1. (10 points) Free! ☺

2. (30 points) What is the output of this program?

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

int f(int n, int a, int b)
{
    cout << n << ", " << a << ", " << b << endl;
    if (n == 0) return a;
    else if (n == 1) return b;
    else return f(n - 1, b, a + b);
}

class B
{
    public:
        B(int m) { this->m = m; }
        int f() { return m; }
        int f(int x) { return m + x; }
        friend bool operator<(B a, B b);
        friend int m(B b);
    protected:
        int m = 3;
};
bool operator<(B a, B b) { return a.m < b.m; }
int m(B b) { return b.m - 1; }

class D : public B
{
    public:
        D(int m) : B(m) { }
        D operator^=(int n)
        {
            m *= n;
            return *this;
        }
};

int main()
{
    cout << f(9, 0, 1) << endl; // (1)-(10) (1 point each)

    B b(3); D d(4);
    cout << b.f() << endl; // (11)
    cout << d.f(5) << endl; // (12)
    cout << b.f(5) << endl; // (13)
    cout << d.f() << endl; // (14)
    d ^= 3;
    cout << m(d) << endl; // (15)
}
3. (15 points) Write functions to double and print the contents of an array.

```c++
int a[] = {1, 2, 3, 4};
f(a, 4); // output: 2, 4, 6, 8
```

(a) Use iteration

(b) Use recursion

(c) Convert the function that you wrote in part (a) or (b) into a template function. The template function should work for arrays of any type.

4. (20 points) Write a class named `Log` (to simulate logarithm) such that it works as in the `main` below.

```c++
int main()
{
    Log a(4), b;
b.of(8);
cout << a << endl; // output: log(4)
cout << b << endl; // output: log(8)

    Log c = a + b;
cout << c << endl; // output: log(32)
    // Hint: log(4) + log(8) = log(4 * 8) = log(32)

    Log c = 3 * a;
cout << c << endl; // output: log(64)
    // Hint: 3 * log(4) = log(4^3) = log(64)

cout << (a > b) << endl; // output: 0
    // Hint: log(4) < log(8), since 4 < 8
try { c.of(-2); }
catch (int e) { if (e == 1) cout << "[ERROR] logarithm of negative!" << endl; }

    return 0;
}
```

5. (25 points) Answer these questions:

(a) What is difference between function overloading and overriding?

(b) What is a try/catch block?

(c) What is the purpose of inheritance in C++?

(d) List some advantages that a vector offers over an array.

(e) Are `typedef` and `#define` the same in C++? Why?