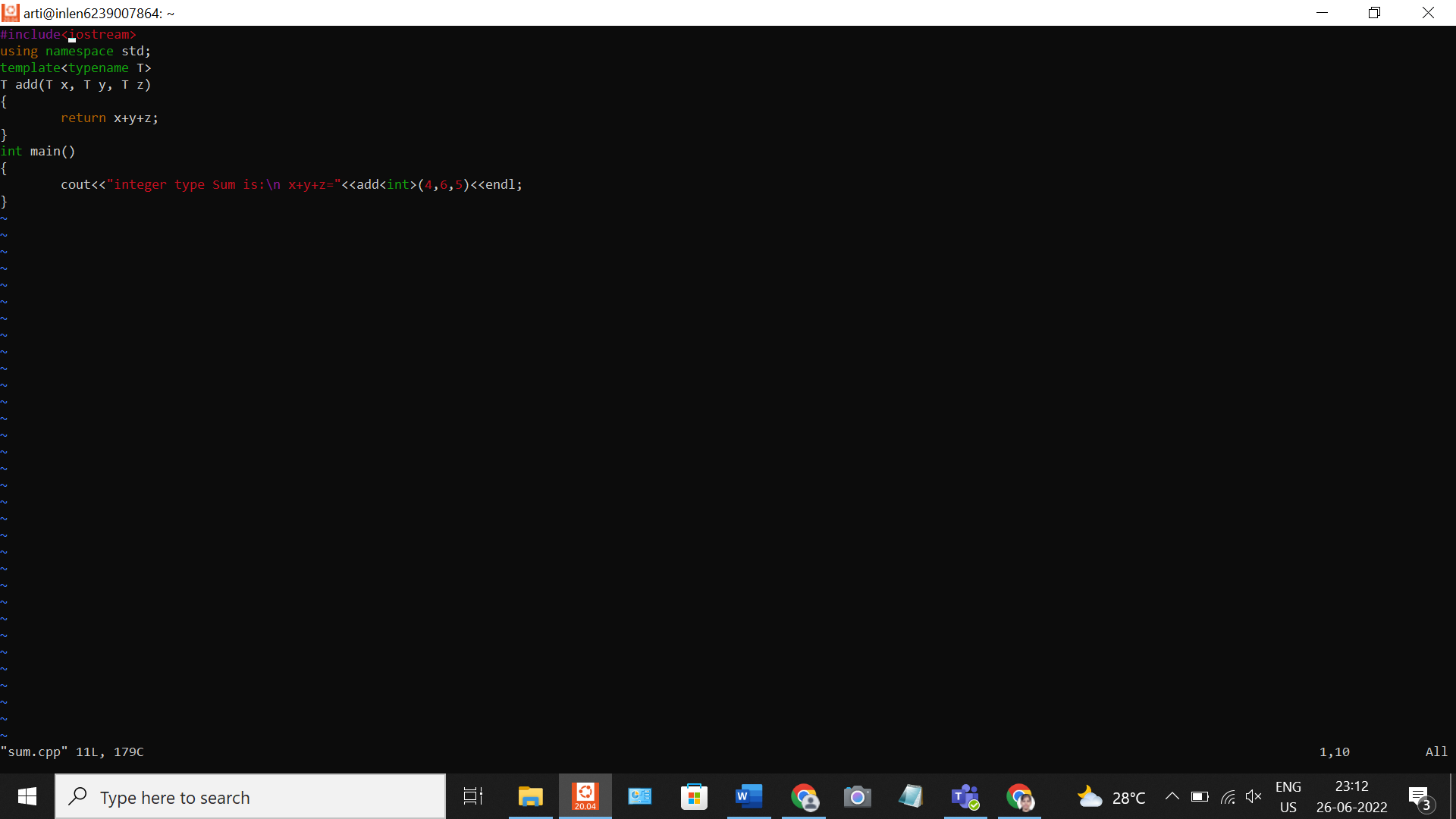
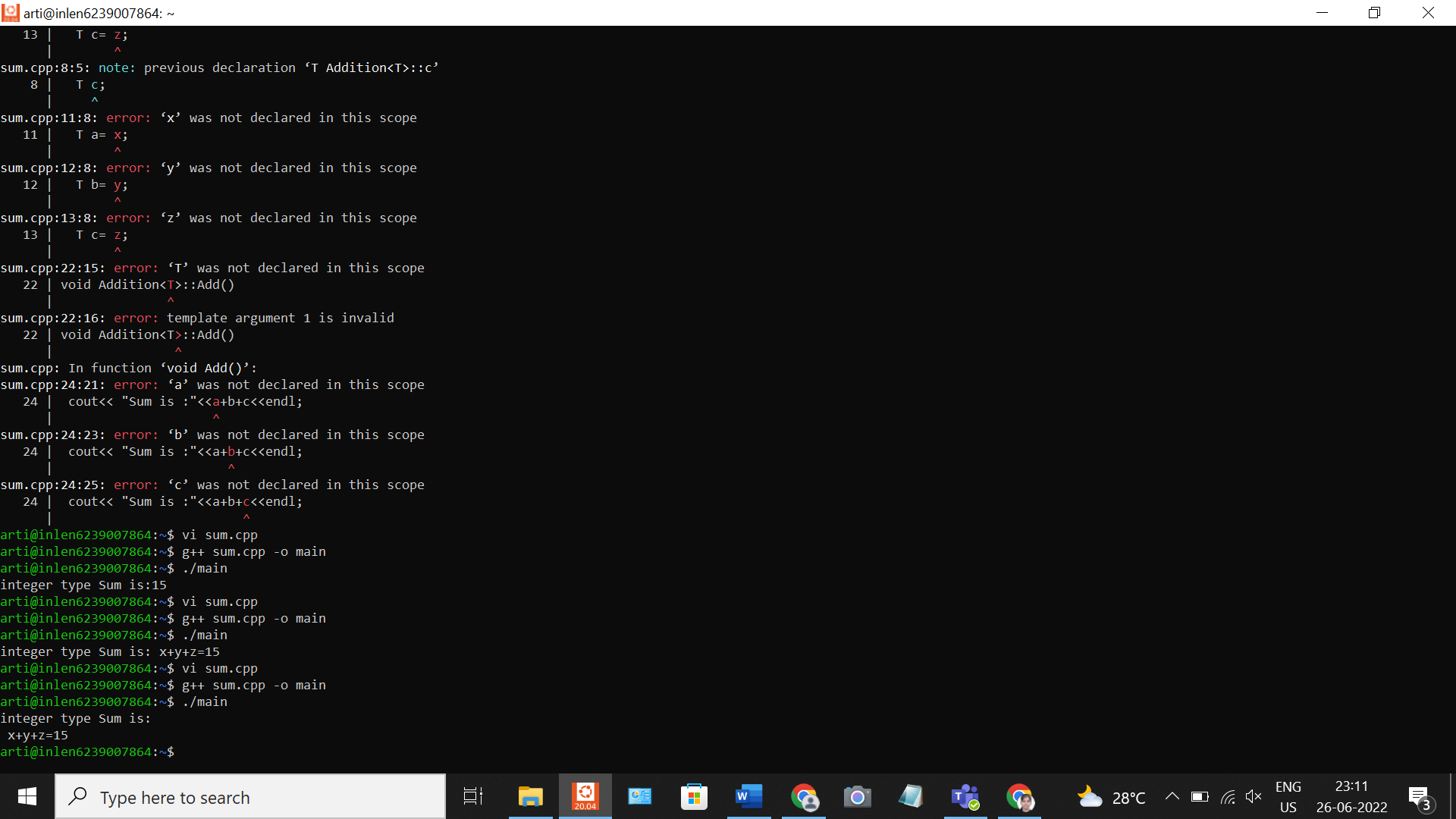
**I/O Generic Programming, Function and Class Templates Assignments**

