

### Esercizio 1

Considerare il programma riportato nel file "esercizio1.s":

- a) Indicare quanto vale  $n$  nel sistema decimale e nel sistema binario, prima e dopo l'esecuzione della macro nel main.
- b) Indicare quanto valgono i flag NZCV di Pstate, quindi assemblare ed eseguire il programma per verificare se la risposta data è corretta.
- c) Indicare quanto vale  $n$  nel sistema decimale e nel sistema binario dopo l'esecuzione della macro nel main, se al posto di `add_to_n 1` si avesse `add_to_n -1`.
- d) Indicare quanto valgono i flag NZCV di Pstate anche in questo ultimo caso, quindi assemblare ed eseguire il programma per verificare se la risposta data è corretta.
- e) Indicare nel sistema decimale e nel sistema binario, il più piccolo numero che se aggiunto ad  $n$  al posto di 1 nel main, farebbe ottenere un risultato che richiederebbe più di un byte.

### Esercizio 2

Indicare tipologia di indirizzamento e significato di ciascuna delle seguenti istruzioni load:

- a) `ldr w0, [x1, #-3]`
- b) `ldr x2, [x1]`
- c) `ldr x1, [x0, #3]!`
- d) `ldr x9, [x8], #8`
- e) `ldr x0, =m`
- f) `ldr x0, m`

### Esercizio 3

Sostituendo `str` a `ldr` nell'esercizio 3, otteniamo le seguenti istruzioni store. Indicare quali delle istruzioni sono valide ed inoltre, indicare tipologia di indirizzamento e significato di ciascuna istruzione valida.

- a) `str w0, [x1, #-3]`
- b) `str x2, [x1]`
- c) `str x1, [x0, #3]!`
- d) `str x9, [x8], #8`
- e) `str x0, =m`
- f) `str x0, m`

### Esercizio 4

Considerare il programma riportato nel file "esercizio4.s":

- a) Aggiungere delle opportune istruzioni load e store (utilizzando uno a scelta tra, post oppure pre indexed immediate offset) per impostare a 0 gli elementi che si trovano in posizioni pari nell'array  $n$ . Assemblare ed eseguire il programma per verificarne il corretto funzionamento. Si ricorda che un array può essere, in un certo senso,

immaginato come una lista in Python e che gli elementi di un array sono memorizzati in modo contiguo in memoria.

- b) Si noti che non è possibile utilizzare delle istruzioni `stp` per svolgere il punto a. Per quale ragione?

### Esercizio 5

Considerare il programma riportato nel file “esercizio5.s”:

- a) Aggiungere delle opportune istruzioni `load` e `store` (utilizzando uno a scelta tra, `post` oppure `pre indexed immediate offset`) per impostare al valore  $x$  (nella sezione `.data`) tutti gli elementi nell'array  $n$ . Assemblare ed eseguire il programma per verificarne il corretto funzionamento. Si ricorda che un array può essere, in un certo senso, immaginato come una lista in Python e che gli elementi di un array sono memorizzati in modo contiguo in memoria.
- b) Rispondere al punto a) utilizzando delle istruzioni `stp` con `pre-indexed immediate offset`.
- c) Modificare il programma in modo tale da impostare al valore dell'elemento in posizione 1 (ovvero al valore 13) tutti gli elementi nell'array  $n$  utilizzando delle istruzioni `stp` con `post-indexed immediate offset`.