1. Write a program which implements a bounded stack (a Last-In-First-Out structure using sequential memory storage), aimed at storing real numbers, and a set of methods operating on this structure:

- the constructor with the size of the stack as a parameter (with memory allocation),
- the copy-constructor (with memory allocation),
- the assignment operator= (with memory dellocation and allocation),
- the comparison operator==,
- the destructor (with memory deallocation),
- push – adding a real number to the stack or throwing a full stack exception,
- pop – removig from the stack a real number pushed there most recently or throwing an empty stack exception,
- top – returning the most recently pushed number or throwing an empty stack exception,
- print – printing the values which are on the stack currently, starting from the one pushed recently,
- empty – informing whether the stack is empty,
- size – returning the number of the items on the stack,
- clear – removing all the items from the stack.

The program should use all these methods to make the operations as follows:

1) create a stack containing the numbers given by the user,
2) print the contents of the stack,
3) remove two values from the stack and print them,
4) print the top number,
5) print the number of values on the stack if it is nonempty, or an appropriate message otherwise,
6) clear the stack.