**Mandatory Assignments**

1. Write a templated function to add() a maximum of 3 input strings OR 3 integers and return the final string or number. add() function may be called with atleast 2 arguments.

**Ans:**





1. Write a templated merge sort function which takes an array of data using merge sort with signature as below. Test with string and float data.

template <class T, int num>

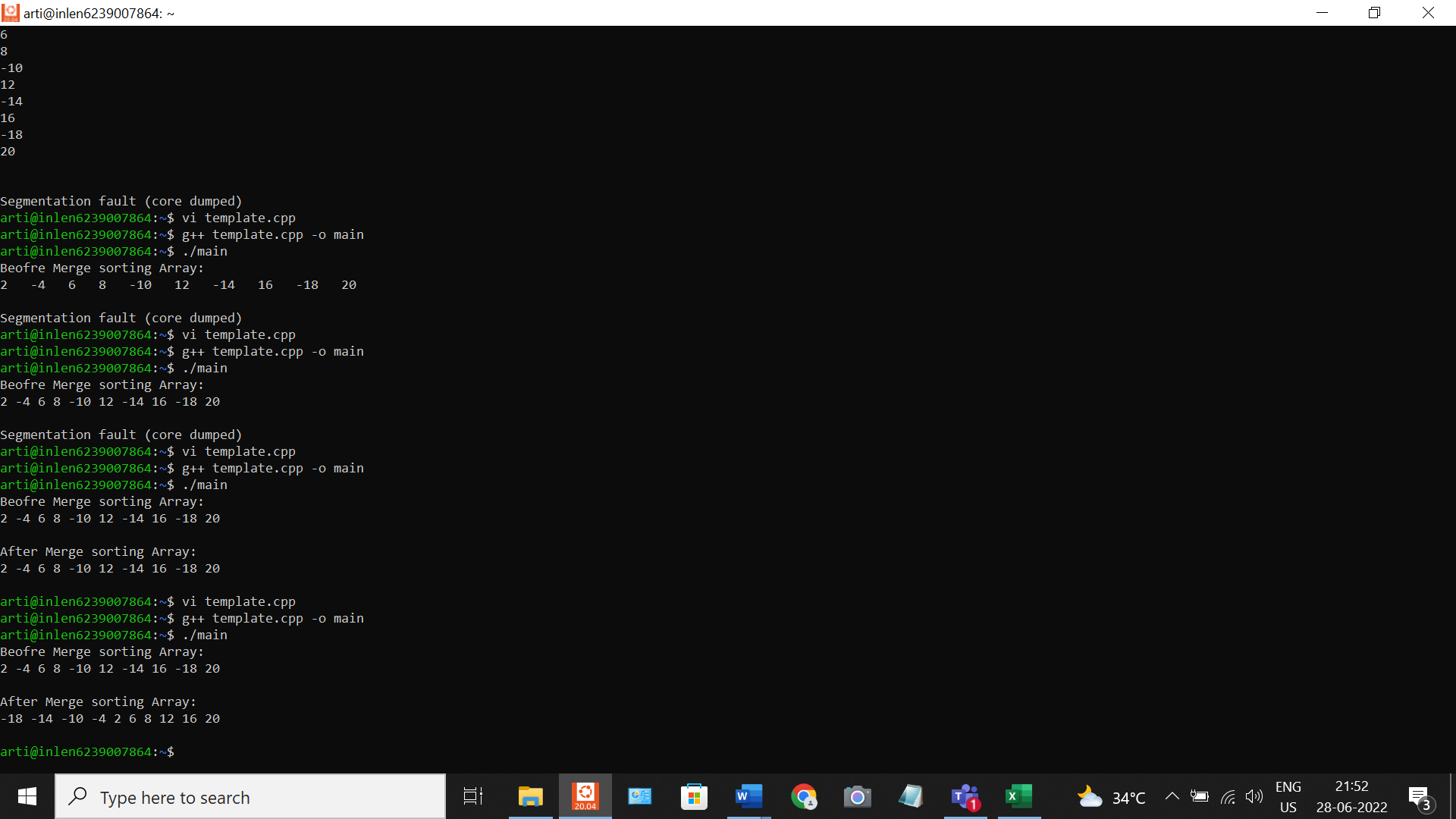
int merge\_sort(T arr[], int num> //num is no. of elements in array

