Octal and Hex Notation

```
01 #include <stdio.h>
02
03 int main() {
04    int i = 017634256034; // 01 111 110 011 100 010 101 110 000 011 100
05    int j = 0x3A9BEC7D;   // 0011 1010 1001 1011 1110 1100 0111 1101
06    unsigned short us;
07
08    printf("i is %d or %X or %o\n", i, i, i);
09    printf("j is %d or %x or %o\n", j, j, j);
10
11    printf("\nOne bit masks:\n");
12    for (us = 1; us != 0; us = us * 2)
13       printf("0x%04x (%07o)\n", us, us);
14 }
15
16 /* Sample run:
17 i is 2121358364 or 7E715C1C or 17634256034
18 j is 983297149 or 3a9bec7d or 7246766175
19
20 One bit masks:
21 0x0001 (0000001)
22 0x0002 (0000002)
23 0x0004 (0000004)
24 0x0008 (0000010)
25 0x0010 (0000020)
26 0x0020 (0000040)
27 0x0040 (0000100)
28 0x0080 (0000200)
29 0x0100 (0000400)
30 0x0200 (0001000)
31 0x0400 (0002000)
32 0x0800 (0004000)
33 0x1000 (0010000)
34 0x2000 (0020000)
35 0x4000 (0040000)
36 0x8000 (0100000)  */
```

Hex Digits:

```
0  0000   8 1000
1  0001   9 1001
2  0010   A 1010
3  0011   B 1011
4  0100   C 1100
5  0101   D 1101
6  0110   E 1110
7  0111   F 1111
```

Read this aloud: 123,456,789

Octal bit interpretation

# of 64s			# of 8s			# of 1s		
4	2	1	4	2	1	4	2	1
1	0	1,	0	1	1,	1	1	0
	5			3			6	

Hexadecimal bit interpretation

256s				16s				1s			
8	4	2	1	8	4	2	1	8	4	2	1
1	0	1	0	1	1	0	0	0	1	0	1
	A				C				5		

0011 1010 1001 1011 1110 1100 0111 1101
3 A 9 B E C 7 D

1101 1110 1010 1101 1011 1110 1110 1111
D E A D B E E F