Snippet Quality	Follows contributing standards. Demonstrates default use of the function (ascending sort).
Snippet LOC (without comments/headers)	15
Snippet Runs	Snippet works and correctly demonstrates the function's default purpose by returning the list in ascending order. Does not show overloaded function.
Author	Luis Malicay

Function name	stable_sort
Description Quality	No description.
Time Complexity (if applicable)	Absent. Up to $n \cdot \log_2(n)$
Includes other Functions	No.
Snippet Quality	Does not follow contributing standards: missing author information header. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	8
Snippet Runs	Snippet does not run. Needs to <code>#include <iostream></code> and <code><algorithm></code> .
Author	TriXy07

Function name	swap
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Constant [non-array] or Linear [array])
Includes other Functions	No.
Snippet Quality	Follows contributing standards. Demonstrates full use of function by swapping the assignments of two variables. Short and easy to follow.
Snippet LOC (without comments/headers)	6
Snippet Runs	Snippet works and correctly demonstrates the function's purpose.
Author	duongoku

Function name	to_string
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	No.
Snippet Quality	Follows contributing standards. Demonstrates full use of function by converting both ints and floats to strings. Short and easy to follow.
Snippet LOC (without comments/headers)	7

Snippet Runs	Snippet does not run. Needs to #include <iostream> and <string>.
Author	Nishanth Sanjeev

Function name	transform
Description Quality	Describes the snippet's specific function applied (adding), but doesn't mention any binary operator can be applied.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	Yes, plus<int>() to add values, and the <vector> functions begin and end .
Snippet Quality	Does not follow contributing standards: 'using namespace std'. Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	10
Snippet Runs	Snippet does not work. Attempts to define array with variable size: "expression must have a constant value".
Author	Ian Fillipe Pontes Ferreira

Function name	upper_bound
Description Quality	Description is concise but accurate. Covers all aspects of function utility. Requirement for list to be pre-sorted missing from description, but present in the snippet's inline comments.
Time Complexity (if applicable)	Absent. $\log_2 n + 1$
Includes other Functions	Only the <vector> functions begin and end as the iterators.
Snippet Quality	Follows contributing standards. Demonstrates default use of the function. Could likely convey same information in few lines.
Snippet LOC (without comments/headers)	38
Snippet Runs	Snippet works and correctly demonstrates the function's purpose. However, it should be specified that the number returned is the position of the upper_bound value in the list, not the value itself.
Author	Thamara Andrade

Function name	__gcd
Description Quality	Description is concise but accurate. Covers all aspects of function utility.
Time Complexity (if applicable)	N/A (Linear)
Includes other Functions	No.

Snippet Quality	Does not follow contributing standards: 'using namespace std" . Demonstrates full use of the function. Extremely easy to follow.
Snippet LOC (without comments/headers)	6
Snippet Runs	Snippet does not work. Needs to #include <iostream> and add std:: to endl instance.
Author	Nishanth Sanjeev

Copyright

I give the instructor full rights to use these materials for educational purposes or in any way they see fit.