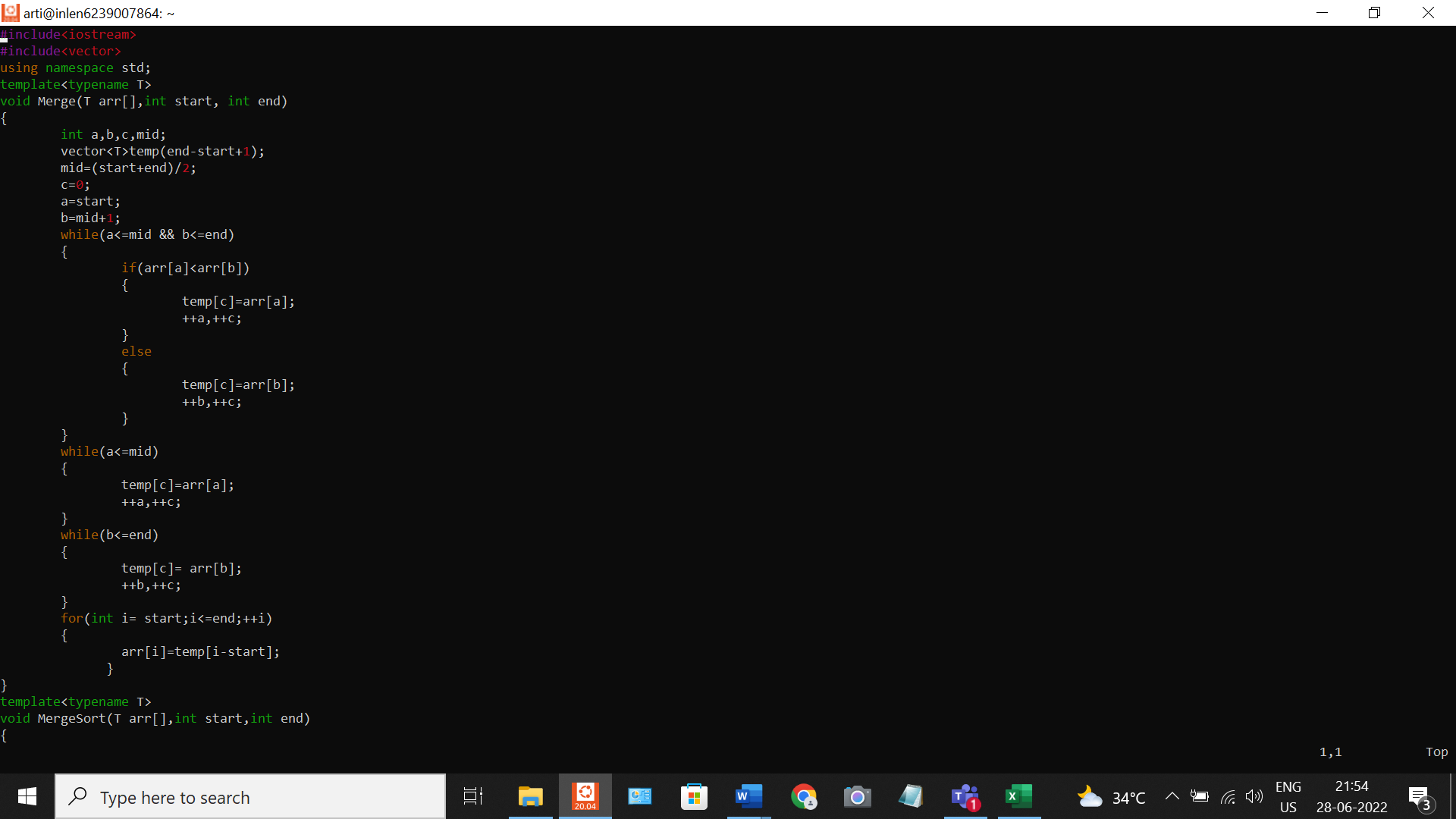
{

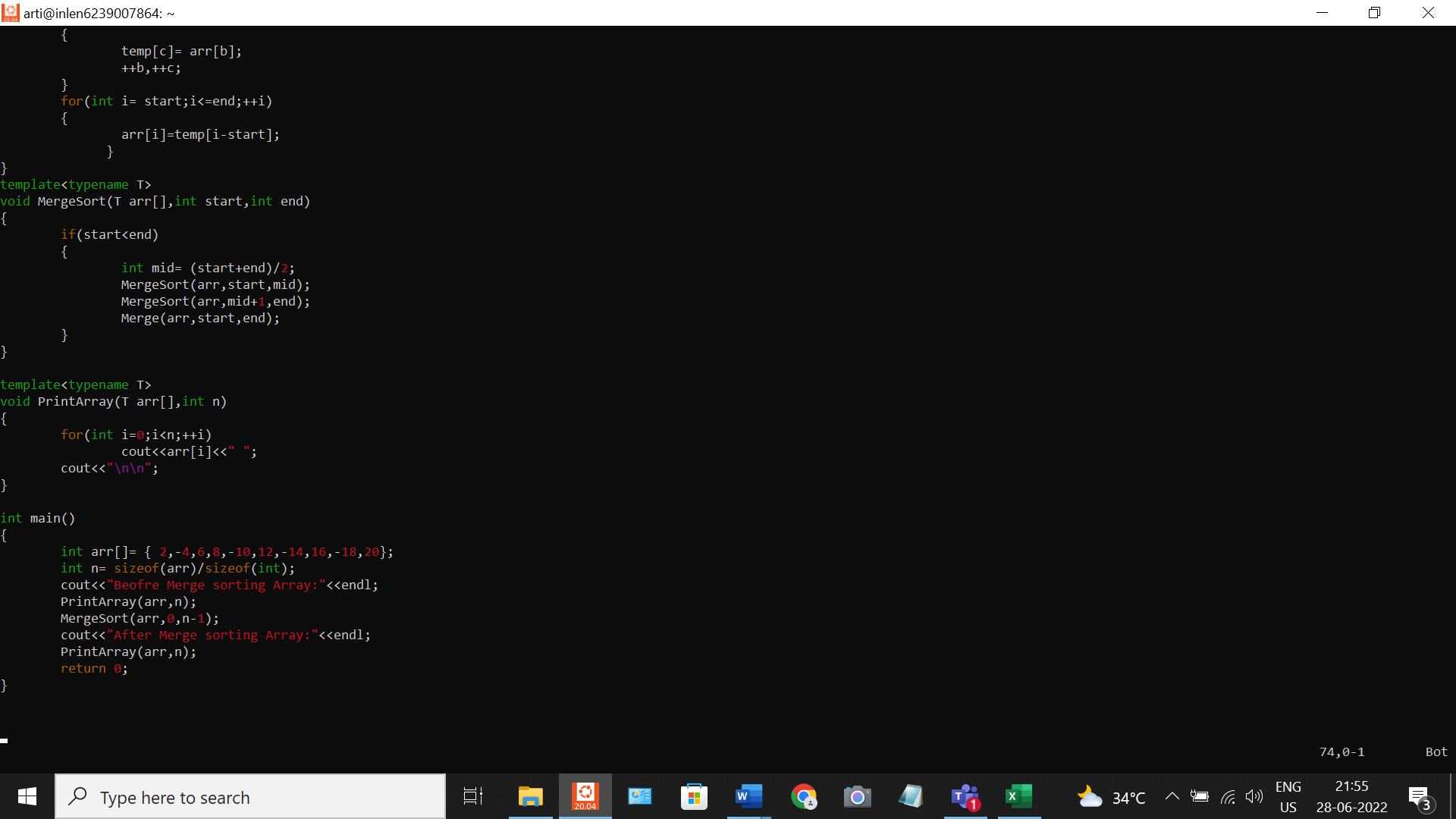
//Add implementation here and return SUCCESS i.e 0

}

**Ans.**







1. Refer the file “function\_template\_classobj.cpp”. Complete the sections marked with TBD 1..4x to get the expected out as mentioned in the file.

* TBD1: Constructor
* TBD2: Create object allocating memory from heap, handle exception
* TBD3: Implement template function myadd()
* TBD4a, b: Deallocate the assigned memory

