Binary Problems

Binary to Decimal

Here's one way you could convert a binary number given as a string to a decimal number in C++:

```
#include <iostream>
#include <string>

using namespace std;

int binaryToDecimal(string binary) {
    int decimal = 0;
    int base = 1;
    for (int i = binary.length() - 1; i >= 0; i--) {
        if (binary[i] == '1') {
            decimal += base;
        }
        base *= 2;
    }
    return decimal;
}

int main() {
    string binary = "10110110";
    cout << binary << " in decimal is " << binaryToDecimal(bi
    return 0;
}
```

This code defines a function **binaryToDecimal()** that takes a string **binary** as an input, and converts it to an int by iterating through the string in reverse order. The **binary. Length() - 1** is the index of the last element in the string, this way we start at the rightmost digit, and each iteration moves to the left.

For each digit that is equal to '1', the function adds **base** to **decimal**. The variable **base** starts with the value 1, and is multiplied by 2 for each iteration to account for the positional value of the digit.

At the end of the loop the final value of **decimal** variable is the decimal number that corresponds to the binary number given as a string.

This example will output:

```
10110110 in decimal is 182
```

Decimal to Binary

```
#include <iostream>
#include <string>

using namespace std;

string decimalToBinary(int decimal) {
    string binary = "";
    while (decimal > 0) {
        int remainder = decimal % 2;
        binary = to_string(remainder) + binary;
        decimal /= 2;
    }
    return binary;
}

int main() {
    int decimal = 25;
    cout << decimal << " in binary is " << decimalToBinary(de
    return 0;
}
```

This code defines a function **decimalToBinary()** that takes an integer **decimal** as an input, and converts it to a string by using a while loop. The **while (decimal > 0)** keep running the loop till the number is completely converted to binary.

At each iteration of the loop, the function calculates the remainder of dividing **decimal** by 2, which will always be either 0 or 1. The remainder is then added to the left of the current binary string. Next, the **decimal** is divided by 2 and this becomes the new decimal number.

At the end of the loop the final value of **binary** variable is the binary number that corresponds to the decimal number.

This example will output:

```
25 in binary is 11001
```