

PES, Section 2.7
Bit access

1. What is the output value (in hexadecimal) of the following sequence of operations?

```
unsigned char x = 0x33;           // 0011 0011
x = SetBit(x, 1, 0);              // 0011 0001
x = x << 2;                        // 1100 0100
x = SetBit(x, 4, 1);              // 1101 0100
x = SetBit(x, 3, 0);              // 1101 0100
x = x >> 1;                        // 0110 1010
x = SetBit(x, 2, 1);              // 0110 1110
```

2. What is the value of B after executing each of the following two statements? Assume that the input provided to A is the constant value 10, and that the value does not change.

```
B = (a == 1) ? 20 : 30;           B = 30
```

```
B = (a == 10) ? 20 : 30;         B = 20
```

3. Write a RIMS-compatible C-language for-loop that sets B to the reverse of A, in other words B7 = A0, B6 = A1, etc.

```
unsigned char i;
unsigned char tmp = 0;
for(i = 0; i < 8; i++)
    tmp = SetBit(tmp, i, GetBit(A, 7-i));
B = tmp;
```

4. Write a RIMS-compatible C-language for-loop that counts the number of times a bit of A is **followed** by a bit of the opposite parity (01 or 10) and writes the value to B. For example 00100110 has 4 cases: 00100110, 00100110, 00100110, 00100110.

```
unsigned char i;
unsigned char cnt = 0;
for(i = 0; i < 7; i++)
    cnt = (GetBit(A, i) ^ GetBit(A, i+1)) ? cnt+1 : cnt;
B = cnt;
```

Note: Be careful with the loop bounds for counter i. It would be an error to call GetBit(A, -1) or GetBit(A, 8).