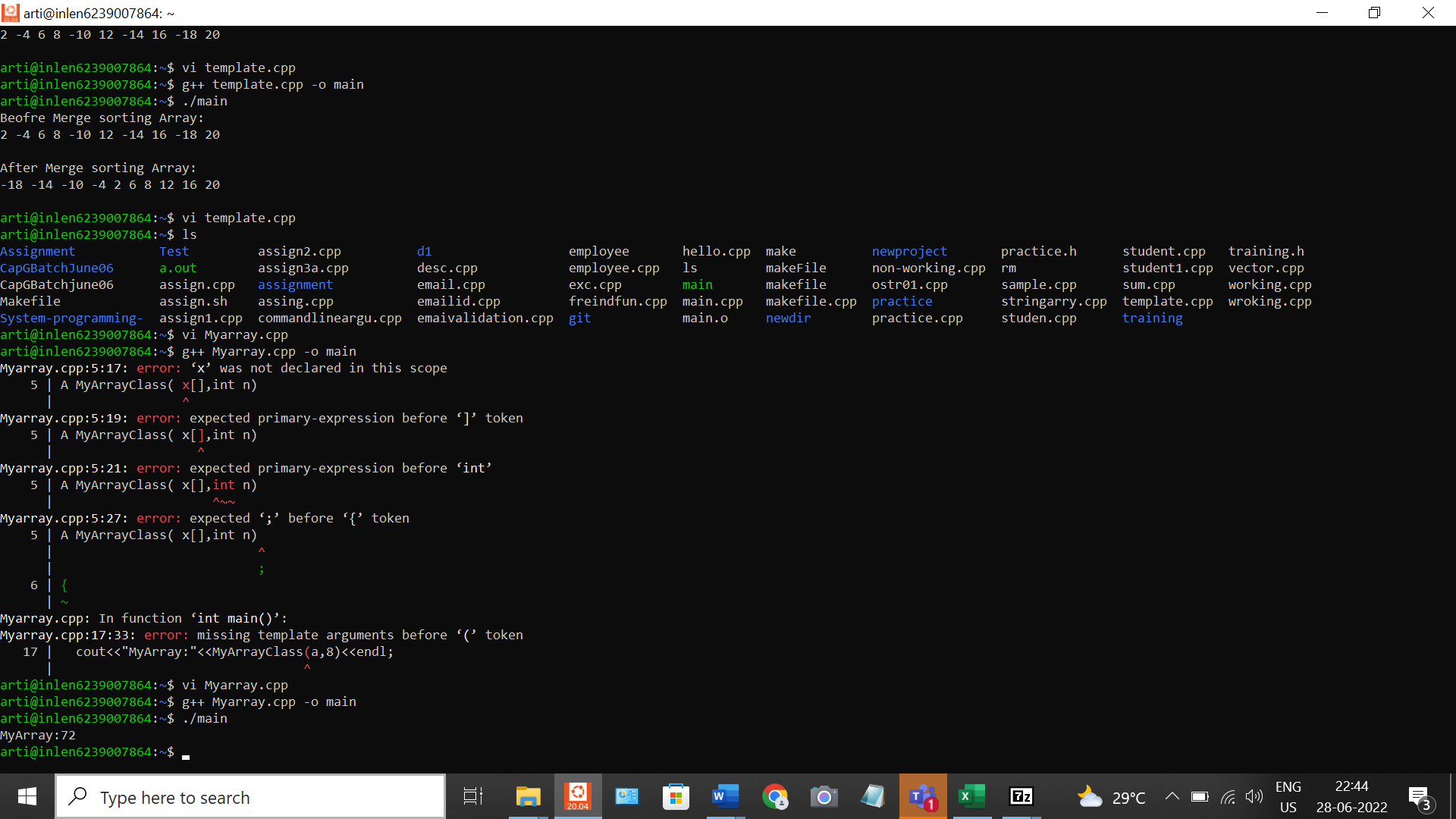
Verify your output with expected output

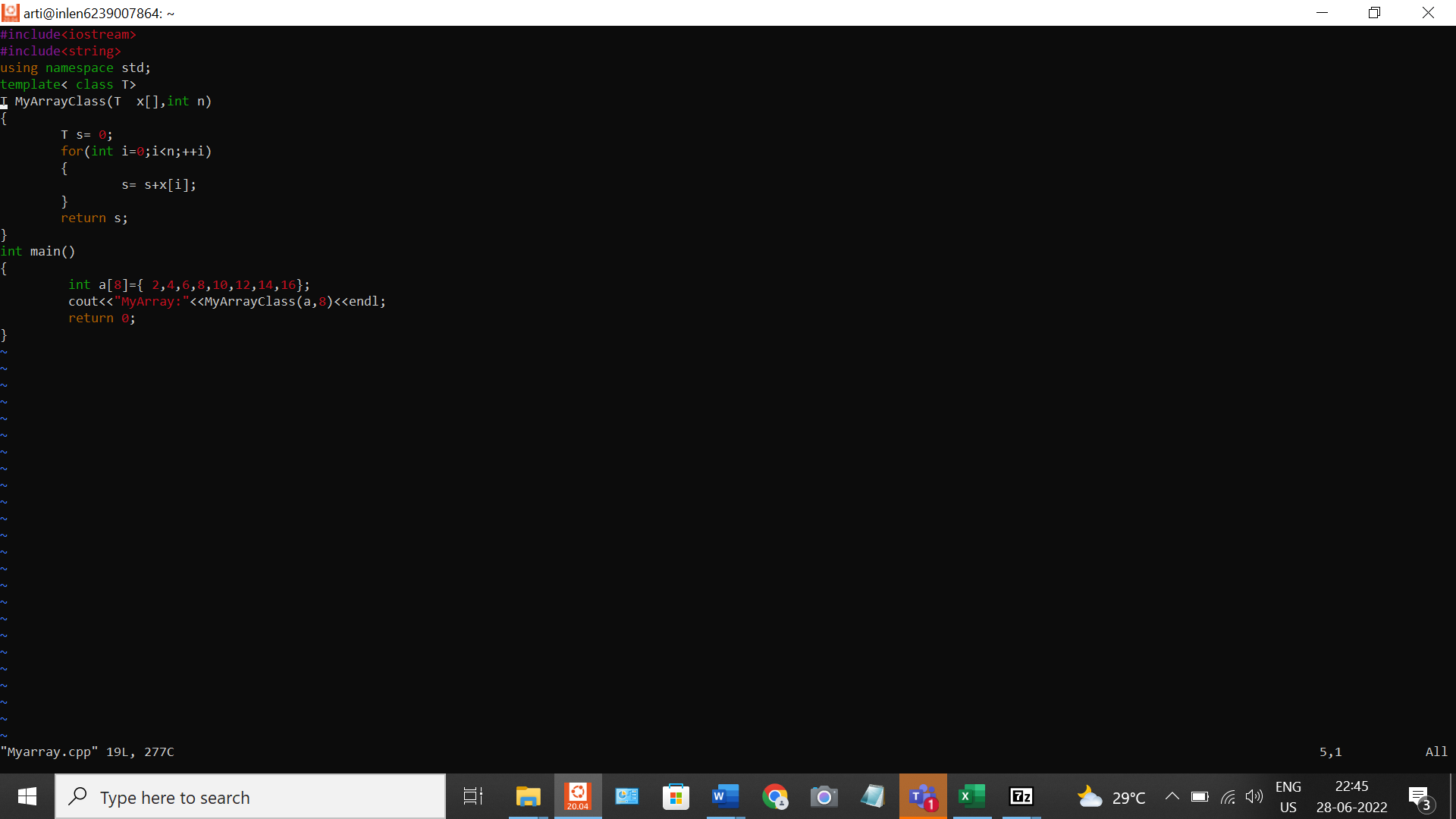
1. Create a templated class named “MyArrayClass” with templated 1D array of MAX\_ELEMENTS.
   1. Populate the array using function below.

