1. Perform the following number conversions:
   (a) \(0xABC0F1\) to binary
   (b) Binary \(1100000111001111\) to hexadecimal
   (c) Binary \(1100000111001111\) to octal (base 8)

2. Convert between decimal and hexadecimal:
   (a) 147 to hexadecimal
   (b) \(0xAE\) to decimal

3. Solve directly in hexadecimal: \(0xB75D + 0x8AF\)

4. Perform the following operations on \(x = 0xA5\) and \(y = 0x2C\) (answer in hex):
   (a) \(x & y\)
   (b) \(x && y\)
   (c) \(\neg x \& !\left(y \mid \left(x ^ y\right)\right)\)

5. Perform the following shift operations on the byte \(x = 0xB9\) (answer in hex):
   (a) \(x \ll 3\)
   (b) \(x \gg 1\), logical
   (c) \(x \gg 3\), arithmetic
6. Write out and sum the non-zero powers of two for the two functions shown. Express output in decimal:

(a) \( B2U_5(0x2E) \)

(b) \( B2T_5(0x2E) \)

7. Apply the function \( T2U_5 \) to the two following decimal values. Express the answer as a decimal number.

(a) -7

(b) 12

8. Assuming an 8-bit machine that uses 2’s complement arithmetic, apply the appropriate casting and express the result as a relational value (true or false):

(a) \(-127 == 127U\)

(b) \(-127-1U == 127\)

(c) \(-1U < 128U\)

(d) \(255U == -128\)

9. Express \( x*K \) using on only the specified number of operations:

(a) \( K = 7 \), using 1 shift and 1 Add/Sub

(b) \( K = -13 \), using the fewest shifts and Add/Subs

10. Write C expressions that evaluate to 1 when the following conditions are true and to 0 when they are false. Assume \( x \) is of type int.

(a) Any bit of \( x \) equals 1.

(b) Any bit of \( x \) equals 0.

(c) Any bit in the least significant byte of \( x \) equals 1.