

1)

```
var a = 1, b = 1, c, d;
```

```
c = ++a; alert(c);      // 2
```

префиксная форма, сначала происходит инкрементирование, потом присваивание

```
d = b++; alert(d);      // 1
```

постфиксная форма, сначала происходит присваивание, потом инкрементирование

```
c = (2+ ++a); alert(c);  // 5
```

**согласно приоритетам операции: ++a = 3 (первое инкрементирование a было выше),
далее c = 2 + 3 = 5**

```
d = (2+ b++); alert(d);  // 4
```

b++=2, 2+2 = 4, d=4

```
alert(a);                // 3
```

два раза инкрементировали выше

```
alert(b);                // 3
```

два раза инкрементировали выше

2)

Чему будет равен x в примере ниже?

```
var a = 2;
```

```
var x = 1 + (a *= 2);
```

согласно приоритетам операции: $a * 2 = 4$, $x = 1 + 4 = 5$

проверим:

Подтвердите действие

5

OK

все верно))

7)

```
alert( null > 0 ); // (1) false  
alert( null == 0 ); // (2) false  
alert( null >= 0 ); // (3) true
```

Причина в том, что нестрогое равенство и сравнения `>` `<` `>=` `<=` работают по-разному. Сравнения преобразуют `null` в число, рассматривая его как `0`. Поэтому выражение (3) `null >= 0` истинно, а `null > 0` ложно.

С другой стороны, для нестрогого равенства `==` значений `undefined` и `null` действует особое правило: эти значения ни к чему не приводятся, они равны друг другу и не равны ничему другому. Поэтому (2) `null == 0` ложно.