

### **ESERCIZIO 1 – C [LEGGERE ATTENTAMENTE TUTTO IL TESTO SEGUENTE PRIMA DI INIZIARE A LAVORARE]**

Cercare nella specifica completa del linguaggio della JVM (sul sito ORACLE indicato nei lucidi e nella pagina DIR del corso e facilmente reperibile facendo una ricerca su google con le parole chiave “instruction set JVM oracle”) la definizione dell’istruzione **IF\_ICMPLT label**. Utilizzare il codice operativo (sia mnemonico che numerico) indicato nella specifica.

Riportare la definizione nella relazione per il primo esercizio (potete scegliere se inserire la versione originale in inglese o una traduzione in italiano), quindi creare il codice MAL che realizza tale nuova istruzione e integrarlo nel file mic1ijvm.mal (create una copia del file prima di modificarlo e cambiate nome al file copia: mic1ijvmA\_NomeBreveGruppo.mal). Le nuove istruzioni MAL inserite nel microinterprete dovranno essere **TUTTE** debitamente commentate; non dimenticate di inserire il comando “.label” all’inizio del sorgente del microinterprete per indicare al compilatore MAL dove posizionare la prima istruzione (if\_icmpge1) della sequenza di microistruzioni aggiunta. Fare anche le modifiche necessarie al file ijvm.conf (affinché il compilatore di programmi IJVM integrato nell’emulatore possa riconoscere e tradurre la nuova istruzione).

Realizzare una serie di test per verificare che l’istruzione nuova si comporti come desiderato (anche utilizzando la finestra di Debug messa a disposizione dall’emulatore). Nella relazione si dovrà spiegare sia l’idea generale che sottende l’implementazione sia come sono stati scelti i test per verificare il corretto funzionamento; per quanto riguarda quest’ultimo punto, riportare in una tabella per ogni test lo stato (valore dei registri e delle parole di memoria su cui opera l’istruzione) subito prima e subito dopo l’esecuzione della nuova istruzione.

**NOTA:** l’istruzione confronta i due elementi in cima allo stack  $\text{mem}[\text{SP}-1]$  e  $\text{mem}[\text{SP}]$  (o registro TOS) per verificare se  $\text{mem}[\text{SP}-1] < \text{mem}[\text{SP}]$ ; l’implementazione dell’istruzione dovrà tenere conto del fatto che se si fa calcolare all’ALU la differenza  $\text{mem}[\text{SP}-1] - \text{mem}[\text{SP}]$  al fine di verificare se i due valori soddisfano la condizione richiesta per effettuare il salto si potrebbe verificare overflow e in questo caso l’istruzione si potrebbe comportare in modo diverso dal previsto. Realizzare due versioni di microcodice: una più semplice che NON tiene conto del problema dell’overflow e una più completa che invece ne tiene conto. Mostrare la differenza di comportamento quando le due versioni sono sottoposte ai diversi test.

**SUGGERIMENTO PER L’ORGANIZZAZIONE DEL LAVORO:** Ipotizzando che il gruppo sia formato da tre persone vi suggerisco la seguente modalità di organizzazione del lavoro. Ciascuno legga attentamente questa pagina, quindi dedicate un po’ di tempo per discutere in gruppo possibili soluzioni arrivando ad una traccia di massima della struttura del codice. Due persone nel gruppo implementano le due versioni del codice: quella che non tiene conto dell’overflow e quella che invece ne tiene conto (vedere NOTA sopra). La terza persona invece preparerà i test e si occuperà di sperimentare le due versioni di microcodice esteso. Quest’ultima persona si occuperà anche di redigere la relazione raccogliendo le informazioni necessarie dagli altri componenti del gruppo. Nella parte iniziale della relazione dovrete specificare chi si è occupato di ciascun compito.