Sample Midterm

These are sample questions that are very similar to the ones I will ask on the midterm.

1. Which of the following is not a valid C variable name?
   (a) hello
   (b) chomp_burp
   (c) whilex
   (d) more-more-more
   (e) TrUcKiNg

2. Evaluate the following expressions, assuming \( a = 1 \), \( b = -3 \), and \( c = 0 \). Treat them independently, so (for example) after evaluating (b), use the above values for (c).
   (a) \( a + b < c \)
   (b) \( c == !a == c \)
   (c) \( a = +++b; \) give the values of both \( a \) and \( b \) as well
   (d) \( a-- == !b; \) give the value of \( a \) as well
   (e) \( a / b \)

3. True or False: If \( x = -1 \), then if \( x \) printf("1"); else printf("2"); prints 2.

4. Write a recursive function to add the integers from \( a \) to \( b \). Please assume that \( a \leq b \).

5. What are all possible outputs of the following code fragment?

```c
void f(int a, int b)
{
    printf("%d %d\n", a, b);
}

void main(void)
{
    int i = 5;
    f(i++, i++);
}
```

6. Given the definitions

```c
int numbs[10];
int *ptr = numbs;
```

which of the following are equivalent, and why?
   (a) numbs[3]
   (b) numbs + 3
   (c) *(numbs + 3)
   (d) *(ptr + 3)
   (e) *ptr + 3

7. Use the following code fragment to answer parts (a), (b), and (c):

```c
for(x = i = 0; i <= 100; i += 2, x += i);
```

(a) In one short sentence, what does this for loop do?

(b) Is the following while loop equivalent? If not, how does its result differ? (Hint: look at the values of both \( x \) and \( i \).)

```c
x = i = 0;
while( i++ <= 100)
x += ++i;
```

(c) Does the following for loop do the same thing? If not, what does it do?
8. What does this function do?

```c
char *x(char *s, char c)
{
    char *r = NULL;
    do{
        while(*s && *s != c)
            s++;
        if (*s)
            r = s;
    } while(*s++);
    return(r);
}
```

9. The following segment of code is supposed to print the number of times the routine `a_again` is called. Yet, regardless of the input, it prints 0. Why? How would you fix it?

```c
void a_again(int acount)
{
    ++acount;
}

int main(void)
{
    int c;
    int counter = 0;

    while((c = getchar()) != EOF)
        if (c == 'a' || c == 'A')
            a_again(counter);

    printf("%d\n", counter);
    return(0);
}
```