

**Soluzione – Descrivere lo scopo delle istruzioni in Assembly riportate di seguito**

ISTRUZIONE	DESCRIZIONE
0x00001141 <+8>: mov EAX,0x20	Sposta il valore intero decimale 32 nel registro EAX
0x00001148 <+15>: mov EDX,0x38	Sposta il valore intero decimale 56 nel registro EDX
0x00001155 <+28>: add EAX,EDX	Somma il registro EDX ad EAX, praticamente somma 56 a 32 ed aggiorna il registro EAX con la somma = 88
0x00001157 <+30>: mov EBP, EAX	Muove il contenuto del registro EAX, ovvero 88 nel registro EBP
0x0000115a <+33>: cmp EBP,0xa	Controlla (CMP= compare) l'uguaglianza tra il valore 0xa che in decimale è il numero 10 con il valore contenuto in EBP ovvero 88
0x0000115e <+37>: jge 0x1176 <main+61>	Effettua un salto condizionale se la destinazione di «cmp» è maggiore o uguale del valore di controllo. Considerato che 88 > 10, il salto viene effettuato
0x0000116a <+49>: mov eax,0x0	Sovrascrive il valore di EAX con il valore 0, ovvero sposta 0 in EAX
0x0000116f <+54>: call 0x1030 <printf@plt>	Chiamata di funzione ad una funzione a noi nota, è la funzione printf