For each statement, describe the values of k for which the code in the {...} block is executed. (10)

A)
```
if( k > 5 && k < 10 )
    {...}
```

k = 6, 7, 8, 9

B)
```
if( !k )
    {...}
```

k = 0

C)
```
if( k & 0x40 )
    {...}
```

k = if bit 6 in k, is set then a bit will be on and it will be true
Hint: a certain bit needs to be in a given state for the code to be executed.
Note the bits in a 16-bit number are (15,14,13,12, 11,10,9,8, 7,6,5,4, 3,2,1,0)

D)
```
for( k = 0; k < 10; k+=2 )
    {...}
```

k = 0, 2, 4, 6, 8

E)
```
if( k | 0x20 )
    {...}
```

k = Always, regardless of k, since the or operation forces a bit high a bit is always on

Assuming we have a variable (***16-bit***) in our program named "Pattern", write the C code to perform the following masking operations (i.e. Pattern &= 0x01;). (8)

Note the bits in a 16-bit number are (15,14,13,12, 11,10,9,8, 7,6,5,4, 3,2,1,0)

A) Toggle bit 10

```
Pattern = Pattern ^ 0x0400;
```

B) Force bits 3 and 6 to be high

```
Pattern |= 0x0048;
```

C) Force bits 2 and 12 to be low

```
Pattern &= ~(0x1004);
```

D) Force bits 2 to be high and bit 4 to be low.

```
Pattern = ( ( Pattern | 0x0004 ) & 0xffef );
```

or you could use

```
Pattern |= 0x04;
Pattern &= ~0x0010;  // Equivalent to Pattern &= 0xffef;
```

Binary Addition: Perform the following 8-bit additions, **showing all of the carry's, assuming a input carry of 0**. Also includes the interpretation of the addition as an 8-bit signed number, converted to decimal.

```
0   0   0   0   1   1   1   0   0
  0   0   0   0   1   0   1   0          10
+ 0   0   1   0   1   1   1   1        + 47
  0   0   1   1   1   0   0   1          57
```

```
0   1   0   0   1   0   0   0   0
  0   1   0   0   1   0   1   0          74
+ 0   1   0   0   1   0   0   1        + 73
  1   0   0   1   0   0   1   1        -109   OVERFLOW
"Carries into sign bit don't match"
```

```
1   1   0   0   1   0   0   0   0
  1   1   0   0   1   0   1   0         -55
+ 1   1   0   0   1   0   0   1       +(-54)
  1   0   0   1   0   0   1   1        -109
```