

# COMSC-200

## Lab 9

Ryan Jacoby

8 November 2020

### 1 Shifting and Printing an Integer

Write a program that right-shifts an integer variable four bits. The program should print the integer before and after the shift operation. Does your system place zeros or ones in the vacated bits.

My system places 0s in the vacated bits; 255(b11111111) becomes 15(b00001111).

```
1 // Ryan Jacoby
2
3 #include<iostream>
4
5 using namespace std;
6
7 int main() {
8     int val;
9     cout << "Enter a value to be bitshifted: ";
10    cin >> val;
11
12    val = val >> 4;
13    cout << "Your value bitshifted 4 itmes is: " << val << '\n';
14    return 0;
15 }
```

Listing 1: main.cpp

```
rjacoby@ryan-pc: ~/devel/cpp/comsc200/Lab 9
> ./lab91
Enter a value to be bitshifted: 255
Your value bitshifted 4 times is: 15
~> ~/devel/cpp/comsc2/Lab 9 > on P master :1 11 77
```

## 2 Multiplication Via Bit Shifting

Left-shifting an unsigned integer by one bit is equivalent to multiplying the value by 2. Write a function `power2` that takes two integer arguments, *number* and *pow* and calculates  $number * 2^{pow}$ . Use a shift operator to calculate the result. The program should print the values as integers and as bits.

```
1 // Ryan Jacoby
2
3 #include<iostream>
4
5 using namespace std;
6
7 int power2(int, int);
8 void print_16bit(int);
9
10 int main() {
11     int a, b;
12     cout << "Enter an integer: ";
13     cin >> a;
14
15     cout << "Enter another integer: ";
16     cin >> b;
17
18     cout << "A: " << a << "\tb";
19     print_16bit(a);
20
21     cout << "\nB: " << b << "\tb";
22     print_16bit(b);
23
24     cout << "\n\n A * 2^B = " << power2(a, b) << "\tb";
```

```

25     print_16bit(power2(a, b));
26     cout << '\n';
27     return 0;
28 }
29
30 int power2(int number, int pow) {
31     return number << pow;
32 }
33
34 void print_16bit(int n) {
35     for(int i = 15; i >= 0; i--) cout << ((n >> i) & 1U);
36 }

```

Listing 2: main.cpp

```

rjacoby@ryan-pc: ~/devel/cpp/comsc200/Lab 9
> ./lab92
Enter an integer: 9
Enter another integer: 12
A: 9    b00000000000001001
B: 12   b0000000000001100

A * 2^B = 36864    b1001000000000000
~/devel/cpp/comsc2/Lab 9 > on P master i1 i1 77

```

### 3 Even Parity Simulation

```

1 // Ryan Jacoby
2
3 #include<iostream>
4 #include<ctime>
5
6 using namespace std;
7
8 int power2(int, int);
9 void print_32bit(int);
10 int count1(int);
11 unsigned int parity_enc(unsigned int *);
12 bool parity_dec(unsigned int);
13 int randomize(int);
14

```

```

15
16 int main() {
17     srand(time(NULL));
18
19     int correct = 0;
20
21     for(int i = 0; i < 10000; i++) {
22         unsigned int n = rand() % (1 << 30);
23         parity_enc(&n);
24         n = randomize(n);
25
26         if(parity_dec(n)) correct++;
27     }
28
29     cout << "Percent transmission: " << correct / 100.0 << "%\n";
30
31     return 0;
32 }
33
34 int power2(int number, int pow) {
35     return number << pow;
36 }
37
38 void print_32bit(int n) {
39     for(int i = 31; i >= 0; i--) cout << ((n >> i) & 1U);
40 }
41
42 int count1(int n) {
43     int ret = 0;
44     for(int i = 31; i >= 0; i--) if((n >> i) & 1U == 1) ret++;
45     return ret;
46 }
47
48 unsigned int parity_enc(unsigned int * n) {
49     if(count1(*n) % 2 == 1) {
50         *n = (*n | (1 << 31));
51         return *n;
52     }
53     return *n;
54 }
55
56 bool parity_dec(unsigned int n) {
57     return count1(n) % 2 == 0;
58 }
59
60 int randomize(int n) {
61     if(rand() % 10 == 0) return (1 << rand() % 32) ^ n;
62     return n;
63 }

```

Listing 3: main.cpp

