```
Name: Jude Onyia
Student ID: V00947095
Email: judeonyia10@gmail.com
Course: ECE596C
Section: T01

Assignment ID: cpp_compile_time
Assignment Title: Compile-Time Computation

Submission Source: https://github.com/uvic-seng475-2020-05/cpp_compile_time-Jude
Onyia.git
Commit ID: 9873843310751781db8f11e9558bb79da4a7f42f

Submitted Files
===============

drwxrwxr-x    4096 2020-06-01 19:27 ./app
-rw-rw-r--    1658 2020-06-01 19:27 ./app/test_biquad_filter.cpp
-rw-rw-r--    2535 2020-06-01 19:27 ./app/test_cexpr_basic_string.cpp
-rw-rw-r--    1170 2020-06-01 19:27 ./app/test_cexpr_math.cpp
-rw-rw-r--     418 2020-06-01 19:27 ./app/test_mandelbrot.cpp
-rw-rw-r--     651 2020-06-01 19:27 ./CMakeLists.txt
-rw-rw-r--     146 2020-06-01 19:27 ./IDENTIFICATION.txt
drwxrwxr-x    4096 2020-06-01 19:27 ./include
drwxrwxr-x    4096 2020-06-01 19:27 ./include/ra
-rw-rw-r--    6019 2020-06-01 19:27 ./include/ra/biquad_filter.hpp
-rw-rw-r--    9615 2020-06-01 19:27 ./include/ra/cexpr_basic_string.hpp
-rw-rw-r--    2999 2020-06-01 19:27 ./include/ra/cexpr_math.hpp
-rw-rw-r--    2633 2020-06-01 19:27 ./include/ra/mandelbrot.hpp
-rw-rw-r--  464306 2020-06-01 19:27 ./README.pdf

Results
=======

Package          Operation Target           Status
nonprog          generate  ---              OK (0.1s)
string_orig      generate  ---              OK (0.2s)
string_orig      configure ---              OK (0.9s)
string_orig      build     test_cexpr_basic OK (0.5s)
string_orig      build     test_mandelbrot  OK (4.7s)
string_sane      generate  ---              OK (0.2s)
string_sane      configure ---              OK (0.7s)
string_sane      build     test_cexpr_basic FAIL (2 0.5s 407L)
string_sane      build     test_mandelbrot  OK (4.9s)
math_orig        generate  ---              OK (0.1s)
math_orig        configure ---              OK (0.9s)
math_orig        build     test_cexpr_math  OK (1.9s)
math_orig        build     test_biquad_filt OK (4.1s)
math_sane        generate  ---              OK (0.2s)
math_sane        configure ---              OK (0.7s)
math_sane        build     test_cexpr_math  FAIL (2 5.5s 464L)
math_sane        build     test_biquad_filt OK (9.1s)

Normally, an operation is indicated as having a status of either "OK" or
"FAIL".  A status of "?" indicates that the operation could not be performed
for some reason (e.g., due to an earlier error or being a manual step).
The time (in seconds) required for an operation is denoted by an expression
consisting of a number followed by the letter "s" (e.g., "5.0s").
In the case of a test that consists of multiple test cases, the number of
```

```
failed test cases and total number of test cases is expressed as a fraction
(e.g., "10/50" means 10 test cases failed out of 50 test cases in total).
The length (in lines) of the log file generated by an operation is denoted by
an expression consisting of a number followed by the letter "L" (e.g., "10L").
To ascertain the reason for the failure of an operation, check the contents
of the log file provided.

Legend
======

Package: nonprog
    Nonprogramming exercises

Package: string_orig
    The code as originally submitted by the student.
Build target: test_cexpr_basic_string
    Build the test_cexpr_basic_string program.
Build target: test_mandelbrot
    Build the test_mandelbrot program.

Package: string_sane
    Code with modifications to perform API sanity checking.
Build target: test_cexpr_basic_string
    Build the (dummy) test_cexpr_basic_string program.
Build target: test_mandelbrot
    Build the (dummy) test_mandelbrot program.

Package: math_orig
    The code as originally submitted by the student.
Build target: test_cexpr_math
    Build the test_cexpr_math program.
Build target: test_biquad_filter
    Build the test_biquad_filter program.

Package: math_sane
    Code with modifications to perform API sanity checking.
Build target: test_cexpr_math
    Build the (dummy) test_cexpr_math program.
Build target: test_biquad_filter
    Build the (dummy) test_biquad_filter program.
```

```
/home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
-S/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-st
ring_sane/source
-B/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-st
ring_sane/derived --check-build-system CMakeFiles/Makefile.cmake 0
/usr/bin/gmake  -f CMakeFiles/Makefile2 test_cexpr_basic_string
gmake[1]: Entering directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-str
ing_sane/derived'
/home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
-S/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-st
ring_sane/source
-B/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-st
ring_sane/derived --check-build-system CMakeFiles/Makefile.cmake 0
/home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
cmake_progress_start
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/derived/CMakeFiles 2
/usr/bin/gmake  -f CMakeFiles/Makefile2
CMakeFiles/test_cexpr_basic_string.dir/all
gmake[2]: Entering directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-str
ing_sane/derived'
/usr/bin/gmake  -f CMakeFiles/test_cexpr_basic_string.dir/build.make
CMakeFiles/test_cexpr_basic_string.dir/depend
gmake[3]: Entering directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-str
ing_sane/derived'
cd
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/derived &&
/home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
cmake_depends "Unix Makefiles"
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/derived
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/derived
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/derived/CMakeFiles/test_cexpr_basic_string.dir/DependInfo.cmake --color=
Scanning dependencies of target test_cexpr_basic_string
gmake[3]: Leaving directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-str
ing_sane/derived'
/usr/bin/gmake  -f CMakeFiles/test_cexpr_basic_string.dir/build.make
CMakeFiles/test_cexpr_basic_string.dir/build
gmake[3]: Entering directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-str
ing_sane/derived'
[ 50%] Building CXX object
CMakeFiles/test_cexpr_basic_string.dir/app/test_cexpr_basic_string.cpp.o
/home/frodo/public/ugls_lab-4.0.70/bin/clang++
-I/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-st
ring_sane/source/include  -pedantic-errors -std=gnu++17 -o
CMakeFiles/test_cexpr_basic_string.dir/app/test_cexpr_basic_string.cpp.o -c
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/gcc/lib64: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/clang/lib: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/boost/lib: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/catch/lib64: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/CGAL/lib64: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/jasper/lib64: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/clang/lib64: 'linker'
```



```
input unused [-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/gcc/lib64'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/clang/lib'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/boost/lib'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/catch/lib64'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/CGAL/lib64'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/jasper/lib64'
[-Wunused-command-line-argument]
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:14:3: warning: expression result
unused [-Wunused-value]
                sizeof(string);
                ^~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:15:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::value_type);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:16:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::pointer);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:17:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_pointer);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:18:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::reference);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:19:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_reference);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:20:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::iterator);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:21:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_iterator);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:14:3: warning: expression result
unused [-Wunused-value]
                sizeof(string);
                ^~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:111:2: note: in instantiation of
function template specialization 'test_1<char, 128>' requested here
        test_1<char, 128>();
        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:15:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::value_type);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:16:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::pointer);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
```

```
ng_sane/source/app/test_cexpr_basic_string.cpp:17:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_pointer);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:18:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::reference);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:19:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_reference);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:20:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::iterator);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:21:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_iterator);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:26:6: error: object of type
'string' (aka 'cexpr_basic_string<char, 128UL>') cannot be assigned because its
copy assignment operator is implicitly deleted
        s3 = s;
             ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:44:34: note: explicitly
defaulted function was implicitly deleted here
                        constexpr cexpr_basic_string& operator=(const
cexpr_basic_string&) = default;                                  ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:238:21: note: copy assignment
operator of 'cexpr_basic_string<char, 128>' is implicitly deleted because field
'nullChar_' is of const-qualified type 'const
ra::cexpr::cexpr_basic_string<char, 128>::value_type' (aka 'const char')
                        const value_type nullChar_ = value_type(0);
                        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:27:6: error: object of type
'string' (aka 'cexpr_basic_string<char, 128UL>') cannot be assigned because its
copy assignment operator is implicitly deleted
        s4 = std::move(s3);
             ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:44:34: note: explicitly
defaulted function was implicitly deleted here
                        constexpr cexpr_basic_string& operator=(const
cexpr_basic_string&) = default;                                  ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:238:21: note: copy assignment
operator of 'cexpr_basic_string<char, 128>' is implicitly deleted because field
'nullChar_' is of const-qualified type 'const
ra::cexpr::cexpr_basic_string<char, 128>::value_type' (aka 'const char')
                        const value_type nullChar_ = value_type(0);
                        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:14:3: warning: expression result
unused [-Wunused-value]
                sizeof(string);
                ^~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:112:2: note: in instantiation of
function template specialization 'test_1<unsigned char, 255>' requested here
        test_1<unsigned char, 255>();
        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:15:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::value_type);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:16:3: warning: expression result
unused [-Wunused-value]
```



```
                sizeof(typename string::pointer);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:17:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_pointer);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:18:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::reference);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:19:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_reference);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:20:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::iterator);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:21:3: warning: expression result
unused [-Wunused-value]
                sizeof(typename string::const_iterator);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:26:6: error: object of type
'string' (aka 'cexpr_basic_string<unsigned char, 255UL>') cannot be assigned
because its copy assignment operator is implicitly deleted
        s3 = s;
             ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:44:34: note: explicitly
defaulted function was implicitly deleted here
                        constexpr cexpr_basic_string& operator=(const
cexpr_basic_string&) = default;                                  ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:238:21: note: copy assignment
operator of 'cexpr_basic_string<unsigned char, 255>' is implicitly deleted
because field 'nullChar_' is of const-qualified type 'const
ra::cexpr::cexpr_basic_string<unsigned char, 255>::value_type' (aka 'const
unsigned char')
                        const value_type nullChar_ = value_type(0);
                        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:27:6: error: object of type
'string' (aka 'cexpr_basic_string<unsigned char, 255UL>') cannot be assigned
because its copy assignment operator is implicitly deleted
        s4 = std::move(s3);
             ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:44:34: note: explicitly
defaulted function was implicitly deleted here
                        constexpr cexpr_basic_string& operator=(const
cexpr_basic_string&) = default;                                  ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:238:21: note: copy assignment
operator of 'cexpr_basic_string<unsigned char, 255>' is implicitly deleted
because field 'nullChar_' is of const-qualified type 'const
ra::cexpr::cexpr_basic_string<unsigned char, 255>::value_type' (aka 'const
unsigned char')
                        const value_type nullChar_ = value_type(0);
                        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:14:3: warning: expression result
unused [-Wunused-value]
                sizeof(string);
                ^~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:113:2: note: in instantiation of
function template specialization 'test_1<wchar_t, 64>' requested here
        test_1<wchar_t, 64>();
        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:15:3: warning: expression result
unused [-Wunused-value]
```

```
                  sizeof(typename string::value_type);
                  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:16:3: warning: expression result
unused [-Wunused-value]
                  sizeof(typename string::pointer);
                  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:17:3: warning: expression result
unused [-Wunused-value]
                  sizeof(typename string::const_pointer);
                  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:18:3: warning: expression result
unused [-Wunused-value]
                  sizeof(typename string::reference);
                  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:19:3: warning: expression result
unused [-Wunused-value]
                  sizeof(typename string::const_reference);
                  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:20:3: warning: expression result
unused [-Wunused-value]
                  sizeof(typename string::iterator);
                  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:21:3: warning: expression result
unused [-Wunused-value]
                  sizeof(typename string::const_iterator);
                  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:26:6: error: object of type
'string' (aka 'cexpr_basic_string<wchar_t, 64UL>') cannot be assigned because
its copy assignment operator is implicitly deleted
                  s3 = s;
                     ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:44:34: note: explicitly
defaulted function was implicitly deleted here
                        constexpr cexpr_basic_string& operator=(const
cexpr_basic_string&) = default;
                                                               ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:238:21: note: copy assignment
operator of 'cexpr_basic_string<wchar_t, 64>' is implicitly deleted because
field 'nullChar_' is of const-qualified type 'const
ra::cexpr::cexpr_basic_string<wchar_t, 64>::value_type' (aka 'const wchar_t')
                        const value_type nullChar_ = value_type(0);
                                         ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/app/test_cexpr_basic_string.cpp:27:6: error: object of type
'string' (aka 'cexpr_basic_string<wchar_t, 64UL>') cannot be assigned because
its copy assignment operator is implicitly deleted
                  s4 = std::move(s3);
                     ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:44:34: note: explicitly
defaulted function was implicitly deleted here
