**Test Plan**

**CST8152 – Compilers, Assignment #1**

# A. Standard Test

* Method: Buffer runs in 3 modes: fixed-size, additive self-incrementing and multiplicative with standard input file **ass1.pls**. Run in fixed-size mode with standard empty file **ass1e.pls**.
* Source files: buffer.c, buffer.h, platy\_bt.c
* Output files: ass1ai.out, ass1e.out, ass1fi.out, ass1mi.out
* Tools: Total Commander, Command Prompt
* Expected result: The output files should match byte-to-byte with the provided output files,
* Result:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | File | Mode | Working |
| 1 | ass1ai.out | Additive | Y |
| 2 | ass1e.out | Fixed | Y |
| 3 | ass1fi.out | Fixed | Y |
| 4 | ass1mi.out | Multiplicative | Y |

# B. Additional Test

* Method: Buffer runs in 3 modes: fixed-size, additive self-incrementing and multiplicative with input file **bigf.pls**. Modified b\_allocate with different values of init\_capacity, inc\_factor and o\_mode
* Source files: buffer.c, buffer.h, platy\_bt.c
* Output files: see attached **Additional Test Files for a Working Buffer**
* Expected result: The output files should match byte-to-byte with the provided output files. In case the buffer cannot be created, the output file size should be 0 with error message.
* Tools: Total Commander, Command Prompt
* Result: see attached **Additional Test Files for a Working Buffer.**

# C. Leaking Memory Test

* Method: Buffer runs in any mode with standard input file ass1.pls. Using Visual Studio 2019 CRT Library to enable memory leak detection and run the program in debug mode using Visual Studio 2019.
* Source files: buffer.c, buffer.h, platy\_bt\_leaking.c (inside **Modified platybt\_c** directory).
* Output files: ass1ai.out or ass1fi.out or ass1fi.out.
* Expected result: The output files should match byte-to-byte with the provided output files. Visual Studio detected no memory leak.
* Tools: Total Commander, Visual Studio 2019
* Result:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | File | Mode | Working |
| 1 | ass1ai.out | Additive | Y |
| 2 | ass1fi.out | Fixed | Y |
| 3 | ass1mi.out | Multiplicative | Y |

# D. B\_FULL Macro Test

* Method: Buffer runs in any mode with standard output file ass1.pls and fixed mode with empty file ass1e.pls. The **B\_FULL** name can be defined or undefined. **B\_FULL** will be called to test buffer capacity in function **display()**.
* Source files: buffer.c, buffer.h, platy\_bt\_macro.c (inside **Modified platybt\_c** directory).
* Output files: macro\_ass1ai.out, macro\_ass1fi.out, macro\_ass1fi.out, macro\_undef\_ass1ai.out, macro\_undef\_ass1fi.out, macro\_undef\_ass1mi.out
* Expected result: The output files should print out files have same content with the standard output file, but with another line specifying the capacity status of the character array.
* Tools: Total Commander, Command Prompt
* Result:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | File | Mode | Working |
| 1 | macro\_ass1ai.out | Additive | Y |
| 2 | macro\_ass1fi.out | Fixed | Y |
| 3 | macro\_ass1mi.out | Multiplicative | Y |
| 4 | macro\_undef\_ass1ai.out | Additive | Y |
| 5 | macro\_undef\_ass1fi.out | Fixed | Y |
| 6 | macro\_undef\_ass1mi.out | Multiplicative | Y |