1. Reverse the order of the bytes within a 32-bit C variable named x32

```
asm
(
        “REV %[reg1], %[reg2]”
        : [reg1] “=r” (x32)
        : [reg2] “r” (x32)
);
```

2. Rotate the contents of a 64-bit Variable named x64 right by 1 bit position

```
asm
(
        “ROR %[reg1], %[reg2], %[const]”
        : [reg1] “=r” (x64)
        : [reg2] “r” (x64)
        : [const] “i” (1)
)
```

4.

a.

```
static inline uint32_t BFC(uint32_t src, uint32_t lsb, uint32_t width)
{
        asm
        (
                “BFC %[reg1], %[const1], %[const2]”
                : [reg1] “+r” (src)
                : [const1] “i” (lsb)
                : [const2] “i” (len)
        );
        return src;
}
```

b.

```
static inline uint32_t UBFX(uint32_t src, uint32_t lsb, uint32_t width)
{
        uint32_t result;
        asm
        (
                “UBFX %[reg1], %[reg2], %[const1], %[const2]”
                : [reg1] “=r” (result)
                : [reg2] “r” (word)
                : [const1] “i” (lsb)
                : [const2] “i” (len)
        );
        return result;
}
```

c.

```
static inline int32_t SBFX(uint32_t src, uint32_t lsb, uint32_t width)
{
        uint32_t result;
        asm
        (
                "SBFX %[reg1], %[reg2], %[const1], %[const2]"
                : [reg1] "=r" (result)
                : [reg2] "r" (word)
                : [const1] "i" (lsb)
                : [const2] "i" (len)
        );
        return result;
}
```

d.

```
static inline uint32_t BFI(unit32_t dst, uint32_t src, uint32_t lsb, uint32_t width)
{
        asm
        (
                "BFI %[reg1], %[reg2], %[const1], %[const2]"
                : [reg1] "+r" (dst)
                : [reg2] "r" (src)
                : [const1] "i" (lsb)
                : [const2] "i" (len)
        );
        return dst;
}
```