                        constexpr cexpr_basic_string& operator=(const
cexpr_basic_string&) = default;
                                                               ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-stri
ng_sane/source/include/ra/cexpr_basic_string.hpp:238:21: note: copy assignment
operator of 'cexpr_basic_string<wchar_t, 64>' is implicitly deleted because
field 'nullChar_' is of const-qualified type 'const
ra::cexpr::cexpr_basic_string<wchar_t, 64>::value_type' (aka 'const wchar_t')
                        const value_type nullChar_ = value_type(0);
                                         ^
32 warnings and 6 errors generated.
gmake[3]: ***
[CMakeFiles/test_cexpr_basic_string.dir/app/test_cexpr_basic_string.cpp.o]
Error 1
gmake[3]: Leaving directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-str
ing_sane/derived'
gmake[2]: *** [CMakeFiles/test_cexpr_basic_string.dir/all] Error 2
gmake[2]: Leaving directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-str
```



```
ing_sane/derived'
gmake[1]: *** [CMakeFiles/test_cexpr_basic_string.dir/rule] Error 2
gmake[1]: Leaving directory
'/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-str
ing_sane/derived'
gmake: *** [test_cexpr_basic_string] Error 2
ERROR: build failed to generate executable test_cexpr_basic_string
```

```
/home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
-S/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-ma
th_sane/source
-B/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-ma
th_sane/derived --check-build-system CMakeFiles/Makefile.cmake 0
/usr/bin/gmake  -f CMakeFiles/Makefile2 test_cexpr_math
gmake[1]: Entering directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-mat
h_sane/derived'
/home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
-S/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-ma
th_sane/source
-B/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-ma
th_sane/derived --check-build-system CMakeFiles/Makefile.cmake 0
/home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
cmake_progress_start
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/derived/CMakeFiles 2
/usr/bin/gmake  -f CMakeFiles/Makefile2 CMakeFiles/test_cexpr_math.dir/all
gmake[2]: Entering directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-mat
h_sane/derived'
/usr/bin/gmake  -f CMakeFiles/test_cexpr_math.dir/build.make
CMakeFiles/test_cexpr_math.dir/depend
gmake[3]: Entering directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-mat
h_sane/derived'
cd
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/derived &&
/home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
cmake_depends "Unix Makefiles"
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/derived
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/derived
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/derived/CMakeFiles/test_cexpr_math.dir/DependInfo.cmake --color=
Scanning dependencies of target test_cexpr_math
gmake[3]: Leaving directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-mat
h_sane/derived'
/usr/bin/gmake  -f CMakeFiles/test_cexpr_math.dir/build.make
CMakeFiles/test_cexpr_math.dir/build
gmake[3]: Entering directory
`/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-mat
h_sane/derived'
[ 50%] Building CXX object
CMakeFiles/test_cexpr_math.dir/app/test_cexpr_math.cpp.o
/home/frodo/public/ugls_lab-4.0.70/bin/clang++
-I/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-ma
th_sane/source/include  -pedantic-errors -std=gnu++17 -o
CMakeFiles/test_cexpr_math.dir/app/test_cexpr_math.cpp.o -c
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/gcc/lib64: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/clang/lib: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/boost/lib: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/catch/lib64: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/CGAL/lib64: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/jasper/lib64: 'linker'
input unused [-Wunused-command-line-argument]
clang-10: warning:
-Wl,-rpath,/home/frodo/public/ugls_lab-4.0.70/packages/clang/lib64: 'linker'
input unused [-Wunused-command-line-argument]
```



```
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/gcc/lib64'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/clang/lib'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/boost/lib'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/catch/lib64'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/CGAL/lib64'
[-Wunused-command-line-argument]
clang-10: warning: argument unused during compilation:
'-L/home/frodo/public/ugls_lab-4.0.70/packages/jasper/lib64'
[-Wunused-command-line-argument]
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:14:17: error: constexpr variable 'x6' must
be initialized by a constant expression
        constexpr auto x6 = rcm::sin(T(5.0));
                    ^   ~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:22:2: note: in instantiation of function
template specialization 'test<float>' requested here
        test<float>();
        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:41:3: note: constexpr evaluation hit
maximum step limit; possible infinite loop?
                if(y == 0){
                ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:55:17: note: in call to
'mod(3.484586e-07, 6.283185e+00)'
                T reduced_x = mod<T>(x, 2*pi<T>);
                              ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.484586e-07)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                              ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.045376e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.136127e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:17: note: in call to
'sin(9.408382e-06)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                          ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: (skipping 7 calls in
backtrace; use -fconstexpr-backtrace-limit=0 to see all)
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                              ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(6.172839e-02)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.851852e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(5.555555e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.666667e+00)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:14:22: note: in call to 'sin(5.000000e+00)'
        constexpr auto x6 = rcm::sin(T(5.0));
                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:16:17: error: constexpr variable 'x8' must
```

```
be initialized by a constant expression
        constexpr auto x8 = rcm::tan(T(5.0));
                      ^    ~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:38:27: note: constexpr evaluation hit
maximum step limit; possible infinite loop?
        constexpr T mod(T x, T y){
                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:55:17: note: in call to
'mod(3.484586e-07, 6.283185e+00)'
                T reduced_x = mod<T>(x, 2*pi<T>);
                              ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.484586e-07)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.045376e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.136127e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:17: note: in call to
'sin(9.408382e-06)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                          ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: (skipping 8 calls in
backtrace; use -fconstexpr-backtrace-limit=0 to see all)
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.851852e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(5.555555e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.666667e+00)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:89:9: note: in call to
'sin(5.000000e+00)'
                T a = sin<T>(x);
                      ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:16:22: note: in call to 'tan(5.000000e+00)'
        constexpr auto x8 = rcm::tan(T(5.0));
                      ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:14:17: error: constexpr variable 'x6' must
be initialized by a constant expression
        constexpr auto x6 = rcm::sin(T(5.0));
                      ^    ~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:23:2: note: in instantiation of function
template specialization 'test<double>' requested here
        test<double>();
        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:41:3: note: constexpr evaluation hit
maximum step limit; possible infinite loop?
                if(y == 0){
                ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:55:17: note: in call to
'mod(3.484586e-07, 6.283185e+00)'
                T reduced_x = mod<T>(x, 2*pi<T>);
                              ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.484586e-07)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
```



```
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.045376e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.136127e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:17: note: in call to
'sin(9.408382e-06)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                          ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: (skipping 7 calls in
backtrace; use -fconstexpr-backtrace-limit=0 to see all)
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(6.172840e-02)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.851852e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(5.555556e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.666667e+00)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:14:22: note: in call to 'sin(5.000000e+00)'
        constexpr auto x6 = rcm::sin(T(5.0));
                      ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:16:17: error: constexpr variable 'x8' must
be initialized by a constant expression
        constexpr auto x8 = rcm::tan(T(5.0));
                      ^    ~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:38:27: note: constexpr evaluation hit
maximum step limit; possible infinite loop?
        constexpr T mod(T x, T y){
                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:55:17: note: in call to
'mod(3.484586e-07, 6.283185e+00)'
                T reduced_x = mod<T>(x, 2*pi<T>);
                              ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.484586e-07)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.045376e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.136127e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:17: note: in call to
'sin(9.408382e-06)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                          ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: (skipping 8 calls in
backtrace; use -fconstexpr-backtrace-limit=0 to see all)
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.851852e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(5.555556e-01)'
```

```
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.666667e+00)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:89:9: note: in call to
'sin(5.000000e+00)'
                T a = sin<T>(x);
                      ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:16:22: note: in call to 'tan(5.000000e+00)'
        constexpr auto x8 = rcm::tan(T(5.0));
                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:14:17: error: constexpr variable 'x6' must
be initialized by a constant expression
        constexpr auto x6 = rcm::sin(T(5.0));
                    ^     ~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:24:2: note: in instantiation of function
template specialization 'test<long double>' requested here
        test<long double>();
        ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:41:3: note: constexpr evaluation hit
maximum step limit; possible infinite loop?
                if(y == 0){
                ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:55:17: note: in call to
'mod(3.484586e-07, 6.283185e+00)'
                T reduced_x = mod<T>(x, 2*pi<T>);
                              ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.484586e-07)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.045376e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.136127e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:17: note: in call to
'sin(9.408382e-06)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                  ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: (skipping 7 calls in
backtrace; use -fconstexpr-backtrace-limit=0 to see all)
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(6.172840e-02)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.851852e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(5.555556e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.666667e+00)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:14:22: note: in call to 'sin(5.000000e+00)'
        constexpr auto x6 = rcm::sin(T(5.0));
                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:16:17: error: constexpr variable 'x8' must
be initialized by a constant expression
        constexpr auto x8 = rcm::tan(T(5.0));
                    ^     ~~~~~~~~~~~~~~~~
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:38:27: note: constexpr evaluation hit
maximum step limit; possible infinite loop?
```



```
        constexpr T mod(T x, T y){
                    ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:55:17: note: in call to
'mod(3.484586e-07, 6.283185e+00)'
                T reduced_x = mod<T>(x, 2*pi<T>);
                              ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.484586e-07)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.045376e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(3.136127e-06)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:17: note: in call to
'sin(9.408382e-06)'
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                  ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: (skipping 8 calls in
backtrace; use -fconstexpr-backtrace-limit=0 to see all)
                                result = (3*sin(reduced_x/3)) -
(4*cube(sin(reduced_x/3)));
                                                            ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.851852e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(5.555556e-01)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:65:45: note: in call to
'sin(1.666667e+00)'
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/include/ra/cexpr_math.hpp:89:9: note: in call to
'sin(5.000000e+00)'
                T a = sin<T>(x);
                      ^
/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-math
_sane/source/app/test_cexpr_math.cpp:16:22: note: in call to 'tan(5.000000e+00)'
        constexpr auto x8 = rcm::tan(T(5.0));
                            ^
6 errors generated.
gmake[3]: *** [CMakeFiles/test_cexpr_math.dir/app/test_cexpr_math.cpp.o] Error 1
gmake[3]: Leaving directory
'/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-mat
h_sane/derived'
gmake[2]: *** [CMakeFiles/test_cexpr_math.dir/all] Error 2
gmake[2]: Leaving directory
'/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-mat
h_sane/derived'
gmake[1]: *** [CMakeFiles/test_cexpr_math.dir/rule] Error 2
gmake[1]: Leaving directory
'/tmp/assignment_precheck-judeonyia@ugls6.ece.uvic.ca-15563-01WM6NQc/package-mat
h_sane/derived'
gmake: *** [test_cexpr_math] Error 2
ERROR: build failed to generate executable test_cexpr_math
```

```
commit ddd44b41088d6da6abc5b4dde63be21adcc8a2bf
Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
Date:   Fri May 29 00:55:23 2020 -0700

    My First Commit

commit 8b4a82ae0c0c63ccfa2fb34e288cd469e1e087bd
Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
Date:   Fri May 29 01:55:51 2020 -0700

    Set up the structure of cexpr_basic_string class

commit fc4203a5fe1b1fc4d243d6f8972dd860378d44a0
Author: Jude Onyia <judeonyia@ugls19.ece.uvic.ca>
Date:   Fri May 29 02:12:34 2020 -0700

    Checking Folder from assignment precheck

commit f5e4a01ac9557e6b06c7349dbc5f7e196efdff4b
Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
Date:   Fri May 29 02:16:50 2020 -0700

    Removed Assign Precheck Folder

commit bfbe276218a4abd78411a7355926d1c7dd748b5d
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Fri May 29 17:01:36 2020 -0700

    1) Wrote the types and data members of cexpr_basic_string
    2) Wrote the default constructor
    3) Wrote the constructor with parameter of string pointer
    4) Tested

commit f82e1dc817588636e0b5a8d951a623cba0fcbe66
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Sat May 30 00:53:50 2020 -0700

    1) Wrote max size, size and capacity functions
    2) Wrote data, begin and end functions
    3) Wrote operator[] overload

commit 06a99e4cf21d3984ffe77a80ef06e08b3a5fa7ee
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Sat May 30 00:57:12 2020 -0700

    Removed swp files

commit 5fe880a9fb8917e64109972b981a36c701e1a6f0
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Sat May 30 18:49:33 2020 -0700

    1) Wrote push_back() and pop_back()
    2) Wrote  append(pointer), append(obj)
    3) Wrote clear()

commit 3dfb8efadaaa7367243c3db6c29ace0a4efccf77
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Sun May 31 00:27:48 2020 -0700

    1) Wrote Alias for cexpr_basic_string called cexpr_string
    2) Wrote to_string helper function

