1) a função dump imprime o valor em hexadecimal de cada byte do endereço de memoria que ela recebe, então:

Dump de c: como c é um char (1 byte) com valor 150, dump de c deve imprimir o valor correspondente em hexadecimal, que pela tabela ascii é 96.

dump de s: short tem 2 bytes (16 bits)

3 em bin = 0000 0000 0000 0011 -> comp a 2:

1111 1111 1111 1100

+1

= 1111 1111 1111 1101 -> em hexadecimal = f f f d

Vai imprimir fd ff

Dump de i: int tem 4 bytes (32 bits)

Int 151 em bin = 0000 0000 0000 0000 0000 0000 1001 0111 -> comp a 2:

-151 = 1111 1111 1111 1111 1111 1111 0110 1000

+1

= 1111 1111 1111 1111 1111 1111 0110 1001 -> em hexadecimal: f f f f f f 6 9

Vai imprimir 69 ff ff ff

2) dump de l: short tem 2 bytes (16 bits)

Short 32765 em bin = 0111 1111 1111 1101 -> comp a 2:

-32765 = 1000 0000 0000 0010 + 1 = 1000 0000 0000 0011

Em hexadecimal = 8 0 0 3 -> vai imprimir 03 80

Dump de k: short tem 2 bytes (16 bits)

Unsigned short 32771 em bin = 1000 0000 0000 0011

Em hexadecimal = 8 0 0 3 -> vai imprimir 03 80

3) na implementação da função xbyte, criei a variável int i e armazenei nela um valor cujo byte menos significativo é o byte indicado de word e os outros bytes valem 0. Em seguida, usei o condicional para verificar se o bit 7 de i é 1 (porque se for, então seu valor signed é negativo); se sim, então a rep2(i) = 2^8 + i, portanto o valor com sinal de i é -2^8 – i. Se não, i > 0 e seu valor é próprio valor unsigned.

4) No programa 1, x foi declarado como um int, e o valor inteiro com sinal de 0xffffffff é -1. No programa 2, x foi declarado como unsigned int. A representação de 0xffffffff como um inteiro sem sinal é 2^32 – 1, que é maior do que 2.

No programa 3, apesar de x ser declarado como inteiro com sinal, y foi declarado como inteiro sem sinal e em expressões que envolvem int com e sem sinal todos são tratados como sem sinal. Então a expressão x < y considera o valor unsigned de x, que é maior que y.

5) a representação de -1 em 8 bits é 1111 1111. Quando acontece a atribuição de um signed char para um unsigned int, a representação passa a ser em 32 bits; nesse caso, a representação em 32 bits de -1 é 1111 1111 1111 1111 1111 1111 1111 1111, porque sc era um valor signed (então ao invés de adicionar 0s a esquerda, adiciona cópias do bit mais significativo, que nesse caso era 1). Portanto a representação interna de ui é 0xffffffff (em hex).