Exercise 1

1. 2-byte quantities: 8FE0 + B036

$$\begin{array}{ccccc} \text{c} & & \text{c} & & \\ & 8 & F & E & 0 \\ + & B & 0 & 3 & 6 \\ \hline 1 & 4 & 0 & 1 & 6 \end{array}$$

- The carry bit is set to CF = 1.
- The overflow bit is set to OF = 1
- Since both 8FE9 and B036 are both negative the expected summation outcome is negative.
- However 4016 is positive number therefore, the overflow bit is set.
- The signed extend representation: 0000 4016
- Hexadecimal to Decimal Conversion

$$\begin{aligned} 4 \cdot 16^3 + 1 \cdot 16^1 + 6 \\ &= 16406 \end{aligned}$$

Algorithm 1 macro prints

```
mov ax, 03076h ; ax : 03076h
movsx eax, ax ; extend to a signed 4 byte value 0000 4016
call print_int ; prints 16406 to the screen
```

2. 1-byte: E5 + 0E

$$\begin{array}{ccc} & \text{c} & \\ & E & 5 \\ + & 0 & E \\ \hline & F & 3 \end{array}$$

- The carry bit is set to CF = 0
- The overflow bit is set to OF = 0.
- E5 is negative and 0E is positive then there is no expectation of overflow. Also E5 is small negative and 0E is a small positive so the summation should also be in the range.
- Hexadecimal to Decimal Conversion
  F3 is a negative so take the complement

$$\begin{aligned} FF - F3 &= \\ 0C + 1 &= \\ 0D \end{aligned}$$

Then convert it

$$\begin{aligned} 0 \cdot 16^1 + 13 \\ = 13 \end{aligned}$$

Negate the outcome so -13.

- Signed-Extended: FFFF FFF3

Algorithm 2 macro prints

```
mov al, 0F3h ; points ax to F3
movsx eax, al ; extends F3 to a byte signed FFFF FFF3
call print_int ; prints -13
```

3. 2-byte quantities: 5243 + 7DBC

$$\begin{array}{ccccc} & 5 & 2 & 4 & 3 \\ + & 7 & D & B & C \\ \hline & C & F & F & F \end{array}$$

- The carry bit is set to CF = 0
- The overflow bit is set to OF = 1
- The reason is because since 5243 and 7DBC are both positive, the expecation of the summation is positive. However, CFFF is a negative causing the overflow.
- Converting Hexadecimal to Decimal
  Since CFFF is negative, take the negation of it by finding the two's complementary

$$\begin{aligned} FFFF - CFFF = \\ = 3000 + 1 = 3001 \\ \rightarrow \\ 3 \cdot 16^3 + 1 = \\ 12289 \\ \text{negate it} - 12289 \end{aligned}$$

- sign extended: FFFF CFFF

Algorithm 3 macro prints

```
mov ax, 0CFh ; move al to CF
movsz eax, ax ; 4 byte sign extention of eax : FFFF CFFF
call print_int ; prints -12289
```

4. 1-byte quantities: E5 + 3A

$$\begin{array}{ccc} \mathrm{c} & & \\ & \mathrm{E} & 5 \\ + & 3 & \mathrm{A} \\ \hline 1 & 1 & \mathrm{F} \end{array}$$

- Set the Carry Flow into CF = 1
- Set the Over Flow into OF = 0
- Carry Flow because outcome is larger than a 1 byte
- Overflow is 0 because a negative (E5) and positive (3A) summation does not lead to an overflow.
- Sign Extended: 0000 001F
- Since 1F is positive then

$$\begin{aligned} 1 \cdot 16 + 15 = \\ 31 \end{aligned}$$

Algorithm 4 macro prints

```
mov al, 01Fh ; point al to 1F
movsz eax, al ; eax : 4 byte sign extention 0000 001F
call print_int ; prints 31
```