```
#include <stdio.h>

int main()
{
   short a, b, c, d, e ;
   unsigned short x, y ;

/*   Some strange things happen on a computer when we exceed     */
/* the numerical boundaries imposed by allocated size limits.    */
/*
/* In the C programming language (this language), the data type  */
/* 'short' is 16 bits.  The largest possible integer using 16    */
/* bits (using two's complement) is 2^15 - 1 or  32767.  The     */
/* smallest possible number is -2^15 = -32768.                   */

/* The following program explores the edges of these boundaries. */

   a = 32767 ;
   b = a + 1 ;

   printf("a = 32767 ;\n") ;       /* 'printf' is the print function in C  */
   printf("b = a + 1 ;\n") ;
   printf( "Now: a = %hd, b = %hd\n", a, b ) ;

   printf("\n") ;

   c = b - 1 ;
   printf("c = b - 1 ;\n") ;
   printf( "Now: b = %hd, c = %hd\n", b, c ) ;

   printf("\n") ;

   d = -b ;
   printf("d = -b ;\n") ;
   printf( "Now: b = %hd, d = %hd\n", b, d ) ;

   printf("\n") ;

   x = (unsigned) b ;
   printf( "x = (unsigned) b ;\n") ;
   printf( "Now: b = %hx, x = %hx\n", b, x )  ;
   printf( "Now: b = %hd, x = %hu\n", b, x )  ;

   printf("\n") ;

   e = -1 ;
   y = (unsigned) e ;
   printf( "e = -1 ;\n") ;
   printf( "y = (unsigned) e ;\n") ;
   printf( "Now: e = %hx, y = %hx\n", e, y )  ;
   printf( "Now: e = %hd, y = %hu\n", e, y )  ;
}
```

```
----------------------  Sample Run ----------------------
sunset% gcc twos.c
sunset% a.out
a = 32767 ;
b = a + 1 ;
Now: a = 32767, b = -32768

c = b - 1 ;
Now: b = -32768, c = 32767

d = -b ;
Now: b = -32768, d = -32768

x = (unsigned) b ;
Now: b = 8000, x = 8000
Now: b = -32768, x = 32768

e = -1 ;
y = (unsigned) e ;
Now: e = ffff, y = ffff
Now: e = -1, y = 65535
```