Assumptions

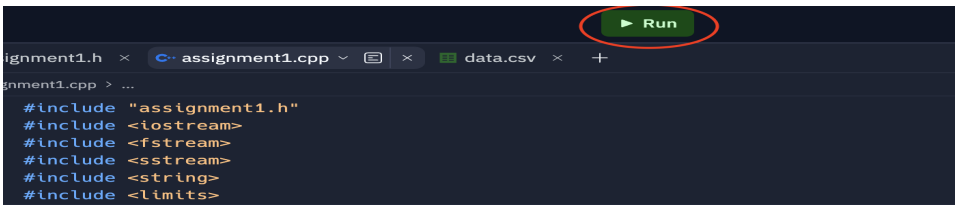
1. The maximum size of the data_array is 150.
2. The input file for the ReadFile function is in CSV format and contains integers separated by commas.
3. If the file contains more than 150 integers, only the first 150 integers will be read and stored.
4. The program assumes that invalid data in the file (not convertible to integers) will be caught as std::invalid_argument exceptions.
5. The ModifyValue function assumes that it will be used to modify values at valid indices within the range [0, data_size).
6. The AddInteger function assumes that it will be used to add integers to the end of the data_array if there is space.
7. The AddInteger function checks for valid integers, and invalid input (e.g., non-integer or float input) is caught as a std::invalid_argument exception.
8. The AddInteger function assumes that if the data_array is full (contains 150 integers), it will throw a std::overflow_error exception.
9. The ReplaceWithZero function assumes that it will be used to replace values at valid indices within the range [0, data_size) with 0.
10. The main function assumes that the user will enter valid integer values when prompted and that they will not input more integers than the data_array can hold.
11. The program assumes that if an error occurs (e.g., invalid input or an overflow), it will print an error message and exit with an error code of 1

How To Use

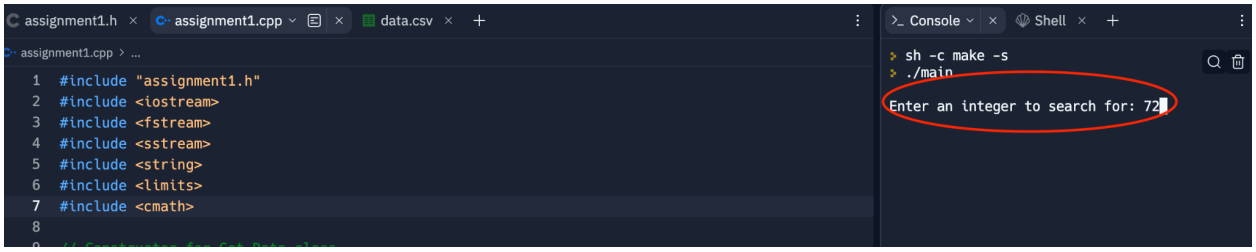
1. Open Assignment1.cpp, Assignment1.h and whichever data file you want to read from
2. Change the name on line 153 at the top of the main function where it says “data.csv” to the name of the data file you would like to use, this program expects to parse through a comma separated value file

```
150  // Main function
151  int main() {
152      Get_Data data;
153      const char* filename = "data.csv"; // Enter file name here
154
```

3. With all 3 files open in the same working directory, click run on the cpp file



4. From the console, enter an integer that you want to search for in the file



5. Click enter
6. Again from the console, enter an integer value for an index you would like to edit
7. Click enter
8. Again from the console, enter the integer value you would like to replace the previous index with
9. Click enter
10. Again from the console, enter an integer value that you would like to add to the end of the array
11. Click enter
12. Again from the console, enter an integer value that you would like to replace with 0
13. Click enter
14. If at any point you enter an invalid input, the code will terminate, refer to step 3 and continue.