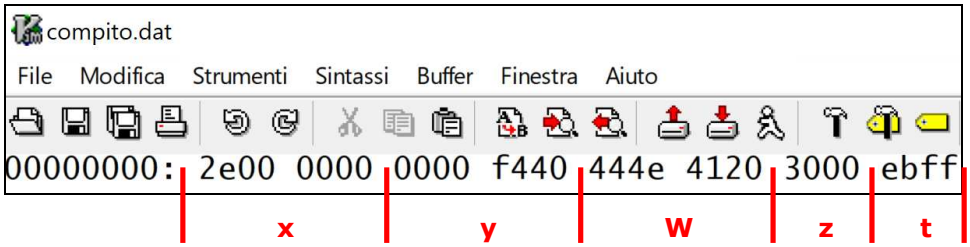
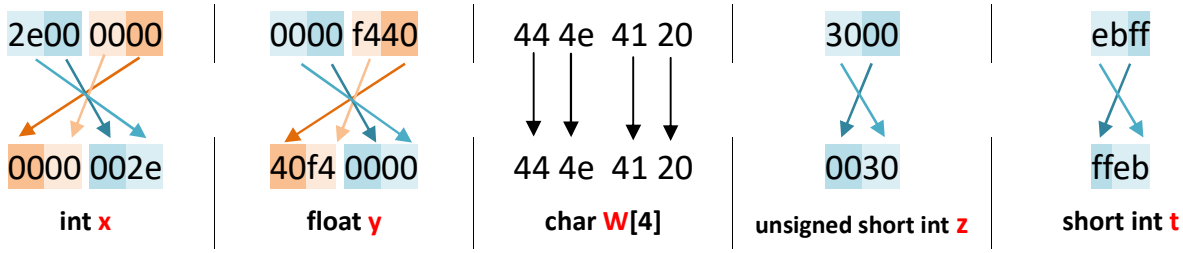
Il file binary **"compito.dat"** contiene un record da 16 byte il cui tracciato è il seguente:

```
struct Rec {
                    int x;          //--- 4 byte
                  float y;          //--- 4 byte
                   char W[4];       //--- 4 byte
   unsigned short int z;            //--- 2 byte
            short int t;            //--- 2 byte
};
```

Il contenuto del file visto con VIM (opzione Converti a esadecimale) è il seguente:




In basso, nella prima riga è riportato il contenuto del file visto con VIM; nel file tutti i **campi numerici multibyte** (int, float, unsigned short int e short int) hanno ordine di memorizzazione ***little endian*** - ordine utilizzato dai processori Intel per la memorizzazione dei dati numerici: si inizia dal byte meno significativo (esempio: 2e per il campo **x**) per finire con il byte più significativo (esempio: 00 per **x**).

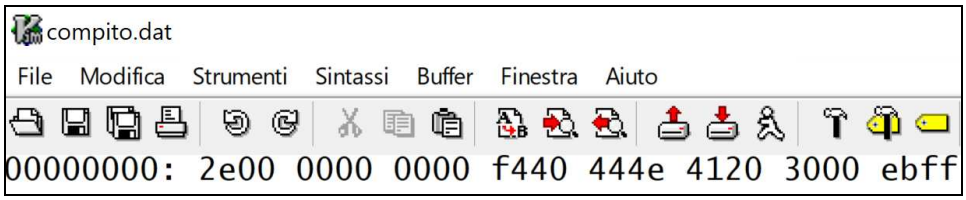



Nella seconda riga sono riportati i **valori dei campi espressi in esadecimale** ottenuti operando l'**inversione dei campi numerici multi-byte**.

