Problem 1. (20 marks)

For each pair of $f(n)$ and $g(n)$ below, decide if $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, or $f(n) = \Theta(g(n))$. Justify your answer using the definitions of these asymptotic notation. Note that more than one of these relations may hold for a given pair; list all correct ones.

(1) $f(n) = \sqrt{n}$ and $g(n) = \sqrt[3]{n}$.

(2) $f(n) = (log_3 n)^2$ and $g(n) = log_2(n^3)$.

(3) $f(n) = 2^n$ and $g(n) = 2^{2n}$.

(4) $f(n) = log_2(n!)$ and $g(n) = nlog_2 n$.

Problem 2. (20 marks)

Determine the Big O notation of the following code snippets:

(1)

```
void exampleFunction(std::vector<int> arr) {
    for (int i = 0; i < arr.size(); i++) {
        std::cout << arr[i] << std::endl;
    }
    for (int i = 0; i < arr.size(); i++) {
        for (int j = 0; j < arr.size(); j++) {
            std::cout << arr[i] << " " << arr[j] << std::endl;
        }
    }
}
```

(2)

```
void fun(int N, int M) {
    std::vector<int> arr;
    int counter = 0;
    for (int i = 0; i < N; i++) {
        arr.push_back(i);
    }
    for (int i = 0; i < M; i++) {
        counter++;
    }
    std::cout << counter << std::endl;
}
```

Problem 3. (20 marks)

Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove each of the following conjectures.

Hint: You can disprove a conjecture by giving negative examples.

1. $f(n) = O(f(n)/3)$.
2. $f(n) = O(f(n/3))$.

Problem 4. (20 marks)

Solve the following recurrence relation: $T(1) = 1$, $T(n) = 4T(\frac{n}{2}) + 1$, where $n > 1$.

Problem 5. (20 marks)

Determine the Big O notation of the following recursive function in C++:

```cpp
int fibonacci(int n) {
    if (n <= 1) {
        return n;
    } else {
        return fibonacci(n-1) + fibonacci(n-2);
    }
}
```