

Architettura degli Elaboratori a.a. 2021/22

Appello d'esame 09/09/2022 (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 MARS 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 MIPS il seguente programma. Sia data in input da utente una stringa di caratteri `S`. La sequenza termina quando viene letto `'\n'` (codice ASCII `0xA`). Prima del carattere `'\n'`, essa può contenere solo simboli numerici: `'0'` (codice ASCII `0x30`), `'1'`, fino a `'9'`. La stringa consta al più di 20 caratteri (`'\n'` incluso).

Il programma deve contare due quantità: (i) il numero di occorrenze in cui un carattere è preceduto da un carattere identico, (ii) la somma totale dei valori numerici presi una cifra alla volta.

Se, ad esempio, `S` è `1123455` allora dovrà essere scritto a schermo su due linee separate:

```
2  
21
```

perché si hanno 2 occorrenze in cui un carattere è preceduto da un carattere identico (il secondo carattere 1 e l'ultimo carattere 5), e la somma totale dei valori numerici presi una cifra alla volta è 21.

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 ad una funzione **contaOccorrenze** definita di seguito e la stampa a terminale delle occorrenze. La stampa a terminale non deve avvenire all'interno della funzione **contaOccorrenze** ma nel comparto *main* chiamante o in una funzione dedicata di propria stesura.

La funzione **contaOccorrenze** accetta come argomento:

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

e restituisce come risultato:

- `$v0`: la somma delle occorrenze in cui un carattere è preceduto da un carattere identico (nell'esempio, 2).
- `$v1`: la somma totale dei valori numerici presi una cifra alla volta (nell'esempio, 21).

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 `$v0=1` e `$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 `$v0=11` e `$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:

4444444

Il suo output atteso (all'interno di `test-03.expt`) è:

5

24

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`).

Bonus

Una soluzione pienamente funzionante, realizzata seguendo una procedura ricorsiva, sarà premiata con un bonus di **1 punto** sul voto finale.

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`.

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: MARS potrebbe andare in crash e cancellare il file che state scrivendo! Sentitevi liberi di salvare un file di backup prima di eseguire del codice rischioso.