

## ESERCIZIO 3 SETTIMANA 10

Dato il codice in Assembly per la CPU x86 allegato qui di seguito, identificare lo scopo di ogni istruzione, inserendo una descrizione per ogni riga di codice.

**mov EAX,0x20** -> Questa istruzione sposta il valore immediato 0x20 (32 in decimale) nel registro EAX. Può essere utilizzata per inizializzare un EAX con un valore specifico.

**mov EDX,0x38** -> Questa istruzione sposta il valore immediato 0x38 (che corrisponde a 56 in decimale) nel registro EDX.

**add EAX,EDX** -> Questa istruzione esegue l'operazione di addizione tra il contenuto del registro EAX e il contenuto del registro EDX, e il risultato viene memorizzato nel registro EAX. Lo scopo di questa istruzione è quello di aggiungere il valore contenuto nel registro EDX al valore contenuto nel registro EAX e memorizzare il risultato nella stessa posizione di memoria di EAX.

**mov EBP,EAX** -> Questa istruzione copia il contenuto del registro EAX nel registro EBP. Copiando il contenuto di EAX in EBP, si può utilizzare il valore contenuto in EAX come nuovo riferimento per accedere alle variabili locali o ai parametri della funzione.

**cmp EBP,0xa** -> Questa istruzione confronta il valore contenuto nel registro EBP con il valore immediato 0xa. Lo scopo principale dell'istruzione CMP è quello di eseguire una sottrazione tra due operandi senza memorizzare il risultato

**jge 0x1176 <main+61>** -> Questa istruzione esegue un salto condizionale. Questa istruzione confronta due valori e se il primo è maggiore o uguale al secondo, esegue un salto all'indirizzo specificato, altrimenti continua l'esecuzione sequenziale.

**mov eax,0x0** -> Questa istruzione assegna il valore immediato al registro EAX. Quindi il valore 0x0 viene caricato nel registro EAX.

**call 0x1030 <printf@plt>** -> Questa istruzione effettua una chiamata di funzione alla posizione di memoria specificata. Questa istruzione effettuerà una chiamata alla funzione printf().