Solutions to Assignment 1

Question 1

The steps involved in evaluating the C expressions are shown below.

```
1.  4 * ' '
    4 * 32
    128

2.  31 / 5 + 7
    6 + 7
    13

3.  y = ( 2 + 3 ) == ( 6 - 1 )
    y = 5 == 5
    y = 1
    1

4.  x = 3 == -3 + 6
    x = 3 == 3
    x = 1
    1

5.  9 <= 4 + 9 / 2
    9 <= 4 + 4
    9 <= 8
    0

6.  ( 0xab & 0xf0 )
    0xa0

7.  ( 0x2e & 0x0f ) | ( 0x2e & 0xf0 )
    0x0e | 0x20
    0x2e

8.  7 * ( 0x0e && 0xe0 )
    7 * 1
    7

9.  ( 0x18 ^ 0xff ) + ( 2 << 1 )
    0xe7 + 4
    0xeb
```

```
10.   ~ ( 256 | '0' )
      ~ ( 0x0100 | 0x0030 )
      ~ 0x130
      0xffecf
```

(all leading digits in the result are "F")

```
11.   4 || ( ' ' == 0x21 )
      4 || 0
      1
```

(the second expression would actually be "short circuited" and would not be evaluated)

Question 2

The program will increment and print the index variable by 2 if the index is between 3 and 5 otherwise it increments it by 1. It will print the following values of i:

```
1
2
3
5
7
```

Question 3

```c
#include <stdio.h>
main()
{
   int i ;
   for ( i=5 ; i<=75 ; i = i+5 ) {
      if ( i >= 25 && i <= 65 ) {
        /* do nothing */
      } else {
        printf ( "%d\n", i ) ;
      }
   }
}
```

Question 4

```
int lowerlen( char s[] )
{
   int i, n ;
   n = 0 ;
   for ( i = 0 ; s[i] != 0 ; i++ ) {
      if ( s[i] >= 'a' && s[i] <= 'z' ) {
         n++ ;
      }
   }
   return n ;
}
```

Question 5

decimal	binary	hex
4	0100	0x4
3	0011	0x3
32	10 0000	0x20
31	1 1111	0x1f
128	1000 0000	0x80
127	0111 1111	0x7f
235	1110 1011	0xeb

Question 6

binary	hex	decimal
1101	0x0d	13
1001 1101	0x9d	157
0100 0010	0x42	66
10 1100	0x2c	44
0101 1110	0x5e	94

Question 7

hex	binary	decimal
0x0a	0000 1010	10
0xa	0000 1010	10
0xAA	1010 1010	170
0xFA	1111 1010	250
0x80	1000 0000	128
0x08	0000 1000	8