*ATTENZIONE! ogni byte è rappresentato da 2 cifre esadecimali*

( per una descrizione più estesa degli ordini di memorizzazione: cfr https://it.wikipedia.org/wiki/Ordine_dei_byte )

**Sapendo che la codifica ASCII del carattere 'A' è $41_H$ (in esadecimale), quali valori sono memorizzati nel record qui riportato?**




**int x** 2e00 0000 (memorizzazione di **x** *little endian*)

**Inversione del campo numerico multi-byte:**
0000 002e (codifica di **x** in esadecimale)

**Conversione da esadecimale a binario:**
$0000002E_{16}$ = $0000000000000010111_0_2$
il primo bit è 0 quindi il numero è POSITIVO, si procede direttamente con la

**Conversione da binario a decimale:**
$101110_2 = 2 + 4 + 8 + 32 = 46_{10}$

***NOTA:*** *si può anche convertire in decimale direttamente dall'esadecimale*
$2E_{16} = 14x16^0 + 2x16^1 = 14x1 + 2x16 = 14 + 32 = 46_{10}$

**x = +46**

---

**float y** 0000 f440 (memorizzazione di **y** *little endian*)

**Inversione del campo numerico multi-byte:**
40f4 0000 (codifica IEEE-P574-SP di **y** espressa in esadecimale)

**Conversione da esadecimale a binario:**
4 0 F 4 0 0 0 $0_{16}$ = 0100 0000 1111 0100 0000 0000 0000 $0000_2$

il primo bit è 0 quindi **y** è POSITIVO

**Calcolo dell'esponente:**
100 0000 1 = 1 + 128 = 129 (in eccesso 127)
129 -127 = +2 (valore dell'esponente per la notazione in virgola mobile normalizzata del numero **y**)

la **mantissa** è **1,**$1110100\ldots_2$ (va inserito 1 prima della virgola, perché NON è memorizzato in IEEE-P574-SP)

**y** = $1{,}11101_2E+2 = 111{,}101_2$ (esponente +2 della base 2 → la virgola si sposta di 2 posti verso destra) =
= $1x2^2 + 1x2^1 + 1x2^0 + 1x2^{-1} + 1x2^{-3} = 4 + 2 + 1 + 1/2 + 1/8 = 7 + 0{,}5 + 0{,}125 = 7{,}625_{10}$

**y = +7,625**

**char W[4]** **i 4 byte contengono:** 44 4E 41 20 (ogni byte è espresso da 2 cifre esadecimali)

**Costruzione della tabella ASCII a partire dal carattere 'A' la cui codifica espressa in esadecimale è $41_{16}$:**

| carattere | codif. esadecimale | carattere | codif. esadecimale | carattere | codif. esadecimale |
|---|---|---|---|---|---|
| **A** | **41** | F | 46 | K | 4B |
| B | 42 | G | 47 | L | 4C |
| C | 43 | H | 48 | M | 4D |
| **D** | **44** | I | 49 | **N** | **4E** |
| E | 45 | J | 4A | | |

**Decodifica di W:**
W[0] = $44_h$ = 'D'
W[1] = $4E_h$ = 'N'
W[2] = $41_h$ = 'A'
W[3] = $20_h$ = ' ' (spazio)

**W = "DNA "**

---

**unsigned short int z** 3000 (memorizzazione di **z** *little endian*)

**Inversione del campo numerico multi-byte:**
0030 (valore di **z** in esadecimale)
**Conversione da esadecimale a binario:**
$0030_{16}$ = 00000000 **0011**0000$_2$

**Conversione da binario a decimale** (direttamente perché il numero è senza segno)**:**
**11**0000$_2$ = 16 + 32 = 48

**z = 48**

---

**short int t** ebff (memorizzazione di **t** *little endian*)

**Inversione del campo numerico multi-byte:**
ffeb (valore di **t** in esadecimale)
**Conversione da esadecimale a binario:**
$FFEB_{16}$ = **1**111 1111 **1110** 1011$_2$
il primo bit è **1** quindi **t** è un valore negativo e va determinato il suo **opposto**

**Rappresentazione in CA2 dell'opposto di t:**
0000 0000 **0001** 0101$_2$

**Conversione da binario a decimale** (per determinare il valore del numero positivo, opposto di **t**):
**1**0101$_2$ = 1 + 4 + 16 = $21_{10}$

**t = -21**