Assignment 8

Submit answers to questions 1 and 2:

1. A *palindrome* is a sequence that reads the same backwards as forwards. Write a program that reads in an integer array `a[]` and checks if it is a palindrome i.e. it checks whether the reverse of `a[]` is the same as `a[]`. e.g.
   `a[]={1,2,3,4,5,4,3,2,1}` is a palindrome.
   (You can modify the not-in-place reversal of an array example that we did in class. Once you have the reversed array `b[]`, then check if each element in `a[]` is equal to the corresponding element in `b[]`. It is also possible to check directly if the corresponding elements are equal, without explicitly reversing the array at all.)

2. Consider the following algorithm to find *all* the prime numbers less than a given positive number `N`: Start with all the numbers in an array a[] `={2,3,4,5,…,N-1}`. Now, keeping `a[0]=2`, filter out of this array all other numbers that are divisible by 2 to leave `{2,3,5,7,9,11,13,15,17… }`. Next, keeping `a[1]=3`, filter out of the remainder of the array, all numbers that are divisible by 3, leaving `{2,3,5,7,11,13,17,.. }`. At the ith iteration, keep `a[i]`, and filter from `a[i+1],…,a[L-1],` all numbers that are divisible by `a[i]`, where L is the current length of the array. By the end of this process, all the numbers remaining in the array are prime numbers. Using the filter pattern discussed in the notes, write a C program that implements this algorithm.

3. Write a program to find the value in an integer array a[] of length N that occurs at least N/2 + 1 times, if such a value exists. Try to think of a way to do this in linear (`O(N)`) time.

4. Write a program to remove all punctuation marks from an inputted message. The program should read an input message from a user into an array of characters; it should then modify the contents of the array, so that all punctuation marks are removed; and finally it should print out the modified message using a single `printf` statement.
   If the user types:
   `Too, many, punctuation marks:; by far!!`
   the program should output
   `Too many punctuation marks by far`
   For the purposes of this question, a punctuation mark is any one of the following seven characters: `!'?;:,.`
   The solution this question should use the `fgets` *function to read an entire line of input from the console. This is the input string that must be processed by the program, as per the following code excerpt:*

```
/* declare an array of characters */

char msg[200];

/* read a full line of input from the user and put
   it into msg[]. Stop if newline is reached or 200
   characters have been read. Like scanf(), this
function automatically appends the null terminator
to the msg[] array.
*/

fgets(msg, 200, str);
```