peter_ruszel_260_assign4.cpp | peter_ruszel_260_assign3.cpp | peter_ruszel_260_assign2.cpp

```
30      // My tests
31      cout<<"binary 0000 * 00 = "<<multbin("0000", "00")<<endl;
32      cout<<"binary 0001 * 1111 = "<<multbin("0001", "1111")<<endl;
33      cout<<"binary 10 * 1 = "<<multbin("10", "1")<<endl;
34      cout<<"binary 111 * 0 = "<<multbin("111", "0")<<endl;
35      cout<<"binary 1 * 1 = "<<multbin("1", "1")<<endl;
36      cout<<"binary 0 * 0 = "<<multbin("0", "0")<<endl;
37      cout<<"binary 101 * 11 = "<<multbin("101", "11")<<endl;
38      cout<<"binary 11110 * 00001 = "<<multbin("11110", "00001")<<endl;
39      cout<<"binary 110101 * 00101 = "<<multbin("110101", "00101")<<endl;
40      cout<<"binary 111111 * 11111 = "<<multbin("111111", "11111")<<endl<<endl;
41
42      cout<<"hexadecimal 0 * 0 = "<<multhex("0", "0")<<endl;
43      cout<<"hexadecimal 1 * 1 = "<<multhex("1", "1")<<endl;
44      cout<<"hexadecimal 1 * 0 = "<<multhex("1", "0")<<endl;
45      cout<<"hexadecimal 0 * 1 = "<<multhex("0", "1")<<endl;
46      cout<<"hexadecimal 0000001 * 00000 = "<<multhex("0000001", "00000")<<endl;
47      cout<<"hexadecimal 0ACE * 1 = "<<multhex("0ACE", "1")<<endl;
48      cout<<"hexadecimal 1 * FBB = "<<multhex("1", "FBB")<<endl;
49      cout<<"hexadecimal CD * 8 = "<<multhex("CD", "8")<<endl;
50      cout<<"hexadecimal 9 * B = "<<multhex("9", "B")<<endl;
51      cout<<"hexadecimal A5123 * 111 = "<<multhex("A5123", "111")<<endl;
```

```
binary 0000 * 00 = 0
binary 0001 * 1111 = 1111
binary 10 * 1 = 10
binary 111 * 0 = 0
binary 1 * 1 = 1
binary 0 * 0 = 0
binary 101 * 11 = 1111
binary 11110 * 00001 = 11110
binary 110101 * 00101 = 100001001
binary 111111 * 11111 = 11110100001

hexadecimal 0 * 0 = 0
hexadecimal 1 * 1 = 1
hexadecimal 1 * 0 = 0
hexadecimal 0 * 1 = 0
hexadecimal 0000001 * 00000 = 0
hexadecimal 0ACE * 1 = ACE
hexadecimal 1 * FBB = FBB
hexadecimal CD * 8 = 668
hexadecimal 9 * B = 63
hexadecimal A5123 * 111 = B008653
```

peter_ruszel_260_assign4.cpp | peter_ruszel_260_assign3.cpp | peter_ruszel_260_assign2.cpp

```
    // Professor's tests
    cout<<"binary 10001 * 11 = "<<multbin("10001", "11")<<endl; //you should get 110011
    cout<<"binary 100 * 110001 = "<<multbin("100", "11001")<<endl; //you should get 1100100
    cout<<"binary 110 * 1010 = "<<multbin("110", "1010")<<endl; //you should get 111100
    cout<<"binary 11111111 * 10 = "<<multbin("11111111", "10")<<endl; //you should get 111111110
    cout<<"binary 10101010 * 1 = "<<multbin("10101010", "1")<<endl;  //you should get 10101010
    cout<<"binary 0 * 11110000 = "<<multbin("0", "11110000")<<endl<<endl; //you should get 0

    cout<<"hexadecimal F * A = "<<multhex("F", "A")<<endl; //you should get 96
    cout<<"hexadecimal 1A * 5 = "<<multhex("1A", "5")<<endl; //you should get 82
    cout<<"hexadecimal FF * 2 = "<<multhex("FF", "2")<<endl; //you should get 1FE
    cout<<"hexadecimal 104 * 3 = "<<multhex("104", "3")<<endl; //you should get 30C
    cout<<"hexadecimal FABC * 1 = "<<multhex("FABC", "1")<<endl; //you should get FABC
    cout<<"hexadecimal 0 * EFDCAB = "<<multhex("0", "EFDCAB")<<endl<<endl; //you should get 0

    // system("PAUSE");
```

```
binary 10001 * 11 = 110011
binary 100 * 110001 = 1100100
binary 110 * 1010 = 111100
binary 11111111 * 10 = 111111110
binary 10101010 * 1 = 10101010
binary 0 * 11110000 = 0

hexadecimal F * A = 96
hexadecimal 1A * 5 = 82
hexadecimal FF * 2 = 1FE
hexadecimal 104 * 3 = 30C
hexadecimal FABC * 1 = FABC
hexadecimal 0 * EFDCAB = 0
```