1. data types that are smaller than integer (i.e. char, short) are automatically promoted to signed int. That is why (signed short) -1 > (unsigned short) 1 returns false. Both -1 and 1 are promoted to signed int data type, which preserves the sign.

2. If the data types (after promotion to int) in an expression are different, the values are promoted to the data type that can represent the largest number. Some examples:

   signed long x > unsigned long y
   x is promoted to unsigned long because unsigned long can represent $2^{64} - 1$, while signed long can represent $2^{63} - 1$. $2^{64} - 1 > 2^{63} - 1$.

   unsigned int x > signed long y
   x is promoted to signed long because $2^{63} - 1 > 2^{32} - 1$.

3. (similar to 2) If one of the values in expression is long double, then all values are promoted to long double. Otherwise, if one of the values is double, then all values are promoted to double. Otherwise, if one of the values is float, then all values are promoted to float. These data types can represent VERY large numbers, and considering highest values that can be represented: long double > double > float.

   This article was helpful to me:
   http://icecube.wisc.edu/~dglo/c_class/promo_conv.html