```

```
commit ee65da40a2cec5787b274eac9e9539a84f718176
Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
Date:   Sun May 31 01:26:13 2020 -0700

    Set up for Mandelbrot section

commit b3b55701af97708b386a6eb79a54d77292b53919
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Sun May 31 21:07:54 2020 -0700

    1) Wrote basic Mandelbrot hpp and cpp
    2) Modified to_string function to to accomodate clang compiler

commit 1db2c71fe13c812a0f718986d289ea88d06be7ec
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Sun May 31 21:26:38 2020 -0700

    1) Printed the mandelbrot to a file called mandelbrot.pnm
    2) Displayed file on image software. Confirming it is mandelbrot

commit cf9bcf5545031c9845cbb379ec2911d3bb1b814c
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Mon Jun 1 01:00:08 2020 -0700

    1) Wrote template variable pi
    2) Wrote template functions abs,sqr, cube and mod

commit e52269371567c4f87e4a75c1ea4650dacde9eba3
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Mon Jun 1 02:33:02 2020 -0700

    1) Wrote template functions for sin, cos, tan and sqrt

commit d450669cb83518f16fde276b25d96f3baf63570c
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Mon Jun 1 14:44:45 2020 -0700

    1) Wrote biquad_filter_coefs class
    2) Wrote constexpr constructor to initialize parameters
    3) Wrote constexpr constructor to copy from another obj

commit 9873843310751781db8f11e9558bb79da4a7f42f
Author: Jude Onyia <judeonyia10@gmail.com>
Date:   Mon Jun 1 17:32:15 2020 -0700

    1) Wrote template functions for lowpass, highpass and bandpass filters
    2) Wrote template functions for low shelf boost and low shelf cut filters
```

