1. The greatest common divisor of integers \( x \) and \( y \) is the largest integer that evenly divides both \( x \) and \( y \). Write a recursive function \( \text{gcd} \) that returns the greatest common divisor of \( x \) and \( y \). The \( \text{gcd} \) of \( x \) and \( y \) is defined recursively as follows: if \( y \) is equal to 0 then \( \text{gcd}(x, y) \) is \( x \), otherwise, \( \text{gcd}(x, y) \) is \( \text{gcd}(y, x \mod y) \).

2. Can \text{main} be called recursively? Write a program containing a function \text{main}. Include a static variable called count which is initialized to 1 (\text{static int count = 1;}). Postincrement and print the value of \text{count} each time \text{main} is called. Compile your program. What happens?

3. Given below is a main program which creates an ordered array of 12 integers. The program prompts the user to enter an integer called key. It then calls a function named \text{BinSearch} which does a binary search on the array to determine if the key is in the array. The main program reports the results.

Write the function \text{BinSearch} as both a nonrecursive and a recursive function. Run your functions using the main program below to verify that they work.

```cpp
#include<iostream>
using namespace std;
const int ASIZE = 12;

void BinSearch(int a[], int first, int last, int key, bool &found, int &location);

void main()
{
    int a[ASIZE] = {0, 1, 4, 6, 9, 12, 34, 53, 54, 67, 87, 99};
    int finalIndex = ASIZE-1;
    int key, location;
    bool found;
    cout << "Enter the number to be located... ";
    cin >> key;
    while(key < 100)
    {
        BinSearch(a, 0, finalIndex, key, found, location);
        if(found)
            {cout << key << " is in location " << location << endl; }
        else
            {cout << key << " is not in the array. " << endl; }
        cout << "Enter the number to be located... ";
        cin >> key;
    }
    return;
}
```