

ESERCIZIO W22D1 PRATICA 1

Traccia:

Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly.

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.

Ricordate che i numeri nel formato 0xYY sono numeri esadecimali. Per convertirli in numeri decimali utilizzate pure un convertitore online, oppure la calcolatrice del vostro computer (per programmatori).

```
0x00001141 <+8>:  mov  EAX,0x20
0x00001148 <+15>:  mov  EDX,0x38
0x00001155 <+28>:  add   EAX,EDX
0x00001157 <+30>:  mov  EBP,EAX
0x0000115a <+33>:  cmp   EBP,0xa
0x0000115e <+37>:  jge   0x1176 <main+61>
0x0000116a <+49>:  mov  eax,0x0
0x0000116f <+54>:  call  0x1030 <printf@plt>
```

Descrizione per ciascuna istruzione nel codice Assembly fornito:

1. **mov EAX, 0x20**: Questa istruzione assegna il valore esadecimale **0x20** al registro **EAX**. In termini decimali, questo è il numero 32.
2. **mov EDX, 0x38**: Qui viene assegnato il valore esadecimale **0x38** al registro **EDX**, che corrisponde a 56 in decimale.
3. **add EAX, EDX**: Questa istruzione esegue un'addizione tra i valori contenuti nei registri **EAX** (che ora è 32) e **EDX** (che ora è 56). Quindi, **EAX** diventerà **32 + 56 = 88** in decimale.
4. **mov EBP, EAX**: Il valore attuale di **EAX** (88) viene spostato nel registro **EBP**.
5. **cmp EBP, 0xa**: Questa istruzione confronta il valore contenuto in **EBP** (che è 88) con il valore esadecimale **0xa** (che è 10 in decimale).
6. **jge 0x1176 <main+61>**: Questo è un salto condizionale. Se il confronto precedente (**cmp EBP, 0xa**) indica che **EBP** è maggiore o uguale a **0xa** (ovvero 10), allora il programma salta all'indirizzo **0x1176** (corrispondente alla riga successiva nel programma).
7. **mov eax, 0x0**: Se la condizione del confronto precedente non è soddisfatta (cioè se **EBP** è inferiore a **0xa**), viene caricato il valore **0x0** nel registro **EAX**.
8. **call 0x1030 <printf@plt>**: Infine, il programma esegue una chiamata alla funzione **printf** situata all'indirizzo **0x1030**. Questo probabilmente stampa un output sulla console con il valore corrente di **EAX** (che potrebbe essere **0** se la condizione di confronto non è soddisfatta).

In sintesi, questo codice Assembly sembra fare un calcolo aritmetico (somma) tra due costanti, confrontare il risultato con un valore specifico (10 in decimale) e poi fare una chiamata a **printf** in base al risultato del confronto.