Name: Jude Onyia

Student ID: V00947095

Course: ECE 596C

Due Date: June 5, 2020

Assignment 2: Non – Programming Exercise

8.1)

x?	lvalue
static_cast<std::vector<int>&&>(x)?	Rvalue (xvalue)
x.begin()?	Rvalue (xvalue)
++i?	lvalue
*i?	lvalue
*i += 5?	lvalue
x[0]?	lvalue
++a?	lvalue
a++?	Rvalue (prvalue)
func1(x)?	Rvalue (xvalue)
y = func1(x)?	lvalue

8.24)

Widget b(a);	Copy Constructor	Required with certainty (cannot be elided)
Widget c = a;	Copy Constructor	Required with certainty (cannot be elided)
Widget d(std::move(c));	Move Constructor	Required with certainty (cannot be elided)
Widget e = std::move(d);	Move Constructor	Required with certainty (cannot be elided)
Widget f(make_widget_1());	Move Constructor	Guaranteed to be elided
Widget g(make_widget_2(true));	Move Constructor	May be required (depending on if it can be elided)
c = a;	Copy Assignment	Required with certainty (cannot be elided)
b = std::move(c);	Move Assignment	Required with certainty (cannot be elided)
a = make_widget_1();	Move Assignment	Guaranteed to be elided
a = make_widget_2(true);	Move Assignment	May be required (depending on if it can be elided)
func_1(a);	Copy Constructor	Required with certainty (cannot be elided)
func_1(std::move(a));	Move Constructor	Required with certainty (cannot be elided)
func_1(make_widget_1());	Move Constructor	Guaranteed to be elided
func_2(std::move(b));	Move Constructor	Required with certainty (cannot be elided)

8.25)

Lines marked with ???:	Line of code where a temporary object is created:	Description and explanation:
Line 67: z = x + y;	*Line 53:* counter(x)	This temporary object is created within operator+. It is needed to store the result of x+=y without changing the value of x.
	Line 67: x+y	This temporary object is created in the main function. It is required in order to hold the return of operator+ before it can be assigned to counter z.
Line 68: z = z + z;	*Line 53:* counter(x)	This temporary object is created within operator+. It is needed to store the result of x+=y without changing the value of x.
	Line 68: z+z	This temporary object is created in the main function. It is required in order to hold the return of operator+ before it can be assigned to counter z.
Line 69: y = ++z;	N/A	This line does not need a temporary object because the pre-fixed operator++ returns an lvalue reference to the counter z.
Line 70: z = y++;	*Line 70:* y++	This temporary object is created in the main function. It is required in order to hold the return value of operator++(int) before it can be assigned to counter z.
Line 71: x = z;	N/A	This line does not need a temporary object because the operator= returns an lvalue reference.

8.26)

Some advantages of array-based implementation of a stack are elements stored contiguously in memory, less overhead, and it is more cache friendly. The disadvantage of array-based implementation is it cannot guarantee that each push will take constant time. This is due to the situation where the capacity of the array is full, in this case it would have to copy its entire content to a bigger section of memory before it can push that object.

