Matricola □□□□□□ **Esercizi su Numeri e Assembler - 1 Febbraio 2016**

Cognome □□□□□□□□□□□□□□□□□□ **Nome** □□□□□□□□□□□□□□

Esercizio 1

Data la rappresentazione **A** = 01101, dire a quale intero **a** essa corrisponde nell'ipotesi di aver utilizzato la rappresentazione con *bias* su 5 bit.

Inoltre, data la rappresentazione **R** = {0, 01101, 1000000000}, trovare il numero reale **r** corrispondente nell'ipotesi di rappresentazione in virgola mobile e precisione dimezzata (*half precision*).

a = _____ **r =** _____

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:**

```
.EQU    N,     4
vett:  .BYTE   0x0I, 0x0J, 0x0H, 0x0K

_main:  MOV    $0, %AL
        MOV    $N, %CH
        MOV    $vett, %EBX
loop:   MOV    (%EBX), %DH
        CMP    $5, %DH
        JB     lab
        ADD    %DH, %AL
lab:    INC    %EBX
        DEC    %CH
        JNZ    loop
        call   outbyte
        RET

.INCLUDE "utility"
```

PROMEMORIA

```
CMP arg1,arg2  confronta arg1 (primo argomento) con arg2 (secondo) ed influenza il registro dei flag
               (gli argomenti arg1 e arg2 rimangono invariati).

JB label       salta a lab se e solo se il secondo argomento della precedente CMP era strettamente
               minore primo argomento, interpretando entrambi come naturali.

ADD srg, dest  dest <- sorg + dest  (vengono sommate le rappresentazioni)

JNZ label      salta a label tutte le volte in cui il risultato dell'ultima operazione ha prodotto un
               risultato diverso da zero

DEC reg        decrementa di 1 il contenuto del registro reg

INC reg        incrementa di 1 il contenuto del registro reg

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 0011-0101, stamperebbe a video i caratteri "35"
```

Uscita del programma assembler: _____

Nello spazio sottostante riportare i passaggi più significativi della soluzione

Soluzioni

Soluzione Esercizio 1

Per trovare **a** da **A** basta togliere ad **A** il bias. Il bias vale 2^{K-1}-1. Per K=5, bias = 01111 = 15.
Pertanto 01101-01111 = 13-15 = -2 $\Rightarrow$ Pertanto l'intero **a** cercato vale -2.

Data **R**={0, 01101, 1000000000}, si ha che:
s = 0 (il numero reale **r** cercato è dunque positivo)
E = 01101 ed
F = 1000000000
La mantissa m sarà pertanto il reale 1.1000000000 (per via dell'1 implicito).
L'esponente e = -2, in virtù della rappresentazione con bias (vedere sopra).

Pertanto $\boldsymbol{r} = m \cdot due^{-2} = 1.1 \cdot due^{-2} = 0.011 \cdot due^{0} = 2^{-2} + 2^{-3} = 0.25 + 0.125 =$ **0.375**

Soluzione Esercizio 2

Il programma assembler calcola la somma di tutte le eventuali cifre, tra le quattro meno significative del proprio numero di matricola, che sono maggiori o uguali a 5.

Infatti l'algoritmo preleva dalla memoria tutte e quattro le componenti del vettore *vett*, una ad una, copiandole nel registro DH. Dopo il confronto con l'immediato 5 (istruzione CMP) si avrà un salto a lab solo se DH era minore di 5, nel qual caso il contenuto di AL rimane invariato. Nell'altro caso si incrementa AL di DH, la cifra appena letta. Alla fine il risultato verrà mostrato a video in base 16, attraverso le sue due cifre esadecimali.

Esempio: Se I, J, H e K valgono, rispettivamente 2,7,4 e 5, il programma stamperà a video **0C**, in quanto la somma dei numeri maggiori o uguali a 5 è proprio pari a dodici).