

## start Should be a Multiple of 12

```
01F8:                                E002 F022 F025 000A
0204: 0057 0065 006C 0063 006F 006D 0065 0020 0074 006F 0020 0074
0210: 0068 0065 0020 004C 0043 002D 0033 0020 0073 0069 006D 0075
021C: 006C 0061 0074 006F 0072
```

5. How do you update between iterations  
(what is “update”)?

```
start = start + 12
```

We need **start** to be  
first address for next line,  
and a multiple of 12.

## Our Function So Far...

```
void dump_memory (int addr_s, int addr_e)
{
    int start;
    if (addr_s >= addr_e) {
        addr_e += 0x10000;
    }
    for (start = (addr_s / 12) * 12;
         start < addr_e; start = start + 12) {
        // print one row
    }
}
```

We need another iteration.

## Two Steps for Printing a Row

```
01F8:                                E002 F022 F025 000A
0204: 0057 0065 006C 0063 006F 006D 0065 0020 0074 006F 0020 0074
0210: 0068 0065 0020 004C 0043 002D 0033 0020 0073 0069 006D 0075
021C: 006C 0061 0074 006F 0072
```

Is printing a row just an iteration?

No! It's a sequence:

- **print the address**, then
- **print contents** of 12 memory locations.

## How Do We Print the Row Address?

Where is the address for the row?

```
start
```

How can we print the address?

```
printf ("%04X: ", start & 0xFFFF);
```

Why mask out the high bits?

Remember that **start** may point  
into the virtual copy (**start > 0x10000**).