STACKS AND QUEUES

Problem 1. Dynamic MathStack

The MathStack class has the following member functions.

Function	Description
add	pops the first two values off the stack, adds them together, and pushes the sum onto the stack
sub	pops the first two values off the stack, subtracts the second value from the first, and then pushes the difference onto the stack
mult	Pops the top two values off the stack, multiplies them, and pushes their product onto the stack
div	Pops the top two values off the stack, divides the second value by the first, and pushes the quotient onto the stack
addAll	Pops all values off the stack, adds them, and pushes their sum onto the stack
multAll	Pops all values off the stack, multiplies them, and pushes their product onto the stack

Demonstrate the class with a driver program.

Problem 2. File Reverser

Write a program that opens a text file and reads its contents into a stack of characters. The program should then pop the characters from the stack and save them in a second text file. The order of the characters saved in the second file should be the reverse of their order in the first file.

Problem 3. Write a function that uses a stack to test whether a given string is a palindrome.

Problem 4. Write the function returns a new stack containing all prime numbers appeared at the input stack and the total number of prime numbers. The order of prime numbers on the new stack is not important. After the execution of the function, the content of the input stack must be unchanged.

Problem 5. Write the function transforming a decimal number into a binary number by using stack.

Problem 6. Write the function that removes all even numbers from the given stack. The mutual order of odd numbers must stay unchanged. The function returns the number of removed numbers.

Problem 7. Write the function that takes a given queue and returns a new queue that consists of negative elements from the input queue.

Problem 8. File Filter

Write a program that opens a text file and reads its contents into a queue of characters. The program should then dequeue each character, convert it to uppercase, and store it in a second file.

Problem 9. Write the iterative (then recursive) function that moves elements from the stack into the queue. The input stack must stay unchanged. The order of elements in the queue must be the same as on the stack (the element that is popped from the stack must be the same as the element that is dequeued from the queue).

Problem 10. Write the function that takes the queue and copies into a new queue only elements that contains the subarray which is received as input function argument.

For example, the queue has: "12abc6", "ahgd67", "qtgab0", "ghab45"; the input subarray is "ab". The function must returns the new queue containing the elements: "12abc6", "ghab45".