Consider the following C/C++ code segment.

```c
int g = 4;

int min(int a, int b, int c) {
    int v = a;
    if (b < v)
        v = b;
    if (c < v)
        v = c;
    return v;
}

int p(int i, int j, int k, int l) {
    return min(min(g, i, j), k, l);
}

int gcd(int a, int b) {
    if (b == 0) { 
        return a;
    } else { 
        return gcd(b, a % b);
    }
}
```

Q1. Translate the code segments above into IA32 assembly language using the basic code generation strategy outlined in lectures. You may then generate optimised versions of your code. The % operation can be implemented using the IA32 cdq and idiv instructions.

Q2. What is the maximum depth of the stack (in stack frames) during the calculation of gcd(14, 21)? Draw a diagram showing the state of the stack at its maximum depth during the calculation of gcd(14, 21).

Q3. Using Visual Studio (or similar), create a Win32 application with files t1.h and t1.asm containing the IA32 assembly language for min, p and gcd. Use t1Test.cpp to test min, p and gcd. Please note that the source code provided may need to be modified to work with the development environment you use. Submit your answer, which should include code listings and a snapshot of the console window showing evidence that your program works, via Blackboard. The deadline is on the course web page.