

Matricola

Esercizi su Numeri e Assembler del 20 Luglio 2016

Cognome  Nome

### Esercizio 1

Dato il numero naturale  $A$  la cui codifica in base 5 è 104, trovarne la codifica in base 2.

Successivamente, assumendo che la rappresentazione binaria trovata sia quella dell'esponente  $E$  di un numero reale in *half precision* ( $K=5$ ), dire quanto vale l'esponente intero  $e$  corrispondente.

$A$  (in base 2) \_\_\_\_\_  $e$  \_\_\_\_\_

*(spazio riservato allo svolgimento dell'esercizio)*

**Esercizio 2** Dire cosa stampa a video il seguente programma assembler, assumendo di inserire da tastiera le due cifre meno significative del proprio numero di matricola

```
_main:  CALL    inbyte
        AND     $0x0F, %AL
        CALL    outbyte
        ADD     $4, %AL
        MOV     $0, %BL
loop:   INC     %BL
        SUB     $3, %AL
        JNS     loop
        DEC     %BL
        CALL    sub
        ADD     $3, %AL
        CALL    outbyte
        RET

sub:     PUSH    %EAX
        MOV     %BL, %AL
        CALL    outbyte
        POP     %EAX
        RET

.INCLUDE "utility"
```

## PROMEMORIA

|               |                                                                                                                                                                                                                                                                  |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CALL inbyte   | legge da tastiera due caratteri (purché compresi tra '0' e '9' oppure tra 'A' e 'F'), li interpreta come cifre esadecimali e li utilizza per inizializzare AL.<br>Esempio: qualora venissero inseriti '4' e 'B', in AL ci finirebbe 0100-1011 (0x4B)             |
| AND srg, dest | Il registro dest viene aggiornato con il risultato dell'AND logico del suo precedente valore ed il contenuto del registro srg                                                                                                                                    |
| 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.<br>Esempio: qualora in AL vi fosse 0011-0101, stamperebbe a video i caratteri "35" |
| ADD src,dest  | cambia il contenuto di dest sommandovi quello di src                                                                                                                                                                                                             |
| SUB src,dest  | cambia il contenuto di dest sottraendovi quello di src                                                                                                                                                                                                           |
| JNS label     | salta fintantochè il <i>sign flag</i> vale 0, ossia si smette di saltare alla label non appena l'ultima operazione ha settato il sign flag (questo si verifica l'ultima operazione produce un risultato minore stretto di zero)                                  |

**Uscita del programma assembler:** \_\_\_\_\_

Nello spazio sottostante riportare i passaggi più significativi della soluzione

# Soluzioni

## Soluzione Esercizio 1

Il numero  $104_{(5)}$  corrisponde al decimale  $1*5^2+4*5^0 = 29_{(10)}$ .

$29_{(10)}$  equivale a  $16+8+4+1$ , ossia  $11101_{(2)}$ .

Considerando  $11101$  come il valore di  $E$ , *l'intero e corrispondente è il naturale che si ottiene sottraendo ad  $E$  il bias  $B = K^{5-1}-1=15$ , ossia  $29-15 = 14$ . Dunque l'esponente  $e$  è positivo e vale 14.*

## Soluzione Esercizio 2

Siano  $X$  e  $Y$  le cifre meno significative del proprio numero di matricola ( $Y$  sia quella meno significativa in assoluto).

La chiamata alla subroutine `inbyte` carica nel registro AL l'esadecimale  $0xXY$ . La successiva istruzione `AND` azzererà i quattro bit più significativi. Pertanto la chiamata alla `outbyte` mostrerà a video "0Y".

Dopo aver aggiunto 4 al contenuto di AL, nel registro ci sarà il naturale  $Z = 4+Y$ . Poichè  $Y$  è una cifra fra 0 e 9,  $Z$  può assumere solo i valori naturali che vanno da 4 a 13.

All'interno del ciclo si incrementa il contenuto del registro BL e si sottrae 3 al contenuto di AL fino a quando non si ottiene un risultato negativo (ossia viene settato il *sign flag*). Incrementando BL, esso conterrà il risultato della divisione intera di  $Z$  per 3. Pertanto a video verrà mostrato il risultato dell'operazione  $Z \text{ div } 3$ . Tale risultato viene stampato a video dalla chiamata alla subroutine `sub`.

Tale subroutine salva il contenuto del registro EAX (e dunque anche di AL), prima di caricare in AL il contenuto di BL. Prima di uscire il precedente contenuto di AL viene ripristinato. AL conteneva il resto della divisione intera di  $Z$  per 3, che quindi viene stampato a video. In conclusione l'algoritmo stampa sia  $Z \text{ div } 3$  che  $Z \text{ mod } 3$ .

Riassumendo l'uscita sarà (in verde il risultato dell'AND, in rosso quello della `div` e in blu quello della `mod`):

per  $Y = 0$ ,  $Z = 4$  e l'uscita prodotta complessivamente sarà "000101"

per  $Y = 1$ ,  $Z = 5$  e l'uscita prodotta complessivamente sarà "010102"

per  $Y = 2$ ,  $Z = 6$  e l'uscita prodotta complessivamente sarà "020200"

per  $Y = 3$ ,  $Z = 7$  e l'uscita prodotta complessivamente sarà "030201"

per  $Y = 4$ ,  $Z = 8$  e l'uscita prodotta complessivamente sarà "040202"

per  $Y = 5$ ,  $Z = 9$  e l'uscita prodotta complessivamente sarà "050300"

per  $Y = 6$ ,  $Z = 0$  e l'uscita prodotta complessivamente sarà "060301"

per  $Y = 7$ ,  $Z = 11$  e l'uscita prodotta complessivamente sarà "070302"

per  $Y = 8$ ,  $Z = 12$  e l'uscita prodotta complessivamente sarà "080400"

per  $Y = 9$ ,  $Z = 13$  e l'uscita prodotta complessivamente sarà "090401"