1. Implement stack using - (a) array, (b) dynamic array, and (c) linked list.
    a. Each of the implementations needs to include the following methods:
        i. push
        ii. pop
        iii. peek
        iv. printStack
        v. printStructure
    b. Array size will be given as input.
    c. Only C++ and Java are allowed
    d. You will need to take input from a file. The input format is described in the next section.
    e. You can use built-in linked lists.
    f. You need to have three different implementations in three separate header files.

2. Input and output format:
    a. The first line will mention what type of stack it will be. If it is an array additionally, it will also include a length.
        i. ARRAY = array
        ii. DARRAY = dynamic array
        iii. LL = linked lists
    b. Each of the following lines until the end of the file, will include an operation and an operand.
        i. PUSH = push
        ii. POP = pop
        iii. TOP = peek
        iv. PRINTS = printStack
        v. PRINTT = printStructure

Input	Output
ARRAY 10 PUSH 1 PUSH 2 PUSH 5 POP POP TOP PUSH 4	1 2 1 5 2 1 2 1 1 1 4 1
DARRAY PUSH 1 PUSH 2 PUSH 5 POP POP TOP	1 2 1 5 2 1 2 1 1 1 4 1

PUSH 4	

c. After each operation, it will show the stack state.
d. The output will have to be printed in a file.

3. Bonus task.
    a. Use generics/template to handle any data type.

4. Note that the spec might be updated several times until the submission deadline with additional details and modifications. Check ELMS from time to time if there are any additional updates/requirements.

5. Submission requirements:
    a. Take all the codes(only codes) and put them in a folder with your roll number as the folder's name.
    b. Compress it in .zip format and submit it.