

Architettura degli Elaboratori a.a. 2024/25

Appello d'esame 14/02/2025 (ASM)

Operazioni preliminari

1. Aprire il file AE - Uso del computer in laboratorio per la prova ASM.pdf, contenente informazioni utili per lo svolgimento di questa prova.
2. Aprire tramite RARS il file `program01.asm` in questa directory
3. Completare le seguenti righe con i propri dati in `program01.asm`

```
#####  
# INSERIRE I PROPRI DATI QUI:  
# Nome:  
# Cognome:  
# Matricola:  
#####
```

Esercizio

La routine di caricamento già fornita in `program01.asm` riceve in input una sequenza di interi a 32 bit. Il primo intero fornito dall'utente rappresenta la grandezza del lato di una matrice N . Il programma farà allora $N \times N$ chiamate a sistema per ricevere in input gli altri elementi della matrice, rappresentati e inseriti riga per riga. Il programma scriverà in RAM i valori N e i valori della matrice, e restituirà in `a0` l'indirizzo base della matrice, e in `a1` l'indirizzo base di N . Attenzione: il programma farà quindi un numero di chiamate a sistema dipendente dal primo intero inserito. Ogni input viene eseguito con la digitazione di un numero intero seguita dal tasto invio. Ad esempio se l'utente ha digitato i seguenti 5 numeri intervallati dal tasto invio: 2,1,2,3,4, si intende rappresentare una matrice 2×2 con valori:

```
1 2  
3 4
```

che verrà scritta in RAM nel seguente modo:

- N (cioè il valore 2) verrà inserito in una word con indirizzo salvato in `a1`.
- la matrice verrà inserita in RAM come un vettore di 4 word (in questo caso `.word 1,2,3,4`) con indirizzo base salvato in `a0`.

Si realizzi il seguente programma in assembly RISC-V. Il programma deve stampare a schermo un intero rappresentante la somma dei primi due elementi di ogni riga. Se la matrice è composta da 1 solo elemento, il programma deve riportare tale elemento. Se la matrice è composta da zero elementi, il programma deve riportare 0. Se, ad esempio, $N=3$ e il vettore matrice è uguale a 1 1 4 2 2 6 3 5 6 allora dovrà essere scritto a schermo:

```
14
```

perché i primi due elementi di ogni riga sono rispettivamente 1,1,2,2,3,5. Se, invece, $N=1$ ed il vettore matrice è uguale a -34 allora dovrà essere scritto a schermo:

```
-34
```

perché si tratta di uno dei casi base descritti sopra.

Note: Commentare ogni riga di codice avendo cura di spiegare a cosa servano i registri. Si ricorda che la syscall per la stampa di un intero prevede come argomenti **a7=1** e **a0** assegnato con il valore da stampare.

Risultato atteso

Per ogni input al programma va stampato l'output della procedura suddetta seguito da accapo come spiegato precedentemente.

Ad esempio, il file **test-02.in** contiene il seguente input:

```
2
1
1
2
5
```

e il suo output atteso (all'interno di **test-02.expt**) è:

```
9
```

È possibile studiare i casi di test aprendo i file di input (nel formato **test-xy.in**, dove *xy* è un numero a due cifre) e output atteso (estensione **test-xy.expt**), confrontandoli col proprio (estensione **test-xy.out**).

Verifica di corretta esecuzione dell'esercizio

Per verificare che l'esercizio sia stato completato correttamente eseguire **run.sh** (doppio click sul file dal file manager, oppure esecuzione del comando **./run.sh** da terminale) e visualizzare il risultato aprendo il file **test_results.html**. Per ulteriori informazioni, consultare il file **AE - Uso del computer in laboratorio per la prova ASM.pdf**.

Bonus

Una soluzione pienamente funzionante, dove ogni test verrà completato usando meno di 150 istruzioni sarà premiata con un bonus di **1 punto** sul voto finale.

Note

- Non è consentito modificare il *data segment*.
- Il limite massimo di istruzioni eseguibili è **305900**. Oltre quel numero, l'esecuzione viene automaticamente terminata.
- Il file **program01.vuoto.asm** contiene una copia del file **program01.asm** che può essere utile in caso sia necessario ripartire da capo.
- Attenzione a non eseguire loop infiniti con leak della memoria: RARS potrebbe crashare e cancellare il file che state scrivendo, sentitevi liberi di salvare un file di backup prima di eseguire del codice rischioso.