

Matricola

Esercizi su Numeri e Assembler – 17 Febbraio 2017

Cognome **Nome**

Esercizio 1

Dato il numero reale $r = 0.3125$, trovarne la rappresentazione binaria R in virgola fissa, usando 6 bit per la parte decimale.

$R = 0.$

Nello spazio sottostante riportare i passaggi più significativi della soluzione

Esercizio 2 Siano I, J, H e K le quattro cifre meno significative del proprio numero di matricola (in particolare sia K quella meno significativa di tutte). Dire cosa stampa a video il seguente programma assembler:

```
vett:  .BYTE  0x0I, 0x0J, 0x0H, 0x0K

_main:  MOV    $0, %AL
        MOV    $4, %CH
        MOV    $vett, %EBX
loop:   MOV    (%EBX), %DH
        ADD    %DH, %AL
        INC    %EBX
        DEC    %CH
        JNZ    loop
        OR     $0xF8, %AL
        CMP    $0xF8, %AL
        JE     label
        MOV    $'n', %AL
        CALL   output
        RET
label:  MOV    $'s', %AL
        CALL   output
        RET

.INCLUDE "utility"
```

PROMEMORIA

<code>ADD <i>srg,dst</i></code>	modifica <i>dst</i> sommandovi <i>srg</i>
<code>JNZ <i>label</i></code>	salta a <i>label</i> nel caso in cui l'ultima operazione abbia prodotto un risultato diverso da zero
<code>OR <i>srg, dst</i></code>	modifica <i>dst</i> , in modo tale da contenere l'OR bit a bit tra il precedente contenuto di <i>dst</i> e <i>srg</i> .
<code>CMP <i>arg1,arg2</i></code>	confronta <i>arg1</i> con <i>arg2</i> e modifica il registro dei flag, lasciando invariati <i>arg1</i> e <i>arg2</i>
<code>JE <i>label</i></code>	salta a <i>label</i> solo nel caso in cui i due operandi confrontati dalla precedente CMP erano uguali
<code>CALL output</code>	stampa a video il carattere ASCII associato al corrente contenuto di %AL. Esempio: nel caso in cui in AL si trovi 0x30 stamperebbe a video '0', se ci fosse 0x39 stamperebbe '9'

Uscita del programma assembler: _____

Nello spazio sottostante riportare i passaggi più significativi della soluzione

Soluzione

Soluzione Esercizio 1

Si pone $f_0 = 0.3125$ e poi si applica l'algoritmo "parte intera - parte frazionaria":

$$a_{-1} = I(f_0 * 2 = 0.3125 * 2 = 0.625) = \mathbf{0} \quad f_{-1} = F(0.625) = 0.625$$

$$a_{-2} = I(f_{-1} * 2 = 0.625 * 2 = 1.25) = \mathbf{1} \quad f_{-2} = F(0.25) = 0.25$$

$$a_{-3} = I(f_{-2} * 2 = 0.25 * 2 = 0.5) = \mathbf{0} \quad f_{-3} = F(0.5) = 0.5$$

$$a_{-4} = I(f_{-3} * 2 = 0.5 * 2 = 1.0) = \mathbf{1} \quad f_{-4} = F(1.0) = \mathbf{0}$$

Avendo ottenuto f_{-4} pari a zero, l'algoritmo si ferma. Questo significa che il numero reale 0.3125 ha rappresentazione binaria esatta, pari a $(a_{-1}a_{-2}a_{-3}a_{-4}) = 0101$.

Poichè era richiesta la rappresentazione su 6 bit per la parte frazionaria, se ne aggiungono 2 a zero:

$$R = 0.010100$$

Soluzione Esercizio 2

Il programma somma le 4 cifre meno significative del proprio numero di matricola e poi controlla se tale somma è un multiplo di 8 (nel qual caso stampa 's') oppure no (nel qual caso stampa 'n'). I numeri positivi multipli di 8 sono infatti quelli che terminano con tre bit a zero (0xF8 in binario vale 1111-1000).