

Cognome  Nome **Esercizio 1**

Sia data la seguente rappresentazione  $R_1$  di un numero reale in virgola mobile su 8 bit (due per l'esponente, cinque per la mantissa, uno per il segno):  $R_1 = \{S, E, F\}$ , con  $S = 0$ ,  $E = 10$  ed  $F = 00110$ .

- A quale numero reale  $r_1$  corrisponde la rappresentazione  $R_1$ ?
- Quanto vale in base dieci il minimo numero  $r_2$  rappresentabile su 8 bit, ossia quello associato al “ $-\infty$ ”?

 $r_1$  \_\_\_\_\_ $r_2$  \_\_\_\_\_

Nello spazio sottostante riportare i passaggi più significativi della soluzione

**Esercizio 2 Dire cosa stampa a video il seguente programma assembler**

```
vett:    .ASCII    "010001"

_main:  MOV     $0, %AL
        MOV     $vett,%EBX
        MOV     $6,%CL
loop:   MOV     (%EBX),%AH
        CMP     $'0',%AH
        JNE    lab1
        INC     %AL
        CMP     $3,%AL
        JE     lab3
        INC     %EBX
        DEC     %CL
        JZ     lab2
        JMP    loop
lab1:   MOV     $0,%AL
        INC     %EBX
        DEC     %CL
        JZ     lab2
        JMP    loop
lab2:   MOV     $'n', %AL
        CALL   output
        RET
lab3:   MOV     $'y', %AL
        CALL   output
        RET

.INCLUDE "utility"
```

## PROMEMORIA

<code>JZ label</code>	salta a <code>label</code> nel caso in cui l'ultima operazione abbia prodotto risultato zero
<code>CMP arg1, arg2</code>	confronta <code>arg1</code> con <code>arg2</code> e modifica il registro dei flag, lasciando invariati <code>arg1</code> e <code>arg2</code>
<code>JNE label</code>	salta a <code>label</code> solo nel caso in cui i due operandi confrontati dalla precedente <code>CMP</code> erano diversi
<code>JE label</code>	salta a <code>label</code> solo nel caso in cui i due operandi confrontati dalla precedente <code>CMP</code> erano uguali
<code>JZ label</code>	salta a <code>label</code> solo nel caso in cui l'istruzione precedente ha prodotto zero come risultato
<code>CALL output</code>	stampa a video il carattere ASCII associato al corrente contenuto di <code>%AL</code> . Esempio: nel caso in cui in <code>AL</code> si trovi <code>0x30</code> stamperebbe a video <code>'0'</code> , se ci fosse <code>0x39</code> stamperebbe <code>'9'</code> .

## Uscita del programma assembler: \_\_\_\_\_

Nello spazio sottostante riportare i passaggi più significativi della soluzione

# Soluzioni

## Soluzione Esercizio 1

La mantissa  $m_1 = 1.00110$ , per via dell'uno implicito.

L'esponente  $e_1$  vale  $E_1 - bias = 10 - 01 = +1$

$S=1$  (ossia il numero è negativo).

Pertanto  $r_1 = -1.00110 * due^{+1} = -10.0110 = -(2^{+1} + 2^{-2} + 2^{-3}) = -(2 + 0.25 + .125) = -2.375$

Il massimo reale negativo è quello associato al segno positivo ( $S_2=1$ ), al massimo esponente  $E_2=11$  ed alla massima mantissa  $F_2=11111$ .

La mantissa  $m_2$  associata vale:  $1.11111$ , mentre l'esponente  $e_2$  associato vale:  $11 - 01 = +2$ .

Pertanto  $r_2 = -1.11111 * due^{+2} = -111.111 = 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} = -7.875$

## Soluzione Esercizio 2

Il programma stampa a video il carattere 'y' se la sequenza di caratteri ascii puntata da `vett` contiene almeno tre zeri consecutivi e 'n' altrimenti. Nel caso specifico, stamperà 'y'.