

Name: ASWIN SAMPATHID: 53844684

1. (60 pts) Write a function, "addBinary" which takes two binary strings a and b, return their sum as a binary string.

Logic: ① Make both strings equal by adding 0's at the front of the smaller string
 ② Add one more zero at front to handle carry. ③ Loop from behind. & keep track of sum and carry and assign it to the final string. ④ Return final string

```
string addBinary(string a1, string b1) {
    if(a1.size() < b1.size()) {
        string a2 = addZeros(a1, b1.size() - a1.size());
        string b2 = b1;
    } else {
        string b2 = addZeros(b1, a1.size() - b1.size());
        string a2 = a1;
    }
    // Add zero to handle carry
    string b = "0" + b2;
    string a = "0" + a2;
    string c(b.size()); // Declare final string.
    c = addBinaryStringHelper(a, b);
    return c;
}
```

Time complexity: $O(n)$ where n is the bigger string size + 1

```
string addBinaryStringHelper(
    string a, string b) {
    string c(a.size());
    int carry = 0;
    for(i = a.size() - 1; i >= 1; i--) {
        c[i] = (a[i] - '0') ^ (b[i] - '0')
            ^ (carry < 1 ? 0 : 1) + '0';
        carry = (a[i] - '0') & (b[i] - '0')
            & carry;
    }
    c[0] = carry + '0';
    return c;
}
```

```
string addZeros(string s, int n) {
    while(n-- > 0) {
        s = "0" + s;
    }
    return s;
}
```

2. (40 pts each) Write one testcases and one edgecase for your code above.

TC 1: $a = 0010$
 $b = 011$ → make both equal $a = 00010$
 $b = 00011$ → Track from back and handle carry

```

      0 0 0 1 0
    + 0 0 0 1 1
    -----
    0 0 1 0 1
    c = 0101
```

TC 2: $a = 1011$ → 01011
 $b = 1001$ → 01001

```

      0 1 0 1 1
    + 0 1 0 0 1
    -----
    1 0 1 0 0
    c = 10100
```

Edgecase: strings