

Name:

Login:

IN-CLASS EXERCISE

Bitwise operations

1. Write the result of each expression. Assume all are unsigned char (8 bits).

expression	result
17 5	
88 & 14	
17 88 & 14	
19 >> 2	
19 << 2	
17 88 >> 2	
17 & 88 << 3	

2. Fill in the following function

```
typedef unsigned char uchar;
```

```
uchar get_bits(uchar buf[4], int start_bit, int end_bit) {  
    // start_bit and end_bit are inclusive
```

```
}
```

0	00000000	64	01000000	128	10000000	192	11000000
1	00000001	65	01000001	129	10000001	193	11000001
2	00000010	66	01000010	130	10000010	194	11000010
3	00000011	67	01000011	131	10000011	195	11000011
4	00000100	68	01000100	132	10000100	196	11000100
5	00000101	69	01000101	133	10000101	197	11000101
6	00000110	70	01000110	134	10000110	198	11000110
7	00000111	71	01000111	135	10000111	199	11000111
8	00001000	72	01001000	136	10001000	200	11001000
9	00001001	73	01001001	137	10001001	201	11001001
10	00001010	74	01001010	138	10001010	202	11001010
11	00001011	75	01001011	139	10001011	203	11001011
12	00001100	76	01001100	140	10001100	204	11001100
13	00001101	77	01001101	141	10001101	205	11001101
14	00001110	78	01001110	142	10001110	206	11001110
15	00001111	79	01001111	143	10001111	207	11001111
16	00010000	80	01010000	144	10010000	208	11010000
17	00010001	81	01010001	145	10010001	209	11010001
18	00010010	82	01010010	146	10010010	210	11010010
19	00010011	83	01010011	147	10010011	211	11010011
20	00010100	84	01010100	148	10010100	212	11010100
21	00010101	85	01010101	149	10010101	213	11010101
22	00010110	86	01010110	150	10010110	214	11010110
23	00010111	87	01010111	151	10010111	215	11010111
24	00011000	88	01011000	152	10011000	216	11011000
25	00011001	89	01011001	153	10011001	217	11011001
26	00011010	90	01011010	154	10011010	218	11011010
27	00011011	91	01011011	155	10011011	219	11011011
28	00011100	92	01011100	156	10011100	220	11011100
29	00011101	93	01011101	157	10011101	221	11011101
30	00011110	94	01011110	158	10011110	222	11011110
31	00011111	95	01011111	159	10011111	223	11011111
32	00100000	96	01100000	160	10100000	224	11100000
33	00100001	97	01100001	161	10100001	225	11100001
34	00100010	98	01100010	162	10100010	226	11100010
35	00100011	99	01100011	163	10100011	227	11100011
36	00100100	100	01100100	164	10100100	228	11100100
37	00100101	101	01100101	165	10100101	229	11100101
38	00100110	102	01100110	166	10100110	230	11100110
39	00100111	103	01100111	167	10100111	231	11100111
40	00101000	104	01101000	168	10101000	232	11101000
41	00101001	105	01101001	169	10101001	233	11101001
42	00101010	106	01101010	170	10101010	234	11101010
43	00101011	107	01101011	171	10101011	235	11101011
44	00101100	108	01101100	172	10101100	236	11101100
45	00101101	109	01101101	173	10101101	237	11101101
46	00101110	110	01101110	174	10101110	238	11101110
47	00101111	111	01101111	175	10101111	239	11101111
48	00110000	112	01110000	176	10110000	240	11110000
49	00110001	113	01110001	177	10110001	241	11110001
50	00110010	114	01110010	178	10110010	242	11110010
51	00110011	115	01110011	179	10110011	243	11110011
52	00110100	116	01110100	180	10110100	244	11110100
53	00110101	117	01110101	181	10110101	245	11110101
54	00110110	118	01110110	182	10110110	246	11110110
55	00110111	119	01110111	183	10110111	247	11110111
56	00111000	120	01111000	184	10111000	248	11111000
57	00111001	121	01111001	185	10111001	249	11111001
58	00111010	122	01111010	186	10111010	250	11111010
59	00111011	123	01111011	187	10111011	251	11111011
60	00111100	124	01111100	188	10111100	252	11111100
61	00111101	125	01111101	189	10111101	253	11111101
62	00111110	126	01111110	190	10111110	254	11111110
63	00111111	127	01111111	191	10111111	255	11111111