COMPITO S10 L3 ASSABLY X86

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).

```
0×00001141 <+8>:    mov   EAX,0×20
0×00001148 <+15>:   mov   EDX,0×38
0×00001155 <+28>:   add   EAX,EDX
0×00001157 <+30>:   mov   EBP, EAX
0×0000115a <+33>:   cmp   EBP,0xa
0×0000115e <+37>:   jge    0×1176 <main+61>
0×0000116a <+49>:   mov   eax,0×0
0×0000116f <+54>:   call   0×1030 <printf@plt>
```

0x00001141 <+8>: mov (move) EAX, 0x20 = '32' in decimale
1) --> asegnazione di 0x20 a EAX

0x00001148 <+15>: mov (""""") EDX, 0x38 = '56' in decimale
2) --> asegnazione di 0x38 a EDX

0x00001155 <+28>: add (somma) EAX, EDX
3) --> somma di 'EAX + EDX' >>> '32 + 56 = 88'

0x00001157 <+30>: mov (move) EBP, EAX
4) --> asegnazione di '88' da EAX a EBP

0x0000115a <+33>: cmp (compara) EBP, 0xa = '10' in decimale
5) --> confronto tra '88 (des) - 10 (sor) = flag_78 zf0 cf0'

0x0000115e <+37>: jge (jump,x>=y) 0x1176='4470' <main+61>
6) --> condizione 4470 >= flag_78 else <main+61>='riferimento linea 61' = finisce

0x0000116a <+49>: mov (move) eax,0x0 = '0'
7) --> asegnazione di 0 a EAX

0x0000116f <+54>: call (chiama) 0x1030 = 'printf' <printf@plt>
8) --> richiamo funzione PRINTF

<+61> to continue...>>>> *starts play YES - roundbout*