Input and output, variables, functions, parameters

1. (1p) Write a program that asks the user for their name and then prints “Hello [NAME]” and how many letters the name has to the console.
   You may find std::cout, std::cin, and std::size() useful.
2. (2p) Write a program that asks the user for a number. Convert that number to an integer and write a function to return the second power of that number.
   You may find std::stoi() useful.
3. (3p) Write a program that asks the user for a string and a delimiter as command-line arguments. Write a function that finds all delimiters in the string and prints the text separated by those delimiters in their own lines.
   For example if the program’s name was separator.exe, you could type “separator.exe I.am.a.student” and the program would print “I”, “am”, “a”, and “student” on separate lines.
4. (2p) File **assignment_function_value_reference_clock.cpp** contains code that performs function calls by value and by reference with char and string data types. Run the code. Which was faster for operating on a char variable, pass by value or pass by reference? How about when operating on a string variable? Why do you think the results are as they are?
   Write your answers as comments at the top of the file.
5. (3p) Write a program that returns the n’th (user-defined index) number in the Fibonacci sequence (https://en.wikipedia.org/wiki/Fibonacci_sequence).
   You may want to use unsigned long integers (https://www.tutorialspoint.com/cplusplus/cpp_data_types.htm) or similar data types that allow very large numbers.
6. (4p) Write a program that performs encryption and decryption on strings using Caesar cipher (https://en.wikipedia.org/wiki/Caesar_cipher). You may use the code given in **assignment_caesar_cipher.cpp** as a base or write from scratch.
   You may find std::string::find() and std::size() useful. Remember that indices start from zero. Remember to convert variables to the same data type before doing operations such as logical comparison to them. You may assume that the message contains only letters of the alphabet (no whitespaces or symbols).