Peter Julius M. Estacio

Grouped with:

Niño Aloysius V. De Mesa

Christian James S. Madariaga

ENGG 151.01-A

February 11, 2026

Progress Report 3 for Project 1: Normalized Crosscorrelation

As of February 11, 2026, major progress has been made. Header implementation was used to compartmentalize validation of integers and doubles, signal file reading, and computing the normalized crosscorrelation signal.

Validation of integers and floating-point numbers was integrated via the functions `is_int()` and `is_floating_pt()`, specifically for processing the signal files for the starting index and the raw signal values themselves.

Signal file reading was implemented via the `readSignalFile()` function. Given a filename and pointers to variables for the starting index and the duration, it can open a signal file, initially store signal values in a vector, then store them in a dynamically initialized array. The starting index and duration variables of the read signal are then modified by the function.

Computing the normalized crosscorrelation was implemented via the `computeCrossCorr()` function. The function uses the starting index, duration, and the signal values of two signals *x* and *y*, then modifies a passed array `*crossCorrData` to store the results of the calculation, along with variables for the resulting array's duration and starting index.

File output was also implemented in the main implementation file, using the resulting array containing the crosscorellation signal along with the results' duration and starting index.

Lastly, the program accepts arguments passed as it is invoked from the command line for the *x* signal filename, *y* signal filename, and the resulting signal filename.

Further work includes verifying if the program meets all necessary specifications, and testing for the robustness of the algorithm and its implementation in the program.

```
PS D:\Peter's Stuff\New Coding\Estacio_Peter\ENGG 151\Project 1> ./main xdata ydata crossCorr
Signal file 'xdata' with starting index -4 and duration 7 read.
Signal file 'ydata' with starting index -4 and duration 8 read.
p_xy(-7) = 0.0271085
p_xy(-6) = -0.187591
p_xy(-5) = 0.241085
p_xy(-4) = 0.280844
p_xy(-3) = -0.536025
p_xy(-2) = 0.323856
p_xy(-1) = -0.130121
p_xy(0) = 0.0462651
p_xy(1) = 0.111687
p_xy(2) = -0.529519
p_xy(3) = 0.311205
p_xy(4) = -0.121807
p_xy(5) = 0.174579
p_xy(6) = -0.0115663
Exported crosscorrelation signal to crossCorr with start index -7 and duration 14.
PS D:\Peter's Stuff\New Coding\Estacio_Peter\ENGG 151\Project 1>
```

sample screenshot – executable main run and displaying results of main program

outputFile

```
1  -7  0.0271085
2  -0.187591
3  0.241085
4  0.280844
5  -0.536025
6  0.323856
7  -0.130121
8  0.0462651
9  0.111687
10 -0.529519
11 0.311205
12 -0.121807
13 0.174579
14 -0.0115663
```

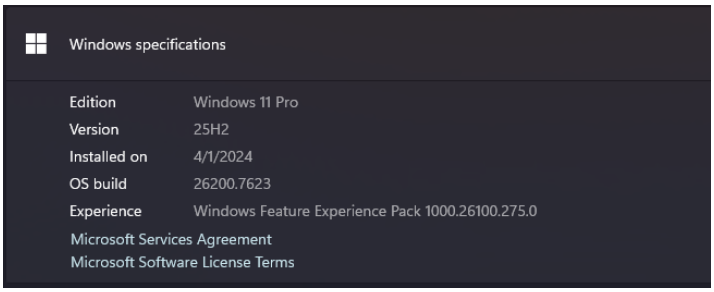
sample screenshot – crosscorrelation signal output to a file named “outputFile”

```
Executing task: C/C++: WINDOWS BUILD (g++.exe)

Starting build...
cmd /c chcp 65001>nul && C:\msys64\mingw64\bin\g++.exe -fdiagnostics-color=always -g *.cpp -o "D:\Peter's Stuff\New Coding\Estacio_Peter\ENGG 151\Project 1\main.exe"

Build finished successfully.
Terminal will be reused by tasks, press any key to close it.
```

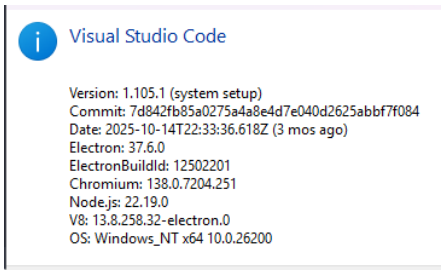
sample screenshot – compilation on Visual Studio Code

```
PS D:\Peter's Stuff\New Coding\Estacio_Peter> g++ -v
Using built-in specs.
COLLECT_GCC=C:\msys64\mingw64\bin\g++.exe
COLLECT_LTO_WRAPPER=C:/msys64/mingw64/bin/../lib/gcc/x86_64-w64-mingw32/15.2.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../gcc-15.2.0/configure --prefix=/mingw64 --with-local-prefix=/mingw64/local --with-native-system-header-dir=/mingw64/include --libexecdir=/mingw64/lib --enable-bootstrap --enable-checking=release --with-arch=nocona --with-tune=generic --enable-mingw-wildcard --enable-languages=c,lto,c++,fortran,ada,objc,obj-c++,jit --enable-shared --enable-static --enable-libatomic --enable-threads=posix --enable-graphite --enable-fully-dynamic-string --enable-libstdcxx-backtrace=yes --enable-libstdcxx-filesystem-ts --enable-libstdcxx-time --disable-libstdcxx-pch --enable-lto --enable-libgomp --disable-libssp --disable-multilib --disable-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-libiconv --with-system-zlib --with-gmp=/mingw64 --with-mpfr=/mingw64 --with-mpc=/mingw64 --with-isl=/mingw64 --with-pkgversion='Rev8, Built by MSYS2 project' --with-bugurl=https://github.com/msys2/MINGW-packages/issues --with-gnu-as --with-gnu-ld --with-libstdcxx-zoneinfo=yes --disable-libstdcxx-debug --enable-plugin --with-boot-ldflags=-static-libstdc++ --with-stage1-ldflags=-static-libstdc++
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 15.2.0 (Rev8, Built by MSYS2 project)
PS D:\Peter's Stuff\New Coding\Estacio_Peter>
```

compiler version – gcc version 15.2.0 (Rev8, Built by MSYS2 project)



operating system - Windows 11 Pro, 25H2, build 26200.7623



IDE – Visual Studio Code ver 1.105.1