

Cognome □□□□□□□□□□□□□□□□ Nome □□□□□□□□□□□□□□□□

Esercizio 1

Dato il numero reale r , la cui rappresentazione in virgola mobile $R = \{s, F, E\}$ sia: $s=1$, $F=01000$ (mantissa su 5 bit) ed $E=010$ (esponente su 3 bit), trovare la rappresentazione del numero reale $r'=2*r$.

$R' = \{s' = \square, F' = \square\square\square\square, E' = \square\square\}$

Nello spazio sottostante riportare i passaggi più significativi della soluzione

Esercizio 2 Dire costa stampa a video il seguente programma assembler, nell'ipotesi che la stringa "cifre" contenga le tre ultime cifre del proprio numero di matricola, modulo 3.

Esempio: se il proprio numero di matricola termina per "846", "XYZ" deve valere "210"

Prima di procedere lo studente scriva qui sotto il valore di X, Y e Z per il proprio numero di matricola:

X= __, Y= __ e Z= __

```

cifre:  .ASCII "XYZ"           # immaginare che al posto di XYZ ci siano le proprie cifre
peso:   .BYTE  0x9, 0x3, 0x1

_main:  MOV    $cifre, %EDI
        MOV    $peso,  %ESI
        MOV    $3,   %DH
        MOV    $0,   %CH
lab:    MOV    (%EDI), %AL
        SUB    $'0', %AL
        MOV    (%ESI), %CL
        MUL   %CL
        ADD   %AL, %CH
        INC   %EDI
        INC   %ESI
        DEC   %DH
        JNZ   lab
        MOV   %CH, %AL
        CALL  outbyte
        RET

.include "utility"
    
```

PROMEMORIA

SUB *srg, dst* modifica il registro *dst* sottraendo al valore precedente ivi contenuto quanto contenuto nel registro *srg*.

MUL *srg* Moltiplica il contenuto del registro sorgente *srg* per AL. Il risultato viene memorizzato nel registro AX.
NB: Nel caso *srg* ed AL contengano valori numerici relativamente piccoli, la parte significativa del risultato riesce ad essere contenuta nella sola parte AL (il che equivale a dire che in questi casi AH vale zero)

JNZ *lab* si salta a *lab* finchè il flag di zero è falso. Non appena l'ultima operazione ha prodotto un risultato uguale a zero si smette di effettuare il salto.

CALL *outbyte* stampa a video la coppia di caratteri ASCII associati alla parte alta e bassa del naturale contenuto nei 4 bit più significativi e meno significativi di AL, rispettivamente.
Esempio: qualora in AL vi fosse 0011-1100, stamperebbe a video i caratteri "3C"

Uscita del programma assembler: _____

Nello spazio sottostante riportare i passaggi più significativi della soluzione

Soluzione

Soluzione Esercizio 1

r' avrà la stessa rappresentazione di r , ma con l'esponente incrementato di uno. Pertanto $E' = 011$.
Complessivamente la rappresentazione cercata è:

$$\mathbf{R'} = \{ \mathbf{s'}=1, \mathbf{F'}=01000, \mathbf{E'}=011} \}$$

Controprova: decodificando \mathbf{R} si scopre che $r = 0.625$, mentre decodificando $\mathbf{R'}$ si scopre che r' è 1.25, laddove 1.25 è effettivamente il doppio di 0.625.

Soluzione Esercizio 2

Esaminiamo il caso in cui il proprio numero di matricola termini per 846. I valori per XYZ saranno 210. Dunque tutto avviene come se la prima riga del programma assembler contiene:

```
cifre: .ASCII "210"
```

Il programma assembler legge il codice ASCII della prima cifra, in questo caso '2', e gli sottrae il codice ASCII di zero. Dunque in AL rimane il naturale associato alla cifra (2 in questo caso). Questa prima cifra viene moltiplicata per il primo elemento del vettore dei pesi (9), ottenendo, in questo caso, $9*2 = 18$, che viene salvato nel registro CL. Dopodiché il programma repete l'operazione, leggendo la cifra successiva (1) e moltiplicandola per il peso successivo (3), ottenendo 3, che viene sommato a CL, che conterrà 21. Lo stesso procedimento viene ripetuto un'ultima volta: $0*1 = 0$. Alla fine in CL si troverà il naturale 21 (che corrisponde al valore decimale associato al numero 210 espresso in base 3). Questo numero verrà alla fine del programma mostrato a video, ma in base 16. Concludendo, il programma, nell'esempio specifico, mostrerà a video la stringa "15".