An advantage of node-based implementation of a stack is that the previously discussed capacity exceeding problem is not evident here. This guarantees the push operation will be done in constant time. Another advantage is that references to objects in the stack are stable (always valid) since the objects they refer to are not forced to be copied or moved somewhere else. Disadvantages of node-based implementation include not storing elements contiguously in memory, per-element overhead, and requires more space than array-based implementation due to the overhead.

```
# Specify Minimum Required Version
cmake_minimum_required(VERSION 3.1 FATAL_ERROR)

# Specify Project and Language
project(cpp_compile_time LANGUAGES CXX)

# For Clang:
set(EXTRA_COMPILE_FLAGS "-fconstexpr-steps=100000000")
set_source_files_properties(app/test_mandelbrot.cpp PROPERTIES COMPILE_FLAGS ${E
XTRA_COMPILE_FLAGS})

# Set Include Directory
include_directories(include)

# Add Executable Program
add_executable(test_cexpr_basic_string app/test_cexpr_basic_string.cpp)
add_executable(test_mandelbrot app/test_mandelbrot.cpp)
add_executable(test_cexpr_math app/test_cexpr_math.cpp)
add_executable(test_biquad_filter app/test_biquad_filter.cpp)
```

```
1  #ifndef CEXPR
2  #define CEXPR
3  #include <cstddef>
4  #include <stdexcept>
5  #include <string>
6  namespace ra::cexpr{
7      // A basic string class template for use in constexpr contexts
8      template<class T, std::size_t M>
9      class cexpr_basic_string{
10         public:
11             // An unsigned integral type used to represent sizes
12             using size_type = std::size_t;
13
14             // The type of each character in the string (i.e., an alias for
15             // the template parameter T)
16             using value_type = T;
17
18             // The type of a mutating pointer to each character in the string
19             using pointer = T*;
20
21             // The type of a non-mutating pointer to each character in the
22             // string
23             using const_pointer = const T*;
24
25             // The type of a mutating reference to a character in the string
26             using reference = T&;
27
28             // The type of a non-mutating reference to a character in the
29             // string
30             using const_reference = const T&;
31
32             // A mutating iterator type for the elements in the string
33             using iterator = pointer;
34
35             // A non-mutating iterator type for the elements in the string
36             using const_iterator = const_pointer;
37
38             // Creates an empty string (i.e., a string containing no
39             // characters)
40             constexpr cexpr_basic_string() : charArray_{0}, charSize_(0) {}
41
42             // Explicitly default some special members
43             constexpr cexpr_basic_string(const cexpr_basic_string&) = default;
44             constexpr cexpr_basic_string& operator=(const cexpr_basic_string&) =
    default;
45             ~cexpr_basic_string() = default;
46
47             // Creates a string with the contents given by the
48             // null-terminated character array pointed to by s
49             // If the string does not have sufficient capacity to hold
50             // the character data provided, an exception of type
51             // std::runtime_error is thrown
52             constexpr cexpr_basic_string(const value_type* s) : charArray_{0}, c
   harSize_(0){
53                 while(s[charSize_] != nullChar_){
54                     if(M <= charSize_){
55                         clear();
56                         throw std::runtime_error("String does not have sufficient capacity");

57                     }
58                     else{
59                         charArray_[charSize_] = s[charSize_];
```

```
60                          ++charSize_;
61                      }
62                  }
63              }
64
65              // Creates a string with the contents specified by the characters
66              // in the iterator range [first, last).
67              // If the string does not have sufficient capacity to hold
68              // the character data provided, an exception of type
69              // std::runtime_error is thrown.
70              constexpr cexpr_basic_string(const_iterator first, const_iterator la
    st) : charArray_{0}, charSize_(0){
71                  if(M < (last - first)){
72                      throw std::runtime_error("String does not have sufficient capacity");
73                  }
74                  else{
75                      for(const_iterator i=first; i < last; ++i){
76                          charArray_[charSize_] = *i;
77                          ++charSize_;
78                      }
79                  }
80              }
81
82              // Returns the maximum number of characters that can be held by a
83              // string of this type.
84              // The value returned is the template parameter M
85              static constexpr size_type max_size() { return M; }
86
87              // Returns the maximum number of characters that the string can
88              // hold. The value returned is always the template parameter M
89              constexpr size_type capacity() const { return M; }
90
91              // Returns the number of characters in the string (excluding the
92              // dummy null character)
93              constexpr size_type size() const { return charSize_; }
94
95              // Returns a pointer to the first character in the string
96              // The pointer that is returned is guaranteed to point to a
97              // null-terminated character array
98              // The user of this class shall not alter the dummy null
99              // character stored at data() + size().
100             value_type* data(){
101                 return charArray_;
102             }
103             const value_type* data() const {
104                 return charArray_;
105             }
106
107             // Returns an iterator referring to the first character in the
108             // string
109             constexpr iterator begin() {
110                 return charArray_;
111             }
112             constexpr const_iterator begin() const {
113                 return charArray_;
114             }
115
116             // Returns an iterator referring to the fictitious
117             // one-past-the-end character in the string
118             constexpr iterator end(){
119                 return (&charArray_[charSize_]);
120             }
```

```
121             constexpr const_iterator end() const {
122                 return (&charArray_[charSize_]);
123             }
124
125             // Returns a reference to the i-th character in the string if i
126             // is less than the string size; and returns a reference to the
127             // dummy null character if i equals the string size.
128             // Precondition: The index i is such that i >= 0 and i <= size().
129             constexpr reference operator[](size_type i){
130                 if(i<0 || i>charSize_){
131                     throw std::domain_error("invalid array access");
132                 }
133                 else if(i==charSize_){ return charArray_[charSize_]; }
134                 else{ return charArray_[i];}
135             }
136             constexpr const_reference operator[](size_type i)const{
137                 if(i<0 || i>charSize_){
138                     throw std::domain_error("invalid array access");
139                 }
140                 else if(i==charSize_){ return nullChar_; }
141                 else{ return charArray_[i];}
142             }
143
144             // Appends (i.e., adds to the end) a single character to the
145             // string. If the size of the string is equal to the capacity,
146             // the string is not modified and an exception of type
147             // std::runtime_error is thrown.
148             constexpr void push_back(const T& x){
149                 if(M <= charSize_){
150                     throw std::runtime_error("String does not have sufficient capacity");
151                 }
152                 else{
153                     charArray_[charSize_] = x;
154                     ++charSize_;
155                 }
156             }
157
158             // Erases the last character in the string.
159             // If the string is empty, an exception of type std::runtime_error
160             // is thrown.
161             constexpr void pop_back(){
162                 if(charSize_ == 0){
163                     throw std::runtime_error("String does not have sufficient capacity");
164                 }
165                 else{
166                     charArray_[charSize_ - 1] = '\0';
167                     --charSize_;
168                 }
169             }
170
171             // Appends (i.e., adds to the end) to the string the
172             // null-terminated string pointed to by s.
