COMP2510 – Review Exercise 3 (Winter 2017)

1. What is the output of each of the following?

(a)
```
int a = 0, b = 0, c = -1;
c = a++ || ++b;
printf("%d %d %d", a, b, c);
```

(b)
```
int a = 1, b = 1, c = -1;
c = --a && b++;
printf("%d %d %d", a, b, c);
```

(c)
```
int n = 3;
int *p = &n;
while (++*p < 8)
  printf("%d ", *p);
```

(d)
```
int a[] = { 5, 3, 1 };
int *p = &a[1];
printf("%d ", *p--);
printf("%d", *p);
```

(e)
```
int a[] = { 5, 3, 1 };
int *p = &a[1];
printf("%d ", --*p);
printf("%d", *p);
```

(f)
```
int  a[] = { 9, 8, 7, 6, 5, 4, 3, 2, 1 };
int *p = a + 1;
while (*p++ > 5)
  printf("%d ", *p);
```

(g)
```
int  a[] = { 12, 10, 8, 6, 4, 2 };
int *p = a + 1;
while (--*p > 5)
  printf("%d ", *p);
```

(h)
```
int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
int *p;
for (p = a + 1; p < a + 8;  p += 2)
  printf("%d ", *p);
```

(i)
```
int  m = 12, n = 34;
int *p = &m;
int *q = &n;
*q = --*p;
*p = (*q)--;
printf("%d %d", *p, *q);
```

(j)
```
int  a[] = { 11, 22 };
int *p = a;
int *q = a + 1;
p++; q--;
printf("%d %d", *p, *q);
```

(k)
```
char a[] = "goodbye!";
char *p;
for (p = a; *p != 0; p += 2)
  putchar(*p);
```

(l)
```
int a[] = {11, 22, 33};
int *b[3];
int **p, **q, **r;
b[0] = a, b[1] = a+1, b[2] = a+2;
p = b, q = b+1, r = b+2;
*p = *r--;
--*q;
printf("%d %d %d", **p, **q, **r);
```

(m)
```
int a[] = {11, 22, 33};
int *p = a, *q = a + 1, *r = a + 2;
*p++ = *r--;
*++q = *r--;
--*p; ++*q; --*r;
printf("%d %d %d", a[0], a[1], a[2]);
```

(n)
```
char a[32] = "0123456789";
sscanf("hi", "%s", a + 4);
printf("%s", a + 2);
```

(o)
```
char  *argv[] = { "hello", "beautiful", "world" };
printf( "%s", *(argv + 1) );
```

(p)
```
char  *argv[] = { "hello", "beautiful", "world" };
printf( "%s", *argv + 1 );
```

2. Suppose we create a binary search tree by inserting the integers 14, 27, 10, 13, 16, 28, 9 in that order into an empty tree.

   (Recall that at each node of a binary search tree, the left subtree contains smaller values while the right subtree contains larger values than the node.)

   What is the output if we print the integers in the tree

   (a) using standard preorder traversal?

   (b) using standard inorder traversal?

   (c) using standard postorder traversal?