// populate MAX\_ELEMENTS in the 1D array with input arr content, count = no . of elements in input array

PopulateData(T \*arr, int count;

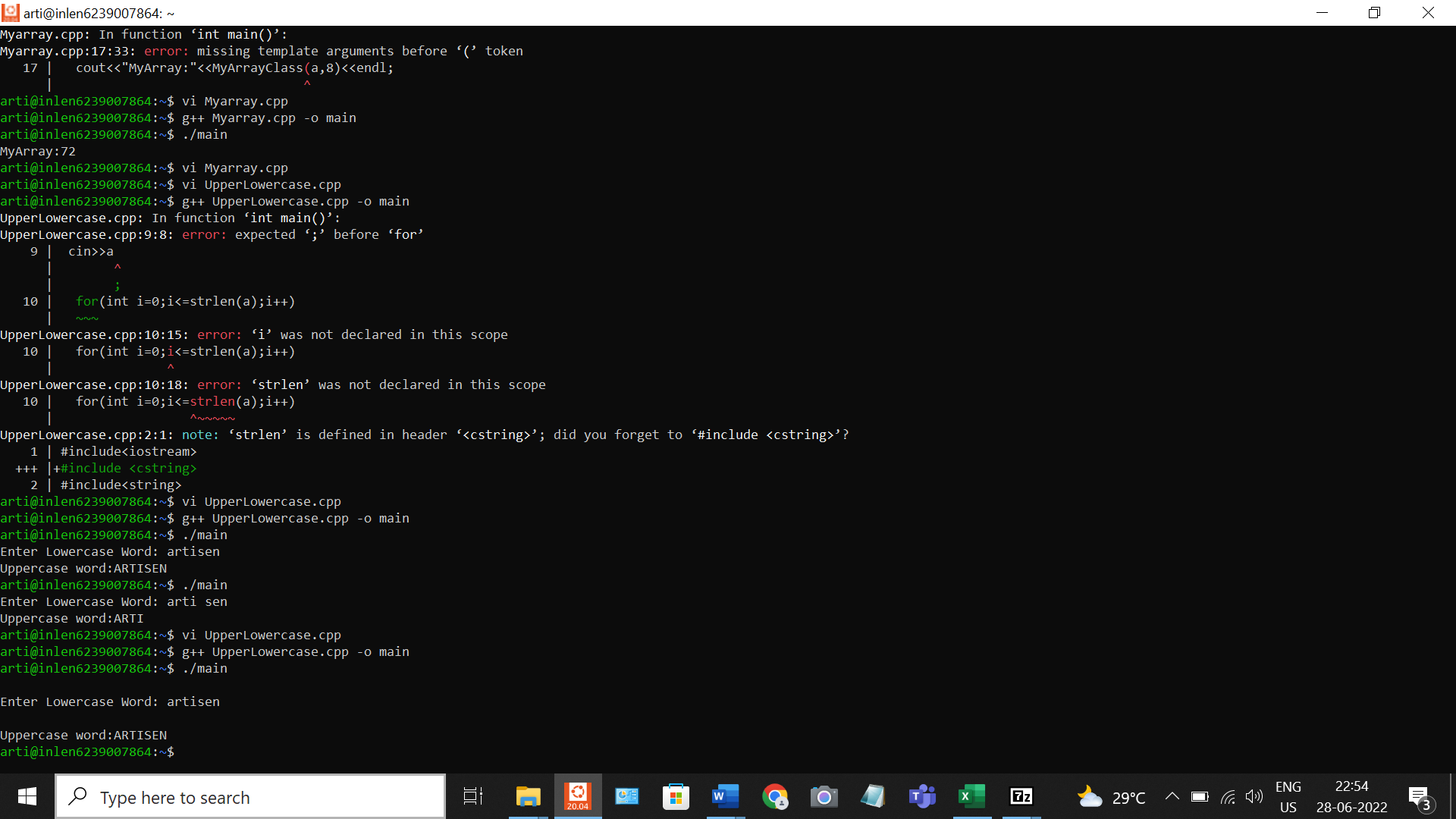
**Ans:**

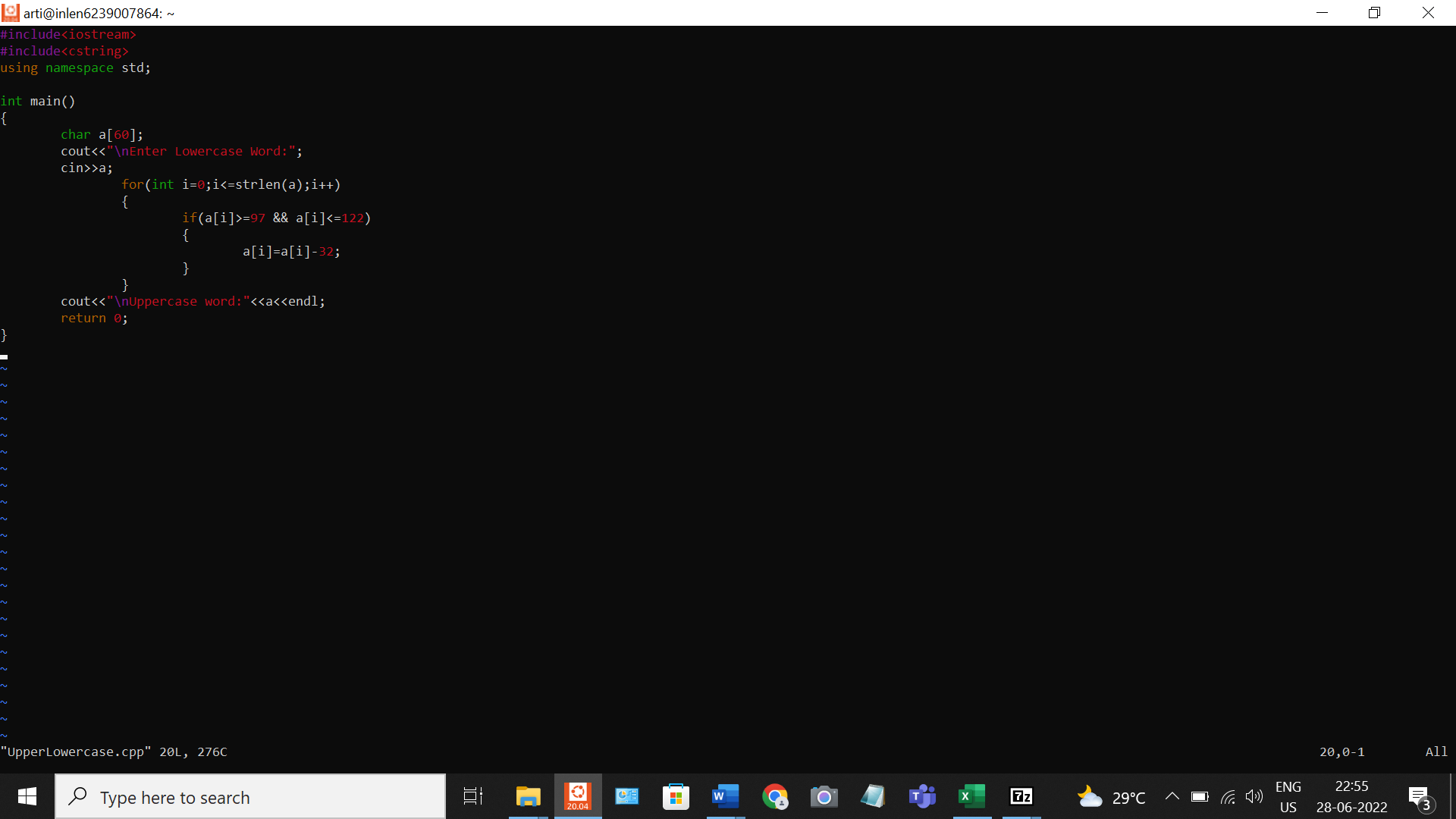




* 1. For character data provide customized handling i.e convert given character to uppercase and store.