173             // Precondition: The pointer s must be non-null.
174             // If the string has insufficient capacity to hold the new value
175             // resulting from the append operation, the string is not modified
176             // and an exception of type std::runtime_error is thrown.
177             constexpr cexpr_basic_string& append(const value_type* s){
178                 size_type addToSize = 0;
179                 if(s == nullptr) {
180                     throw std::runtime_error("Null pointer");
181                 }
182                 else{
```

```
183                         while(s[addToSize] != nullChar_){
184                             if(M <= (charSize_+addToSize)){
185                                 charSize_ += addToSize;
186                                 for(size_type i=0; i<addToSize; ++i){
187                                     pop_back();
188                                 }
189                                 addToSize = 0;
190                                 throw std::runtime_error("String does not have sufficient capacit
    y");
191                                 break;
192                             }
193                             else{
194                                 charArray_[charSize_ + addToSize] = s[addToSize];
195                                 ++addToSize;
196                             }
197                         }
198                         charSize_ += addToSize;
199                     }
200                     return *this;
201                 }
202 
203                 // Appends (i.e., adds to the end) to the string another
204                 // cexpr_basic_string with the same character type (but
205                 // possibly a different maximum size).
206                 // If the string has insufficient capacity to hold the new value
207                 // resulting from the append operation, the string is not modified
208                 // and an exception of type std::runtime_error is thrown.
209                 template<size_type OtherM>
210                 constexpr cexpr_basic_string& append(const cexpr_basic_string<value_
    type, OtherM>& other){
211                     size_type addToSize = 0;
212                     if(M < (charSize_ + other.size())){
213                         throw std::runtime_error("String does not have sufficient capacity");
214                     }
215                     else{
216                         for(size_type i=0; i<other.size(); ++i){
217                             charArray_[charSize_ + i] = other[i];
218                             ++addToSize;
219                         }
220                         charSize_ += addToSize;
221 
222                     }
223                     return *this;
224                 }
225 
226                 // Erases all of the characters in the string, yielding an empty
227                 // string.
228                 constexpr void clear(){
229                     for(size_type i=0; i<charSize_; ++i){
230                         charArray_[i] = '\0';
231                     }
232                     charSize_ = 0;
233                 }
234 
235             private:
236                 value_type charArray_[M+1];
237                 size_type charSize_;
238                 const value_type nullChar_ = value_type(0);
239         };
240 
241         template<std::size_t M>
242         using cexpr_string = cexpr_basic_string<char, M>;
```

```
243
244     constexpr char digit_to_char(std::size_t num){
245         char mychar = '0';
246         switch(num) {
247             case std::size_t(0) : mychar = '0'; break;
248             case std::size_t(1) : mychar = '1'; break;
249             case std::size_t(2) : mychar = '2'; break;
250             case std::size_t(3) : mychar = '3'; break;
251             case std::size_t(4) : mychar = '4'; break;
252             case std::size_t(5) : mychar = '5'; break;
253             case std::size_t(6) : mychar = '6'; break;
254             case std::size_t(7) : mychar = '7'; break;
255             case std::size_t(8) : mychar = '8'; break;
256             case std::size_t(9) : mychar = '9'; break;
257             default: mychar = '0';
258         }
259         return mychar;
260     }
261
262     constexpr std::size_t to_string(std::size_t n, char* buffer,std::size_t size
    , char** end){
263         std::size_t ite = 0;
264         char temp_buf = 0;
265         //char theArray[size_const] = {0};
266         //char theArray_backwards[size] = {0};
267         while(n!=0){
268             buffer[ite] = digit_to_char(n % std::size_t(10));
269             n = n / std::size_t(10);
270             ++ite;
271             if(ite > size){
272                 throw std::runtime_error("String does not have sufficient capacity");
273                 break;
274             }
275         }
276         if(ite <= size){
277             for(std::size_t i=0; i<(ite/2); ++i){
278                 temp_buf = buffer[i];
279                 buffer[i] = buffer[ite-1-i];
280                 buffer[ite-1-i] = temp_buf;
281             }
282             //buffer = theArray;
283             if(end != nullptr){
284                 *end = &buffer[ite];
285             }
286         }
287         return ite;
288     }
289     /*constexpr std::size_t to_string(std::size_t n, char* buffer, std::size_t s
    ize, char** end){
290         const size_t size_const = size;
291         cexpr_string<size_const> obj;
292         std::string n_str = std::to_string(n);
293         if(n_str.size() > size_const){
294             throw std::runtime_error("String does not have sufficient capacity")
    ;
295         }
296         else{
297             for(size_t i=0; i<n_str.size(); ++i){
298                 obj.push_back(n_str[i]);
299             }
300             buffer = obj.data();
301             if(end != nullptr){
```

```
302                     *end = obj.data() + obj.size();
303                 }
304             }
305             return (obj.size());
306         }*/
307 }
308 #endif
```

```
1  #ifndef Mymandelbrot
2  #define Mymandelbrot
3  #include "ra/cexpr_basic_string.hpp"
4  #include <cstddef>
5
6  namespace ra::fractal {
7      // Function to compute mandelbrot
8      constexpr ra::cexpr::cexpr_string<1100000> mandelbrotComputation(const std::
   size_t W, const std::size_t H){
9          ra::cexpr::cexpr_string<1100000> obj;
10         char W_str[10] = {0};
11         std::size_t W_str_size = 0;
12         char H_str[10] = {0};
13         std::size_t H_str_size = 0;
14
15         double c0 = 0.0;
16         double c1 = 0.0;
17         double zn_c0 = 0.0;
18         double zn_c0_next = 0.0;
19         double zn_c1 = 0.0;
20         double zn_c1_next = 0.0;
21         std::size_t n = 0;
22         bool outOfBound = false;
23
24         double a0 = -1.6;
25         double a1 = -1.1;
26         double b0 = 0.6;
27         double b1 = 1.1;
28         double mult_c0 = (b0-a0)/(W-1);
29         double mult_c1 = (b1-a1)/(H-1);
30
31         obj.push_back('P');
32         obj.push_back('1');
33         obj.push_back(' ');
34         W_str_size = ra::cexpr::to_string(W,W_str,10,nullptr);
35         for(std::size_t i=0; i<W_str_size; ++i){
36             obj.push_back(W_str[i]);
37         }
38         obj.push_back(' ');
39         H_str_size = ra::cexpr::to_string(H,H_str,10,nullptr);
40         for(std::size_t i=0; i<H_str_size; ++i){
41             obj.push_back(H_str[i]);
42         }
43         obj.push_back('\n');
44
45
46         for(std::size_t l=0; l<H; ++l){
47             c1 = a1 + ( (double)(H-1-l) * mult_c1 );
48             for(std::size_t k=0; k<W; ++k){
49                 c0 = a0 + ( (double)k * mult_c0 );
50                 while(n < 16){
51                     zn_c0_next = (zn_c0*zn_c0)-(zn_c1*zn_c1) + c0;
52                     zn_c1_next = (2.0*zn_c0*zn_c1) + c1;
53                     zn_c0 = zn_c0_next;
54                     zn_c1 = zn_c1_next;
55                     if( ((zn_c0*zn_c0)+(zn_c1*zn_c1)) > 4.0 ){
56                         outOfBound = true;
57                         break;
58                     }
59                     ++n;
60                 }
61                 if(outOfBound == false){
```

```
                    obj.push_back('1');
                }
                else{
                    obj.push_back('0');
                }
                outOfBound = false;
                n = 0;
                zn_c0 = 0.0;
                zn_c1 = 0.0;
            }
            obj.push_back('\n');
        }

        //obj.push_back('P');
        //obj.push_back('1');
        //obj.push_back(' ');
        //W_str_size = ra::cexpr::to_string(W,W_str,10,nullptr);
        //for(std::size_t i=0; i<W_str_size; ++i){
            //obj.push_back(W_str[i]);
        //}
        //obj.push_back(' ');
        //H_str_size = ra::cexpr::to_string(H,H_str,10,nullptr);
        //for(std::size_t i=0; i<H_str_size; ++i){
            //obj.push_back(H_str[i]);
        //}
        //obj.push_back('\n');

        return obj;

