Architettura degli Elaboratori a.a. 2024/25

Appello d'esame 12/02/2025 (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

Si realizzi in assembly RISC-V il seguente programma. Sia data in input da utente una stringa di caratteri `S`. La stringa è lunga al più 20 caratteri (non è necessario verificare questo vincolo) e consiste dei seguenti simboli: '`0`' (codice ASCII `0x30`), '`1`', ... fino a '`9`'. Inoltre il primo carattere può essere il simbolo '`?`' (codice ASCII `0x3F`). La sequenza termina quando viene letto il carattere `'\n'` (codice ASCII `0xA`). Il programma deve contare il numero di occorrenze pari e dispari, considerando numeri da una cifra ciascuno, solo per le cifre strettamente maggiori di 2 e strettamente minori di 9 ed escludendo il primo carattere. In altre parole, deve contare (dopo aver escluso il primo carattere) il numero di occorrenze pari e dispari **senza contare** le occorrenze dei caratteri '`0`', '`1`', '`2`' e '`9`'.

Se, ad esempio, `S` è `8934` allora dovrà essere scritto a schermo su due linee separate:

```
1
1
```

perché Il primo carattere '`8`' è stato ignorato, dopo il quale si ha 1 occorrenza pari maggiore di 2 e minore di 9 (corrispondente al carattere '`4`') e una occorrenza dispari maggiore di 2 e minore di 9 (corrispondente al carattere '`3`').

Se, invece, `S` fosse `?8934` allora dovrà essere scritto a schermo su due linee separate:

```
2
1
```

perché Il primo carattere '`?`' è stato ignorato, dopo il quale si anno 2 occorrenze pari maggiori di 2 e minori di 9 (corrispondente ai carattere '`8`' e '`4`') e una occorrenza dispari maggiore di 2 e minore di 9 ('`3`').

Nella sezione dedicata al **text segment**, il programma deve avere nel comparto *main* il caricamento dei dati da input (già fornito nel file `program01.asm`: l'indirizzo base della stringa `S` verrà caricato nel registro `a0`), la chiamata a una funzione **`contaOccorrenze`** definita di seguito e la stampa a terminale delle occorrenze.

La funzione `contaOccorrenze` accetta come argomento:

- `a0`: l'indirizzo base della stringa `S`;

e restituisce come risultato:

- `a0`: la somma delle occorenze pari (nell'esempio, 1).
- `a1`: la somma delle occorenze dispari (nell'esempio, 1).

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. La syscall per la stampa di un carattere (necessaria per separare i due output con una `'\n'`) prevede come argomenti `a7=11` e `a0` assegnato con il carattere 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-03.in` contiene il seguente input: '`?80085`', il suo output atteso (all'interno di `test-03.expt`) è:

```
2
1
```

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 130 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.