

1 (5 marks) Memory and Numbers. Consider the execution of the following code. Assume that the compiler has statically allocated **a** to start at address 0x5080, and consecutively allocates the subsequent variables, inserting padding (wasting memory) only where needed to ensure that each variable is aligned. Assume too that char's are signed, and int's and pointers are 4 bytes long.

```

int a;
short b;
char* c;

int calc(int key) {
    a = 0xdecafbad;
    b = (key & 0xffff);
    c = (char *) (key >> 16);
    return (*c) & 0xffff;
}

```

1a Assuming that the code above executes on a **little-endian** machine, give the value in hex for each expression below under the “LE” column, or write UNKNOWN if the value cannot be determined completely. Repeat for a big-endian machine in the “BE” column. Assume that each computation is independent (i.e. all memory is reset to an unknown state before each expression).

```

calc(0x5080fade) :
LE: key: _____, a: _____, b: _____, c: _____, result: _____
BE: key: _____, a: _____, b: _____, c: _____, result: _____

calc(0x5082beef) :
LE: key: _____, a: _____, b: _____, c: _____, result: _____
BE: key: _____, a: _____, b: _____, c: _____, result: _____

calc(0x5084f00d) :
LE: key: _____, a: _____, b: _____, c: _____, result: _____
BE: key: _____, a: _____, b: _____, c: _____, result: _____

calc(0x5086b0ba) :
LE: key: _____, a: _____, b: _____, c: _____, result: _____
BE: key: _____, a: _____, b: _____, c: _____, result: _____

calc(0x5088deed) :
LE: key: _____, a: _____, b: _____, c: _____, result: _____
BE: key: _____, a: _____, b: _____, c: _____, result: _____

```