

Esercizi Assembly 4

M. Sonza Reorda – M. Grosso – M. Monetti

Politecnico di Torino
Dipartimento di Automatica e Informatica

Esercizio 1

- Si scriva un programma in linguaggio Assembly 8086 che dica se un'equazione di secondo grado nella forma $ax^2+bx+c=0$ ha o meno soluzioni reali. I coefficienti a , b e c siano variabili di tipo *word*.
 - Si ricorda che la soluzione di un'equazione di secondo grado ha la forma:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Esercizio 2

- Si scriva un programma in grado di calcolare il valore di un insieme di monete di diverso importo (espresso in centesimi di Euro). Siano dati i seguenti vettori:
 - *valore*, vettore di *word* indicante il valore di ciascun tipo di moneta
 - *monete*, vettore di *byte* indicante il numero di monete di ciascun tipo.
- Ad esempio, con
 - *valore* dw 1, 2, 5, 10, 20, 50, 100, 200
 - *monete* db 100, 23, 17, 0, 79, 48, 170, 211
 si hanno 100 monete da 1 centesimo, 23 monete da 2 centesimi, e così via.
- Il programma deve fornire il risultato aggiornando due variabili precedentemente dichiarate, di tipo *word*, denominate *euro* e *cent*, e rappresentanti rispettivamente l'importo in euro e in centesimi. Nell'esempio, il valore risultante è pari a 63411 centesimi, quindi alla fine del programma le due variabili *euro* e *cent* varranno rispettivamente 634 e 11.

Esercizio 3

- La seguente tabella riassume dati di alcuni microprocessori degli ultimi decenni:

| Nome | Num. di transistor | Area [mm ²] | Anno |
|--------------------|--------------------|-------------------------|------|
| Intel 4004 | 2 300 | 12 | 1971 |
| Intel 8080 | 4 500 | 20 | 1972 |
| Zilog Z80 | 8 500 | 18 | 1976 |
| Intel 8086 | 29 000 | 33 | 1978 |
| Motorola 68000 | 68 000 | 44 | 1979 |
| Intel 80286 | 134 000 | 49 | 1982 |
| Intel 80386 | 275 000 | 104 | 1985 |
| Intel 80486 | 1 180 235 | 173 | 1989 |
| Pentium | 3 100 000 | 294 | 1993 |
| AMD K6 | 8 800 000 | 162 | 1997 |
| AMD K7 | 22 000 000 | 184 | 1999 |
| Pentium 4 Prescott | 112 000 000 | 110 | 2004 |
| ARM Cortex-A9 | 26 000 000 | 31 | 2007 |
| i7 skylake | 1 750 000 000 | 122 | 2015 |

Esercizio 3 [cont.]

- Dati due vettori, *trans* e *area*, contenenti, rispettivamente, i dati su numero di transistor e area della precedente tabella, si scriva un programma che calcoli per ciascun processore la densità in transistor/mm² e la salvi in un vettore di *word*.
- Qualora si ottengano valori non rappresentabili, si scriva il valore 0ffffh.
- Suggerimento: per prevedere l'esito della divisione è sufficiente verificare che la «parte alta» del dividendo sia più piccola del divisore.

Esercizio 4

- Sia data una matrice quadrata di *word* memorizzata per righe (numero di righe pari a DIM, con DIM dichiarato come costante).
- Si scriva un programma che sia in grado di valutare se la matrice quadrata è simmetrica o diagonale. Il programma dovrà stampare a video un valore pari a:
 - 2 se la matrice è diagonale
 - 1 se la matrice è simmetrica
 - 0 se la matrice non è simmetrica.

Esercizio 4 [cont.]

- Si ricorda che in una matrice diagonale solamente i valori della diagonale principale possono essere diversi da 0, mentre una matrice simmetrica ha la proprietà di essere la trasposta di se stessa

- Esempio di matrice diagonale:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

- Esempio di matrice simmetrica:

$$\begin{bmatrix} 1 & 4 & 5 & 6 & 7 \\ 4 & 2 & 8 & 6 & 4 \\ 5 & 8 & 3 & 2 & 9 \\ 6 & 6 & 2 & 4 & 4 \\ 7 & 4 & 9 & 4 & 5 \end{bmatrix}$$

Esercizio 5

- Un modo per calcolare la radice quadrata approssimata di un numero intero consiste nel contare la quantità di numeri dispari che possono essere sottratti dal numero di partenza. La soluzione proposta in linguaggio C è la seguente:

```
main()
{
    int num, sqr=0, disp=1;
    ...
    num--;
    while (num >= 0)
    {
        sqr++;
        disp += 2;
        num -= disp;
    }
    ...
}
```

- Realizzare un programma in Assembly 8086 che calcoli la radice quadrata approssimata di un intero positivo (16 bit).

Esercizio 6

- Sia data una matrice quadrata di *byte* di dimensione 8x8 preinizializzata. La matrice contiene valori *unsigned*.
- Per ogni elemento della matrice si calcoli la somma dei 4 elementi limitrofi (nelle posizioni N, E, S, O; per gli elementi lungo i bordi si consideri solo il sottoinsieme di elementi esistenti). Quindi, si trovi l'elemento per cui tale somma è massima e ne si forniscano le coordinate di riga e colonna.
- In caso di occorrenze multiple, si operi una scelta opportuna.

Esercizio 6 [cont.]

- Esempio:

| | | | | | | | |
|---|----|----|---|----|----|----|----|
| 0 | 4 | 0 | 0 | 0 | 0 | 0 | 60 |
| 0 | 5 | 0 | 0 | 11 | 0 | 0 | 0 |
| 0 | 5 | 7 | 0 | 0 | 10 | 0 | 0 |
| 0 | 0 | 0 | 9 | 0 | 0 | 49 | 0 |
| 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 |
| 0 | 10 | 3 | 9 | 0 | 0 | 12 | 0 |
| 0 | 0 | 58 | 0 | 0 | 17 | 0 | 0 |
| 0 | 1 | 0 | 0 | 3 | 0 | 0 | 0 |

- Risultato: $x = 3, y = 6$