

```
#include <iostream>
using namespace std;
#include <fstream>
#define SP ' '
struct Riga {
    float          x;           //--- 4 + 4 + 8 = 16 byte ogni record
    char           lett[4];     //--- 4 byte (= 32 bit)
    double         YY;         //--- 8 byte (= 64 bit)
};
Riga rec;

void scriviFiles(){
    fFDtxt<< rec.x << SP << rec.lett[0];
    fFDtxt<< rec.lett[1] << rec.lett[2] << rec.lett[3]<< SP;
    fFDtxt<< rec.yy << endl;
    fFDdat.write( (char *) &rec, sizeof(rec) );
}

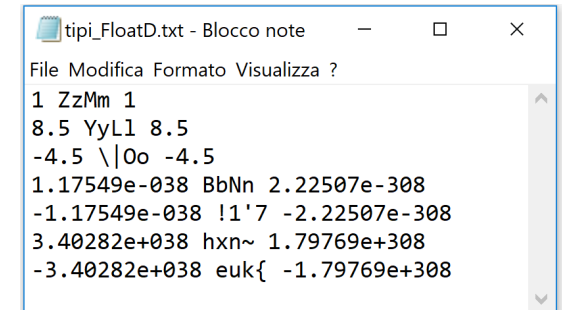
int main(){
    fstream fFDtxt, fFDdat;
    fFDtxt.open("tipi_FloatD.txt", ios::out);
    fFDdat.open("tipi_FloatD.dat", ios::out | ios::binary);
//-----
    rec.x = 1.0;           //--- RIGA/Record 1
    rec.yy = 1.0;
    rec.lett[0] = 0x5A;   //--- codifica ASCII di 'Z' espressa in esadecimale
    rec.lett[1] = 0x7A;   //--- codifica ASCII di 'z' espressa in esadecimale
    rec.lett[2] = 0x4D;   //--- codifica ASCII di 'M' espressa in esadecimale
    rec.lett[3] = 0x6D;   //--- codifica ASCII di 'm' espressa in esadecimale
    scriviFiles();
//-----
    rec.x = 8.5;          //--- RIGA/Record 2
    rec.yy = 8.5;
    for (int k=0; k<4; k++)
        rec.lett[k]--;    //--- codifica ASCII decrementata di 1
    scriviFiles();
//-----
    rec.x = -4.5;         //--- RIGA/Record 3
    rec.yy = -4.5;
    for (int k=0; k<4; k++)
        rec.lett[k] += 3; //--- codifica ASCII incrementata di 3
    scriviFiles();
```

Il programma **tipi\_FloatD.cpp** scrive sul **file di testo** "**tipi\_FloatD.txt**" (nome logico **fFDtxt**) 7 righe (separandole con **endl**) in ognuna delle quali scrive 6 dati: uno di tipo **float**, 4 caratteri e poi uno di tipo **double**.

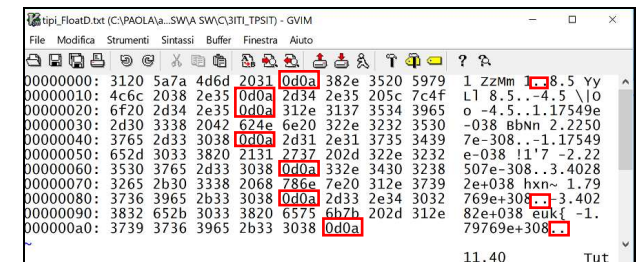
Sulle prima riga scrive il valore 1.0 come **float** (sulla seconda riga 8.5 e sulla terza -4.5), seguito da quattro **char** il cui codice ASCII è espresso in esadecimale (sulla 2^ il codice è decrementato, sulla 3^ incrementato), ultimo dato ancora 1.0 come **double** (poi 8.5 e -4.5).

Sulle ultime 4 righe vengono scritti i **valori massimi e minimi per il tipo float e double**, mentre i char vengono inizializzati di nuovo e poi trattati come degli interi mediante gli operatori binari (gli operatori aritmetici a 2 operandi: **+=**, **-=**, **\*=**, **/=**).

Il file di testo aperto con **Blocco Note** → dopo ogni riga c'è CRLF = newline (**endl** del C++) **0D0Ahex**



Il file di testo aperto con **Vim** → selezionando successivamente **Strumenti > Converti a esadecimale**



```

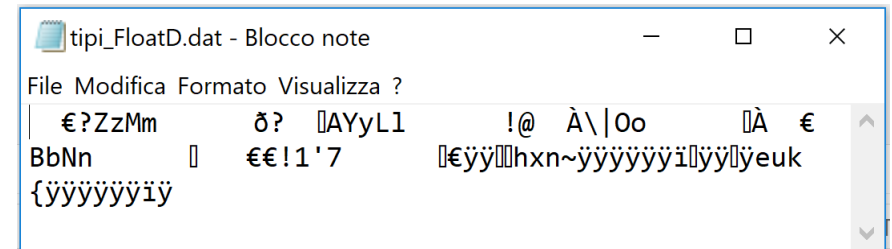
rec.x = 1.17549435e-38; //--- RIGA/Record 4
rec.yy = 2.2250738585072014e-308;
rec.lett[0] = 0x42;
rec.lett[1] = 0x62;
rec.lett[2] = 0x4E;
rec.lett[3] = 'n';
scriviFiles();
//-----
rec.x = -1.17549435e-38; //--- RIGA/Record 5
rec.yy = -2.2250738585072014e-308;
for (int k=0; k<4; k++)
    rec.lett[k] /= 2;
scriviFiles();
//-----
rec.x = 3.40282347e+38; //--- RIGA/Record 6
rec.yy = 1.7976931348623157e+308;
for (int k=0; k<4; k++)
    rec.lett[k] += 71;
scriviFiles();
//-----
rec.x = -3.40282347e+38; //--- RIGA/Record 7
rec.yy = -1.7976931348623157e+308;
for (int k=0; k<4; k++)
    rec.lett[k] -= 3;
scriviFiles();

fFDtxt.close();
fFDdat.close();
return 0;
}
    
```

Anche i dati *float* e *double* vengono memorizzati dai **processori Intel** con il sistema **little endian** (prima il byte meno significativo per finire con quello più significativo), quindi il numero **float 1.0** la cui rappresentazione in floating point nello standard IEEE-P754 è **00111111100000000000000000000000** in esadecimale **0x3F800000**, è memorizzato invertendo i byte: **0000 803F**.  
 Notare la codifica dei numeri **opposti** (sia i minimi che i massimi): differiscono solo per il 1<sup>^</sup> bit che rappresenta il segno: **7F** (1<sup>^</sup>byte) per il positivo, **FF** per il negativo nei record 6 e 7, **00** per il positivo e **80** per il neg. nel 4<sup>^</sup> e 5<sup>^</sup> rec.

Per ogni riga scritta sul file di testo viene scritta con il metodo **write** un record (struttura **rec**) sul **file binary** "**tipi\_FloatD.dat**" (nome logico **fFDdat**). Aprendo il file binary con **Blocco Note** vengono esposti prevalentemente **caratteri non stampabili**, ad eccezione dei caratteri corrispondenti all'array `lett[4]` :

ZzMm YyLl \Oo BbNn !1'7 hxn~ euk{



Aprendo il file di testo con l'editor **Vim** e selezionando *Strumenti > Converti a esadecimale*, ogni **record rec** di tipo **Riga** occupa esattamente una riga **Vim** (16 byte) e **sono consecutivi** (solo a fine file è presente CRLF= **0D0A<sub>hex</sub>**)

