Consider the following C++ program.  

```cpp
#include <iostream>
using namespace std;

int fun(int &x, int &y) {
    if (y <= 0) return x;
    x = x + 2;
    cout << x << y << endl;
    return x * y;
}

int main() {
    int x = 6, y = -1;
    cout << fun(x, y) << endl; // line a
    fun(y, x); // line b
    cout << fun(x, y) << endl; // line c
    return 0;
}
```

What is the output from the program at each of the following lines?

(a) Line a: 6
   (variable y of function fun is -1, thus return x which is 6, prints this value from main)

(b) Line b: 16
   (variable y of function fun is variable x of function main which is 6, x of fun is y of main which is -1. x = x + 2, update x which is -1 (y of main) becomes 1. Prints x of fun and y of fun from function fun)

(c) Line c: 81
   8
   (variable y is no longer less than 0. Variable x of fun is 6, increments to 8. Prints x of fun and y of fun inside function fun. Returns 8*1 which is 8 to main function, this value is then displayed from main)

Write one or more C++ functions that called in the following main function (you have to support function definition with title lines)  

```cpp
int main() {
    int a = 2, b = 3, c = 4;
    // swap a with the biggest of a, b, c. Here prints 4, 3, 2
    swapBig(a, b, c);
    cout << a << b << c << endl;
    return 0;
}
```
void swap(int &x, int &y){
    int temp = x;
    x = y;
    y = temp;
}
void swapBig(int &a, int &b, int &c){
    if ( b > a && b > c ) swap(a, b); // swap a with b if b is the biggest
    else if ( c > a ) swap(a, c); // swap a with c if c is the biggest
}
You only need to write ONE function of the following.
(a) Write a recursive function called firstDown that returns the result of decreasing the first digit in a positive integer by 1.
int main() {
    cout << firstDown(2048) << endl; // prints 1048
    cout << firstDown(19683) << endl; // prints 9683
    return 0;
}
Answer:
int firstDown(int x){
    if ( x < 10 ) return x – 1;
    return firstDown(x / 10) * 10 + x % 10;
}
(b) Write a recursive function called lastEven that returns the last even digit in a positive integer parameter. It should return 0 if there are no even digits. For example, a program that uses the function lastEven follows.
int main() {
    cout << lastEven(23) << endl; // prints 2
    cout << lastEven(1214) << endl; // prints 4
    cout << lastEven(777) << endl; // prints 0
    return 0;
}
Answer:
int lastEven(int x){
    if ( x % 2 == 0 ) return x % 10;
    return lastEven(x / 10);