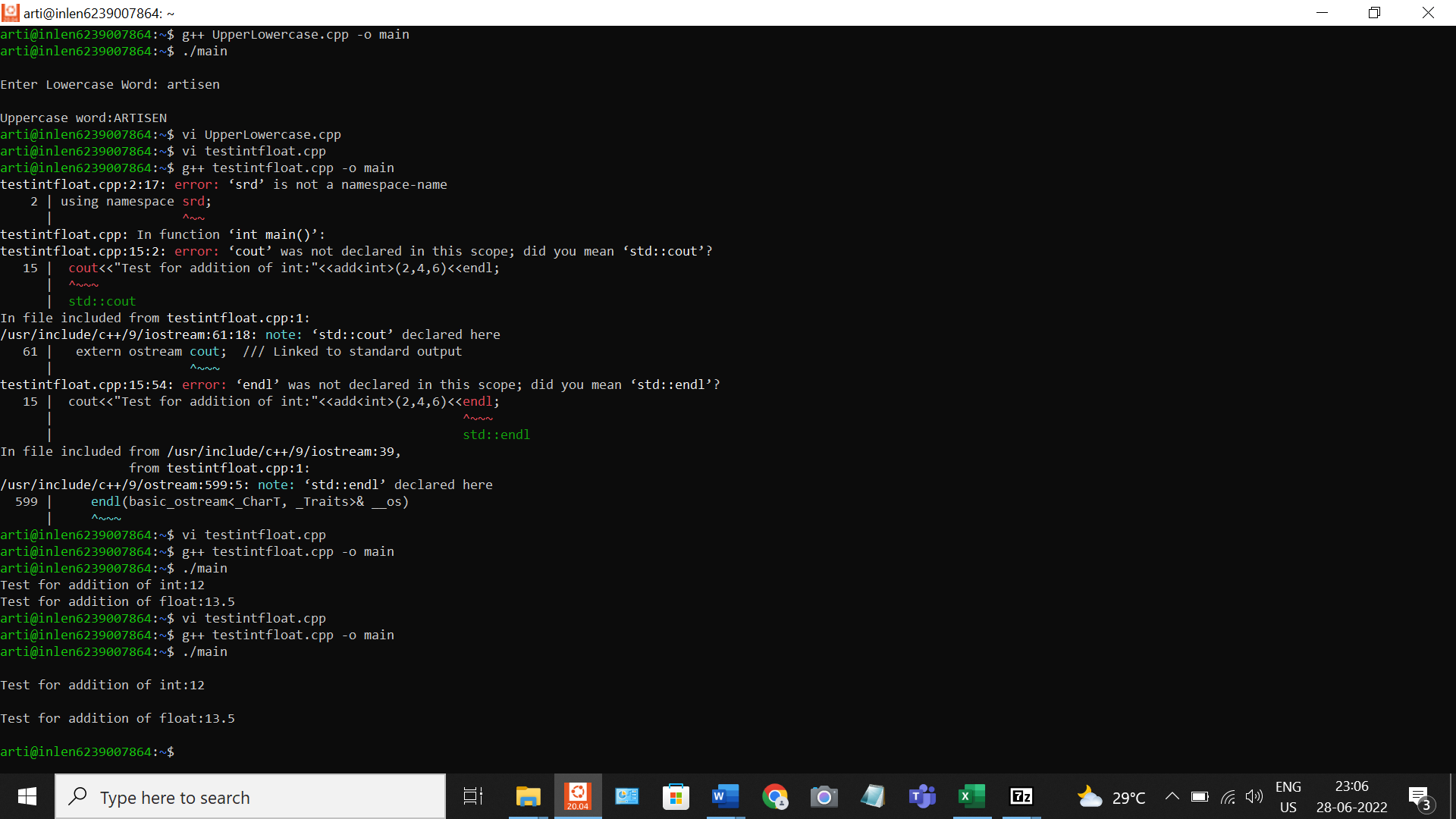
**Ans:**

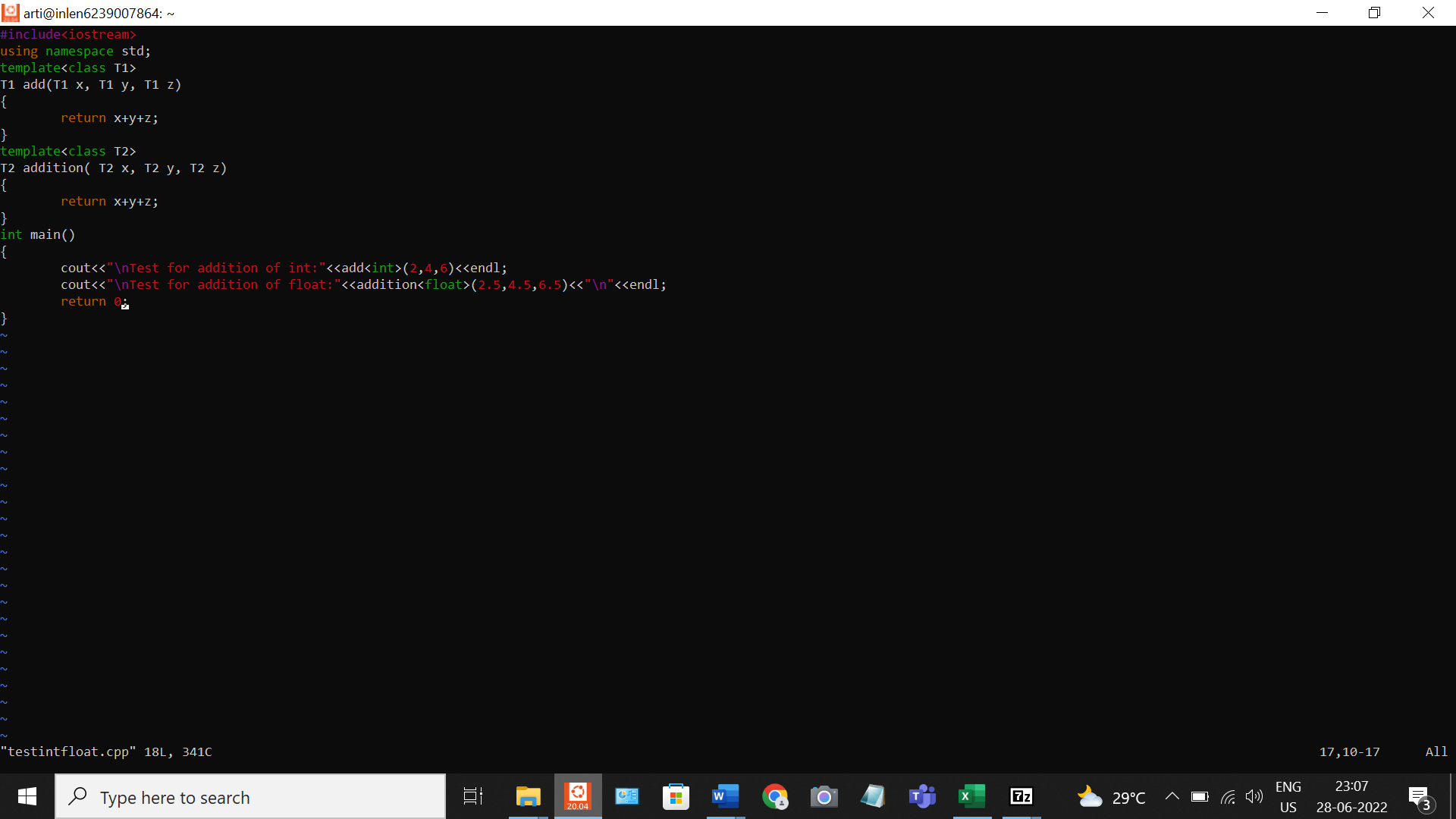




* 1. Test with int, float and char data.

**Ans:**





**Optional Assignments:**

Opt1. Implement a templated stack class with push(), pop(), isempty(), isfull() functions to work with any type of data . Test stack with string and integer data. Extract and display the object type of the stack objects created

Opt2. Implement a class Sort to sort any type of data. Test with char, integer, string datasets.