Christopher Martin, Jennifer Wolfe, Geoffrey Corey
May 1, 2013
CS 472 Lab 3

Part 1

```
#include <stdio.h>
#include <endian.h>

int main(int argc, char **argv)
{
        short val;
        char *p_val;
        p_val = (char *) &val;
        /*
          The following two lines assume big-endian
          Architecture to initialize the variable Val
          to 0x1234.
        */
#if _BYTE_ORDER == _BIG_ENDIAN
        p_val[1] = 0x12;
        p_val[0] = 0x34;
#else
        p_val[1] = 0x34;
        p_val[0] = 0x12;
#endif

        printf("%x\n", val);

        return 0;
}
```

Part 2

```
#include <stdio.h>
#include <stdint.h>

int is_big_endian(void);

int main(int argc, char **argv)
{
        short val;
        char *p_val;
        p_val = (char *) &val;
        /*
          The following two lines assume big-endian
          Architecture to initialize the variable Val
          to 0x1234.
        */

        if(!is_big_endian){
                p_val[0] = 0x12;
                p_val[1] = 0x34;
        } else {
                p_val[0] = 0x34;
                p_val[1] = 0x12;
        }

        printf("%x\n", val);

        return 0;
}

int is_big_endian(void)
{
    union {
        uint32_t i;
        char c[4];
    } bint = {0x01020304};

    return bint.c[0] == 1;
}
```