    }

    // A variable template for a string that represents an image depicting
    // the Mandelbrot set. The image has width W and height H.
    // This object must be of type cexpr_string<M> for some appropriate M.
    // The string is a binary image encoded in the text-based bitmap PNM
    // format.
    // The values of W and H must be such that W >= 2 and H >= 2.
    template<std::size_t W, std::size_t H>
    constexpr auto mandelbrot = mandelbrotComputation(W,H);
}
#endif
```

```
1  #ifndef CexprMath
2  #define CexprMath
3  #include<boost/math/constants/constants.hpp>
4
5  namespace ra::cexpr_math {
6      // The math constant pi.
7      // The type T is a floating-point type.
8      template<class T>
9      constexpr T pi = boost::math::constants::pi<T>();
10
11     // Returns the absolute value of x.
12     // The type T is a floating-point type.
13     template<class T>
14     constexpr T abs(T x){
15         return (x < 0)? (-x) : x;
16     }
17
18     // Returns the square of x.
19     // The type T is a floating-point type.
20     template<class T>
21     constexpr T sqr(T x){
22         return (x * x);
23     }
24
25     // Returns the cube of x.
26     // The type T is a floating-point type.
27     template<class T>
28     constexpr T cube(T x){
29         return (x * x * x);
30     }
31
32     // Returns the remainder after division when x is divided by y.
33     // In particular, x - n y is returned where n is the result obtained by
34     // dividing x by y and then rounding (to an integer value) toward zero.
35     // If y is zero, an exception of type std::overflow_error is thrown.
36     // The type T is a floating-point type.
37     template<class T>
38     constexpr T mod(T x, T y){
39         long long n = 0;
40         T remainder = 0;
41         if(y == 0){
42             throw std::overflow_error("zero modulus");
43         }
44         n = (long long)(x / y);
45         remainder = x - ( ((T)n) * y );
46         return remainder;
47     }
48
49     // Returns the sine of x.
50     // Note that a particular algorithm must be used to implement this
51     // function.
52     // The type T is a floating-point type.
53     template<class T>
54     constexpr T sin(T x){
55         T reduced_x = mod<T>(x, 2*pi<T>);
56         T result = 0;
57         if(reduced_x < 0){
58             result = -1 * sin(abs(reduced_x));
59         }
60         else{
61             if(x <= 0.000001){
62                 result = x;
```

```
63              }
64              else{
65                  result = (3*sin(reduced_x/3)) - (4*cube(sin(reduced_x/3)));
66              }
67          }
68          return result;
69      }
70
71      // Returns the cosine of x.
72      // Note that a particular algorithm must be used to implement this
73      // function.
74      // The type T is a floating-point type.
75      template<class T>
76      constexpr T cos(T x){
77          return (sin<T>( x + (pi<T>/2) ));
78      }
79
80      // Returns the tangent of x.
81      // Note that a particular algorithm must be used to implement this
82      // function.
83      // If the tangent of x is infinite, an exception of type
84      // std::overflow_error is thrown.
85      // The type T is a floating-point type.
86      template<class T>
87      constexpr T tan(T x){
88          T result = 0;
89          T a = sin<T>(x);
90          T b = cos<T>(x);
91          if(b == 0){
92              throw std::overflow_error("infinite tangent");
93          }
94          else{
95              result = a / b;
96          }
97          return result;
98      }
99
100     // Returns the square root of x.
101     // If x is negative, an exception of type std::domain_error is thrown.
102     // Note that a particular algorithm must be used to implement this
103     // function.
104     // The type T is a floating-point type.
105     template<class T>
106     constexpr T sqrt(T x){
107         T ep = std::numeric_limits<T>::epsilon();
108         T result = x;
109         T result_next = x;
110         if(x < 0){
111             throw std::domain_error("square root of negative number");
112         }
113         else{
114             do{
115                 result = result_next;
116                 result_next = result - ( (sqr<T>(result)-x) / (2*result) );
117             } while(abs<T>(result_next - result) > ep);
118         }
119         return result;
120     }
121 }
122 #endif
```

```
#ifndef BiquadFilter
#define BiquadFilter
#include"ra/cexpr_math.hpp"

namespace ra::biquad {
    // Biquad filter coefficients class.
    template<class Real>
    struct biquad_filter_coefs{
        // The real number type used to represent the filter coefficients.
        using real = Real;

        // Creates a set of filter coefficients where the coefficients
        // a0, a1, a2, b0, b1, and b2 are initialized to a0_, a1_, a2_,
        // b0_, b1_, and b2_, respectively.
        constexpr biquad_filter_coefs(real a0_, real a1_, real a2_, real b0_, re
al b1_, real b2_) : a0(a0_), a1(a1_), a2(a2_), b0(b0_), b1(b1_), b2(b2_){}

        // Creates a set of filter coefficients by copying from another set.
        // Since Real and OtherReal need not be the same, this constructor
        // can be used to convert between filter coefficients of different
        // types.
        template<class OtherReal>
        constexpr biquad_filter_coefs(const biquad_filter_coefs<OtherReal>& coef
s) : a0(coefs.a0), a1(coefs.a1), a2(coefs.a2), b0(coefs.b0), b1(coefs.b1), b2(co
efs.b2){}

        // The filter coefficients a0, a1, a2, b0, b1, and b2.
        real a0;
        real a1;
        real a2;
        real b0;
        real b1;
        real b2;
    };

    // Returns the coefficients for a biquad lowpass filter with normalized
    // cutoff frequency f and Q factor q, where f in [0,1] and q > 0.
    // The filter coefficients are always normalized such that the
    // coefficient b0 is 1.
    // The type Real is a floating-point type.
    // All real arithmetic should be performed with the Real type.
    template<class Real>
    constexpr biquad_filter_coefs<Real> lowpass(Real f, Real q){
        Real pi_ = ra::cexpr_math::pi<Real>;
        Real sig_ = ra::cexpr_math::tan<Real>(f*pi_/2.0);
        Real a0 = ra::cexpr_math::sqr<Real>(sig_);
        Real a1 = 2.0 * a0;
        Real a2 = a0;
        Real b0 = a0 + (sig_ / q) + 1.0;
        Real b1 = 2.0 * (a0 - 1.0);
        Real b2 = a0 - (sig_ / q) + 1.0;

        a0=a0/b0; a1=a1/b0; a2=a2/b0; b1=b1/b0; b2=b2/b0;
        biquad_filter_coefs<Real> obj(a0,a1,a2,1.0,b1,b2);
        return obj;
    }

    // Returns the coefficients for a biquad highpass filter with
    // normalized cutoff frequency f and Q factor q, where f in [0,1]
    // and q > 0.
    // The filter coefficients are always normalized such that the
    // coefficient b0 is 1.
```

```
60      // The type Real is a floating-point type.
61      // All real arithmetic should be performed with the Real type.
62      template<class Real>
63      constexpr biquad_filter_coefs<Real> highpass(Real f, Real q){
64          Real pi_ = ra::cexpr_math::pi<Real>;
65          Real sig_ = ra::cexpr_math::tan<Real>(f*pi_/2.0);
66          Real sig_sqr = ra::cexpr_math::sqr<Real>(sig_);
67          Real a0 = 1.0;
68          Real a1 = -2.0;
69          Real a2 = 1.0;
70          Real b0 = sig_sqr + (sig_/q) + 1.0;
71          Real b1 = 2.0 * (sig_sqr - 1.0);
72          Real b2 = sig_sqr - (sig_/q) + 1.0;
73
74          a0=a0/b0; a1=a1/b0; a2=a2/b0; b1=b1/b0; b2=b2/b0;
75          biquad_filter_coefs<Real> obj(a0,a1,a2,1.0,b1,b2);
76          return obj;
77      }
78
79      // Returns the coefficients for a biquad bandpass filter with
80      // normalized cutoff frequency f and Q factor q, where f in [0,1]
81      // and q > 0.
82      // The filter coefficients are always normalized such that the
83      // coefficient b0 is 1.
84      // The type Real is a floating-point type.
85      // All real arithmetic should be performed with the Real type.
86      template<class Real>
87      constexpr biquad_filter_coefs<Real> bandpass(Real f, Real q){
88          Real pi_ = ra::cexpr_math::pi<Real>;
89          Real sig_ = ra::cexpr_math::tan<Real>(f*pi_/2.0);
90          Real sig_sqr = ra::cexpr_math::sqr<Real>(sig_);
91          Real a0 = sig_ / q;
92          Real a1 = 0.0;
93          Real a2 = -a0;
94          Real b0 = sig_sqr + (sig_/q) + 1.0;
95          Real b1 = 2.0 * (sig_sqr - 1.0);
96          Real b2 = sig_sqr - (sig_/q) + 1.0;
97
98          a0=a0/b0; a1=a1/b0; a2=a2/b0; b1=b1/b0; b2=b2/b0;
99          biquad_filter_coefs<Real> obj(a0,a1,a2,1.0,b1,b2);
100         return obj;
101     }
102
103     // Returns the coefficients for a biquad low-frequency shelving
104     // boost filter with normalized cutoff frequency f and gain-control
105     // parameter a, where f in [0,1] and a >= 0.
106     // For a gain in dB of G (where G > 0), choose a = 10 ^ (G / 20).
107     // The filter coefficients are always normalized such that the
