

Matricola □□□□□□

Esercizi su Numeri e Assembler del 29 Giugno 2016

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

**Esercizio 1**

Trovare la rappresentazione binaria **A** in complemento a 2 dell'intero  $a = -31$  su  $p = 6$  bit.

Trovare inoltre la rappresentazione in base 8 del naturale **A** appena trovato.

**A (in base 2)** \_\_\_\_\_ **A (in base 8)** \_\_\_\_\_

*(spazio riservato allo svolgimento dell'esercizio)*

**Esercizio 2** Dire cosa stampa a video il seguente programma assembler (immaginare al posto della **x** la cifra meno significativa del proprio numero di matricola)

```
.EQU N, 3
vett1: .BYTE 2, 3, 1
vett2: .BYTE 0, 2, x

_main: CALL sub
      MOV $N, %CH
      MOV $vett1, %ESI
      MOV $vett2, %EDI
      MOV $0, %CL
loop:  MOV (%ESI), %BL
      MOV (%EDI), %AL
      MUL %BL
      ADD %AL, %CL
      INC %ESI
      INC %EDI
      DEC %CH
      JNZ loop
      MOV %CL, %AL
      CALL outbyte
      RET

sub:   MOV $0x85, %AL
      OR  $0x30, %AL
      AND $0x7F, %AL
      CALL output
      RET

.INCLUDE "utility"
```

## PROMEMORIA

AND srg, dest	Il registro dest viene aggiornato con il risultato dell'AND logico del suo precedente valore ed il contenuto del registro srg
OR srg, dest	Il registro dest viene aggiornato con il risultato dell'OR logico del suo precedente valore ed il contenuto del registro srg
CALL output	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'
MUL %BL	moltiplica il contenuto di %BL per %AL e mette il risultato in %AX
JNZ lab	salta a lab nel caso in cui l'operazione precedente NON abbia messo ad 1 lo zero flag
ADD src,dest	cambia il contenuto di dest sommandoci quello di src
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

Trattandosi di un numero negativo, il naturale  $A$  vale  $2^p - |-31| = 64 - 31 = 33_{(10)}$ , ossia  $A=100001_{(2)}$ .

Per esprimere  $A$  in base 8 basta raggruppare le cifre a tre a tre partendo da sinistra e decodificare ciascun gruppo mediante conversione da binario a decimale:  $A=41_{(8)}$

## Soluzione Esercizio 2

Il programma assembler come prima cosa chiama la sottoprocedura 'sub'. In tale sottoprocedura si carica in AL il naturale 9, che corrisponde all'esadecimale 0xD5. L'OR con 0x30 trasforma il contenuto di AL in F5, mentre il successivo AND con 3F ne trasforma il contenuto in 0x35. La call output stampa il codice ASCII dell'esadecimale 0x35, che è noto essere il carattere '5' (infatti 0x30 corrisponde a '0', 0x31 ad '1', ecc...).

Al rientro della procedura si effettua il prodotto scalare fra il primo vettore a tre componenti ed il secondo:  $2*0 + 3*2+1*X = 6 + X$ . Poiché  $X$  varia da 0 a 9, il risultato in esadecimale varierà da 6 ad E

In conclusione l'uscita del programma sarà "506", oppure "507", ... oppure "50E", a seconda del valore di  $X$ .