CS111 Homework 5

Name your program **hw5.cpp**. Program must be able to compile or you will get at most 0.5 out of 2 points for the assignment.

**Due: Thursday May 3, 2018 by 11:59 PM**

Write one single complete C++ program to do all two sections below:

Use the following main function to test for both part A and part B:

```cpp
int main(){
    cout << "Part A:\n";
    int arr[3] = { 5050155, 5, 707070 };  // Changed the array values
    for (int i = 0; i < 3; i++){
        cout << countEven(" " << arr[i] << " ") = " << countEven(arr[i]) << endl;
        cout << removeEven(" " << arr[i] << " ") = " << removeEven(arr[i]) << endl;
        cout << hasEven(" " << arr[i] << " ") = " << (hasEven(arr[i]) ? "Yes" : "No") << endl;
        printStarDigit(arr[i]);
    }
    cout << "Part B:\n";
    int a[4] = { 7, 2, 8, 3 };  // Changed the array values
    int b[5] = { 3, 4, 5, 6, 7 };
    cout << "The range of array a is " << range(a, 4) << endl;
    cout << "The range of array b is " << range(b, 5) << endl;
    reverse(a, 4);
    reverse(b, 5);
    cout << "Array a reversed: ";
    for (int i = 0; i < 4; ++i)
        cout << a[i] << " ";
    cout << endl;
    cout << "Array b reversed: ";
    for (int i = 0; i < 5; ++i)
        cout << b[i] << " ";
    cout << endl;
    return 0;
}
```
Part A: Recursion (from professor Yang)
Write a recursive function `countEven` that returns the count of even digits in the integer parameter.

Write a recursive function `removeEven` that returns a new number with all the even digits removed from the integer parameter.

Write a recursive function `hasEven` that returns true if the integer parameter contains an even digit, false otherwise.

Write a recursive function called `printStarDigit` which prints out all the digit in the given number passed in as a parameter. Also, it prints out a * at the beginning and end of this number and in between each digit.

Sample run:
Part A:
```
countEven(5050155) = 2
removeEven(5050155) = 55155
hasEven(5050155) = Yes
   *5*0*5*0*1*5*5*

countEven(5) = 0
removeEven(5) = 5
hasEven(5) = No
   *5*

countEven(707070) = 3
removeEven(707070) = 777
hasEven(707070) = Yes
   *7*0*7*0*7*0*
```
Part B: (from professor Michael Fried)
Modify the program by writing the following functions. Do not modify the main function.

1. Write a range function:
   o This function should take 2 parameters: a 1D int array and the number of elements.
   o It should return the difference between the largest element and the smallest element.

2. Write a reverse function:
   o This function should take 2 parameters: a 1D int array and the number of elements.
   o It should reverse the order of the elements. It should not return anything.
   o Hint: You can use the swap function to swap two elements, for example swap(a[i], a[j]) would swap a[i] and a[j].

Sample run:

```
The range of array a is 6
The range of array b is 4
Array a reversed: 3 8 2 7
Array b reversed: 7 6 5 4 3
```

```c++
#include <iostream>
#include <algorithm>
using namespace std;

// Returns the difference between the largest element and the smallest element
int range(int a[], int size) {
   // Fill in.
}

// Reverses the order of the elements
// Feel free to use the swap function
// (it's in the algorithm library so you don't have to write it yourself)
void reverse(int a[], int size) {
   // Fill in.
}
```