Assignment 10

The questions 1-6 below are all about packaging some of the programs that were given in previous assignments into functions. As well as writing the required function, also write a main program that calls the function.
Submit questions 5 and 7 by Friday 25 Nov, 5pm

1. Write a function to compute the area of a triangle, according to the formula [½ (base * height)].

2. Write a function to compute the area of a circle, as PI*radius*radius.

3. Write a function that takes an integer array as input and returns `true` or `false` depending on whether or not it is a palindrome (see Question 1, Assignment 8).

4. Write a function to count how many words occur in a given input string, where a word is a sequence of characters that are not separated by spaces (see Question 1, Assignment 9)

5. Write a function `removeChars` that takes two strings as input and removes all the characters that appear in the second string from the first string e.g. if the first string is `"hello..there, you!"` and the second string is `".!,"`, then, when the function returns, the first string contains `"hellothere you"` (see Question 4, assignment 8).

6. Write a function that takes an integer array `a[]` as input and an integer `k` and returns the kth largest value in the array (see Question 2, assignment 7).

7. The prime factorization of a given positive number N is the set of prime numbers that when multiplied together give N. For instance, the prime factorization of `N=8`, is `{2,2,2}` since `2x2x2 = 8` and the prime factorization of 24 is `{2,2,2,3}` since `2x2x2x3 =24`. Write a function that prints out the prime factorization of an integer N given as its single input argument.

   (Note, it is not actually necessary to compute the set of prime numbers in order to solve this problem. A simple algorithm checks all the numbers between 2 and N, to see if they divide into N. If one does, keep dividing it into N until it no longer divides:

```
for i=2 to N
   while (N % i == 0){ /* while i divides into N */
       print out i
       N = N/i;        /* divide i out of N */
}
```

   A better loop would terminate when N==1.