108     // coefficient b0 is 1.
109     // The type Real is a floating-point type.
110     // All real arithmetic should be performed with the Real type.
111     template<class Real>
112     constexpr biquad_filter_coefs<Real> lowshelf_boost(Real f, Real a){
113         Real pi_ = ra::cexpr_math::pi<Real>;
114         Real sig_ = ra::cexpr_math::tan<Real>(f*pi_/2.0);
115         Real sig_sqr = ra::cexpr_math::sqr<Real>(sig_);
116         Real sqrt_2a = ra::cexpr_math::sqrt<Real>(2.0 * a);
117         Real sqrt_2 = ra::cexpr_math::sqrt<Real>(2.0);
118         Real a0 = (a * sig_sqr) + (sqrt_2a * sig_) + 1.0;
119         Real a1 = 2.0 * ((a*sig_sqr) - 1.0);
120         Real a2 = (a * sig_sqr) - (sqrt_2a * sig_) + 1.0;
121         Real b0 = sig_sqr + (sqrt_2 * sig_) + 1.0;
```

```
122         Real b1 = 2.0 * (sig_sqr - 1.0);
123         Real b2 = sig_sqr - (sqrt_2 * sig_) + 1.0;
124
125         a0=a0/b0; a1=a1/b0; a2=a2/b0; b1=b1/b0; b2=b2/b0;
126         biquad_filter_coefs<Real> obj(a0,a1,a2,1.0,b1,b2);
127         return obj;
128     }
129
130     // Returns the coefficients for a biquad low-frequency shelving
131     // cut filter with normalized cutoff frequency f and gain-control
132     // parameter a, where f in [0,1] and a >= 0.
133     // For a gain in dB of G (where G < 0), choose a = 10 ^ (-G / 20).
134     // The filter coefficients are always normalized such that the
135     // coefficient b0 is 1.
136     // The type Real is a floating-point type.
137     // All real arithmetic should be performed with the Real type.
138     template<class Real>
139     constexpr biquad_filter_coefs<Real> lowshelf_cut(Real f, Real a){
140         Real pi_ = ra::cexpr_math::pi<Real>;
141         Real sig_ = ra::cexpr_math::tan<Real>(f*pi_/2.0);
142         Real sig_sqr = ra::cexpr_math::sqr<Real>(sig_);
143         Real sqrt_2a = ra::cexpr_math::sqrt<Real>(2.0 * a);
144         Real sqrt_2 = ra::cexpr_math::sqrt<Real>(2.0);
145         Real a0 = sig_sqr + (sqrt_2 * sig_) + 1.0;
146         Real a1 = 2.0 * (sig_sqr - 1.0);
147         Real a2 = sig_sqr - (sqrt_2 * sig_) + 1.0;
148         Real b0 = (a * sig_sqr) + (sqrt_2a * sig_) + 1.0;
149         Real b1 = 2.0 * ((a*sig_sqr) - 1.0);
150         Real b2 = (a * sig_sqr) - (sqrt_2a * sig_) + 1.0;
151
152         a0=a0/b0; a1=a1/b0; a2=a2/b0; b1=b1/b0; b2=b2/b0;
153         biquad_filter_coefs<Real> obj(a0,a1,a2,1.0,b1,b2);
154         return obj;
155     }
156
157 }
158 #endif
159
```

```
1  #include "ra/cexpr_basic_string.hpp"
2  #include <iostream>
3  #include <cstddef>
4
5  int main(){
6
7      using std::cout;
8      using std::endl;
9      // Test default constructor
10     ra::cexpr::cexpr_basic_string<char,10> obj_A;
11     const unsigned char s[] = {'j','u','d','e','\0'};
12     // Test constructor with single pointer
13     ra::cexpr::cexpr_basic_string<unsigned char,20> obj_B(s);
14     cout << "obj_B = " << obj_B[0] << obj_B[1] << obj_B[2] << obj_B[3] << endl;
15     cout << "max size: " << obj_B.max_size() << endl;
16     cout << "capacity: " << obj_B.capacity() << endl;
17     cout << "size: " << obj_B.size() << endl;
18     constexpr unsigned char j[] = {'o', 'n', 'y', 'i', 'a', '\0'};
19     // Test constructor with pointers to first and one past last
20     constexpr ra::cexpr::cexpr_basic_string<unsigned char,6> obj_C(&j[0], &j[5])
   ;
21     // Test member function data()
22     const unsigned char* t = obj_C.data();
23     for(std::size_t i=0; i<obj_C.size(); ++i){
24         cout<< *t;
25         ++t;
26     }
27     cout << endl;
28     // Test prevention of pointer (from data()) to change null terminator
29     ra::cexpr::cexpr_basic_string<unsigned char,6> obj_D(obj_C);
30     unsigned char* tt = obj_D.data();
31     for(std::size_t i=0; i<=obj_D.size(); ++i){
32         cout<< *tt;
33         ++tt;
34     }
35     cout << endl;
36     tt = obj_D.end();
37     *tt = 'w';
38     --tt;
39     *tt = 'g';
40     tt = obj_D.begin();
41     for(std::size_t i=0; i<=obj_D.size(); ++i){
42         cout<< *tt;
43         ++tt;
44     }
45     cout << endl;
46     // Test push_back(), pop_back() and append()
47     cout << "obj_B.size(): "<< obj_B.size() << endl;
48     obj_B.push_back('c');
49     cout << "obj_B.size(): "<< obj_B.size() << endl;
50     cout << "obj_C.size(): "<< obj_C.size() << endl;
51     cout << "obj_B.capacity(): "<< obj_B.capacity() << endl;
52     obj_B.append(obj_C);
53     cout << "obj_B.size(): "<< obj_B.size() << endl;
54     for(std::size_t i=0; i<=obj_B.size(); ++i){
55         cout << obj_B[i];
56     }
57     cout << endl;
58     obj_B.pop_back();
59     cout << "Pop_back: ";
60     for(std::size_t i=0; i<=obj_B.size(); ++i){
61         cout << obj_B[i];
```

```
62      }
63      cout << endl;
64      obj_B.append(s);
65      cout << "Append s: ";
66      for(std::size_t i=0; i<=obj_B.size(); ++i){
67          cout << obj_B[i];
68      }
69      cout << endl;
70      // Test clear()
71      obj_B.clear();
72      cout << "obj_B.clear. Then obj_B.size(): "<< obj_B.size() << endl;
73      // Test to_string()
74      //std::size_t n = 596;
75      char* buffer = obj_A.begin();
76      //constexpr std::size_t size = 10;
77      char okay = 'a';
78      char* ptr_okay = &okay;
79      char** end = &ptr_okay;
80      cout<<"To String"<<endl;
81      cout<<"size: "<<(ra::cexpr::to_string(596,buffer,obj_A.capacity(),end))<<endl;
82      cout<<"string: "<<buffer<<endl;
83      cout<<"Before end: "<<(*((*end)-1))<<endl;
84      cout<<"end: "<<(**end)<<endl;
85      return 0;
86  }
87
```

```cpp
#include<fstream>
#include<iostream>
#include"ra/mandelbrot.hpp"

int main(){
    // Force the image (in PNM format) to be computed at compile time.
    constexpr auto s = ra::fractal::mandelbrot<256, 256>;

    std::ofstream outfile;
    outfile.open("mandelbrot.pnm");

    // Output the image (in PNM format).
    outfile << s.begin() << std::endl;
    outfile.close();
    std::cout <<"Written to mandelbrot.pnm"<<std::endl;
    return 0;
}
```

```
1  #include<iostream>
2  #include"ra/cexpr_math.hpp"
3
4  int main (){
5      using std::cout;
6      using std::endl;
7      constexpr long double pi_ = ra::cexpr_math::pi<long double>;
8      cout <<"Pi= "<<pi_<<endl;
9      constexpr float var_A = 23.75;
10     constexpr float var_B = -98.22;
11     cout <<"abs(23.75)= "<<ra::cexpr_math::abs<float>(var_A)<<endl;
12     cout <<"abs(−98.22)= "<<ra::cexpr_math::abs<float>(var_B)<<endl;
13     cout <<"sqr(23.75)= "<<ra::cexpr_math::sqr<float>(var_A)<<endl;
14     cout <<"cube(−98.22)= "<<ra::cexpr_math::cube<float>(var_B)<<endl;
15     cout <<"mod(−6.8*pi_,2*pi_)= "<<ra::cexpr_math::mod<long double>(-6.8*pi_,2*pi_)<
   <endl;
16     cout <<"sin(−5000*pi_/26)= "<<ra::cexpr_math::sin<long double>(-5000*pi_/26)<<end
   l;
17     cout <<"sin(4*pi_/3)= "<<ra::cexpr_math::sin<long double>(4*pi_/3)<<endl;
18     cout <<"cos(−5000*pi_/26)= "<<ra::cexpr_math::cos<long double>(-5000*pi_/26)<<en
   dl;
19     cout <<"cos(4*pi_/3)= "<<ra::cexpr_math::cos<long double>(4*pi_/3)<<endl;
20     cout <<"tan(−5000*pi_/26)= "<<ra::cexpr_math::tan<long double>(-5000*pi_/26)<<end
   l;
21     cout <<"tan(4*pi_/3)= "<<ra::cexpr_math::tan<long double>(4*pi_/3)<<endl;
22     cout <<"srqt(1377)= "<<ra::cexpr_math::sqrt<long double>(1377)<<endl;
23
24     return 0;
25 }
```

```cpp
#include<iostream>
#include"ra/biquad_filter.hpp"

int main () {
    using std::cout;
    using std::endl;
    constexpr ra::biquad::biquad_filter_coefs<long double> obj_A(23.6,12.0,2.9,19.0,34.8,134.5);
    constexpr ra::biquad::biquad_filter_coefs<float> obj_B(obj_A);
    cout<<"obj_A(23.6,12.0,2.9,19.0,34.8,134.5): "<<obj_A.a0<<" "<<obj_A.a1<<" "<<obj_A.a2<<" "<<obj_A.b0<<" "<<obj_A.b1<<" "<<obj_A.b2<<endl;
    cout<<"obj_B(obj_A): "<<obj_B.a0<<" "<<obj_B.a1<<" "<<obj_B.a2<<" "<<obj_B.b0<<" "<<obj_B.b1<<" "<<obj_B.b2<<endl;
    constexpr ra::biquad::biquad_filter_coefs<long double> obj_lp(ra::biquad::lowpass(0.33,0.7071));
    constexpr ra::biquad::biquad_filter_coefs<long double> obj_hp(ra::biquad::highpass(0.67,0.7071));
    constexpr ra::biquad::biquad_filter_coefs<long double> obj_bp(ra::biquad::bandpass(0.56,0.7071));
    constexpr ra::biquad::biquad_filter_coefs<long double> obj_boost(ra::biquad::lowshelf_boost(0.33,1.77827941));//Gain of 5
    constexpr ra::biquad::biquad_filter_coefs<long double> obj_cut(ra::biquad::lowshelf_cut(0.33,2.818382931));//Gain of -9
    cout<<"lp: "<<obj_lp.a0<<" "<<obj_lp.a1<<" "<<obj_lp.a2<<" "<<obj_lp.b0<<" "<<obj_lp.b1<<" "<<obj_lp.b2<<endl;
    cout<<"hp: "<<obj_hp.a0<<" "<<obj_hp.a1<<" "<<obj_hp.a2<<" "<<obj_hp.b0<<" "<<obj_hp.b1<<" "<<obj_hp.b2<<endl;
    cout<<"bp: "<<obj_bp.a0<<" "<<obj_bp.a1<<" "<<obj_bp.a2<<" "<<obj_bp.b0<<" "<<obj_bp.b1<<" "<<obj_bp.b2<<endl;
    cout<<"boost: "<<obj_boost.a0<<" "<<obj_boost.a1<<" "<<obj_boost.a2<<" "<<obj_boost.b0<<" "<<obj_boost.b1<<" "<<obj_boost.b2<<endl;
    cout<<"cut: "<<obj_cut.a0<<" "<<obj_cut.a1<<" "<<obj_cut.a2<<" "<<obj_cut.b0<<" "<<obj_cut.b1<<" "<<obj_cut.b2<<endl;


    return 0;
}
```