

ASSEMBLY X86

MATTIA PASTORELLI

COMANDA:

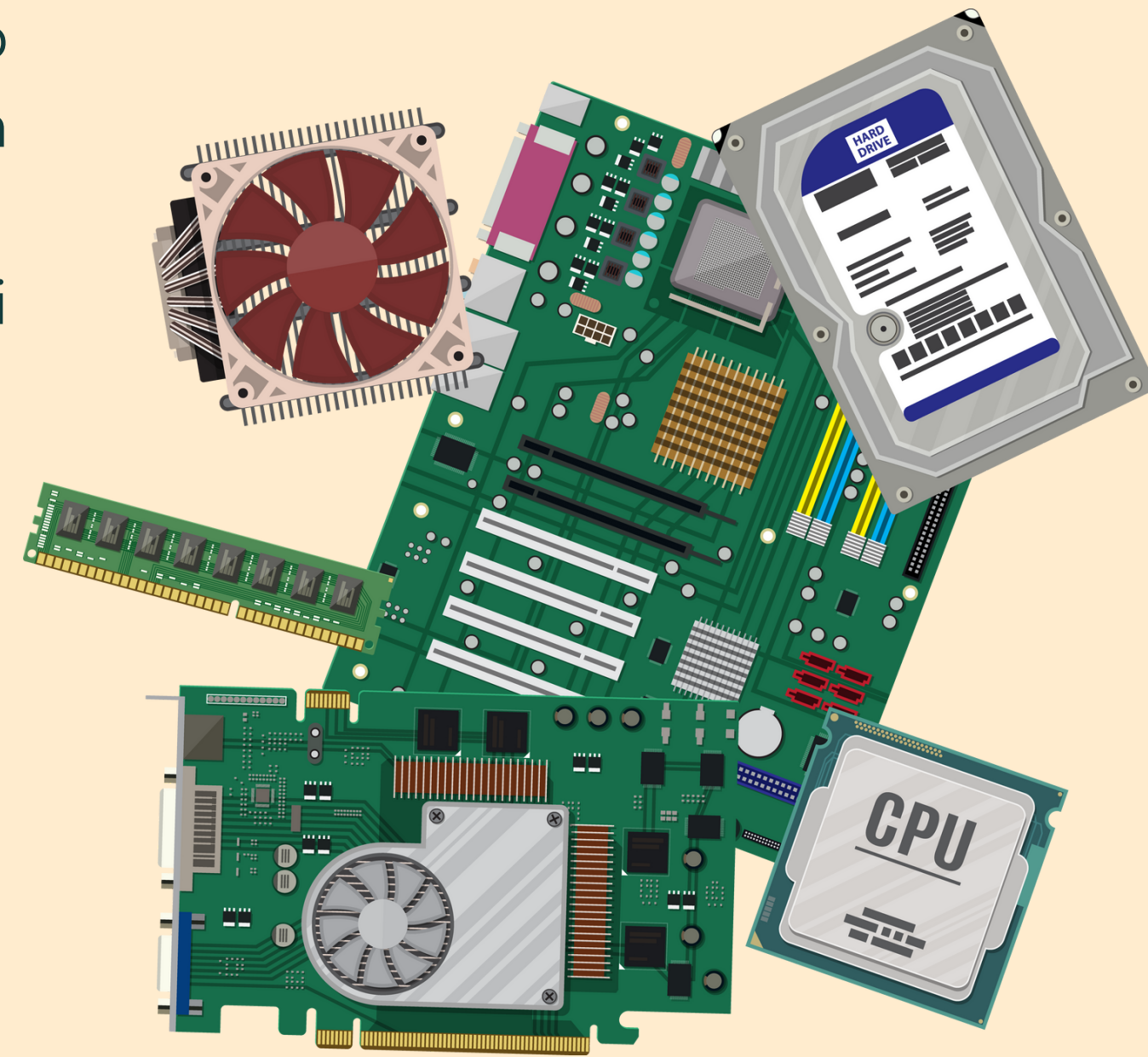


Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly. Dato il codice in Assembly per la CPU x86 allegato qui di seguito, identificare lo scopo di ogni istruzione, inserendo una descrizione per ogni riga di codice. Ricordate che i numeri nel formato 0xYY sono numeri esadecimali. 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>
```

Cos'è l'Assembly?

L'assembly rappresenta un linguaggio di programmazione a basso livello, posizionato direttamente sotto il linguaggio macchina in un computer. Spesso impiegato per scrivere codice che comunica direttamente con l'hardware del computer, gestendo dispositivi di input/output, memoria e registri della CPU.

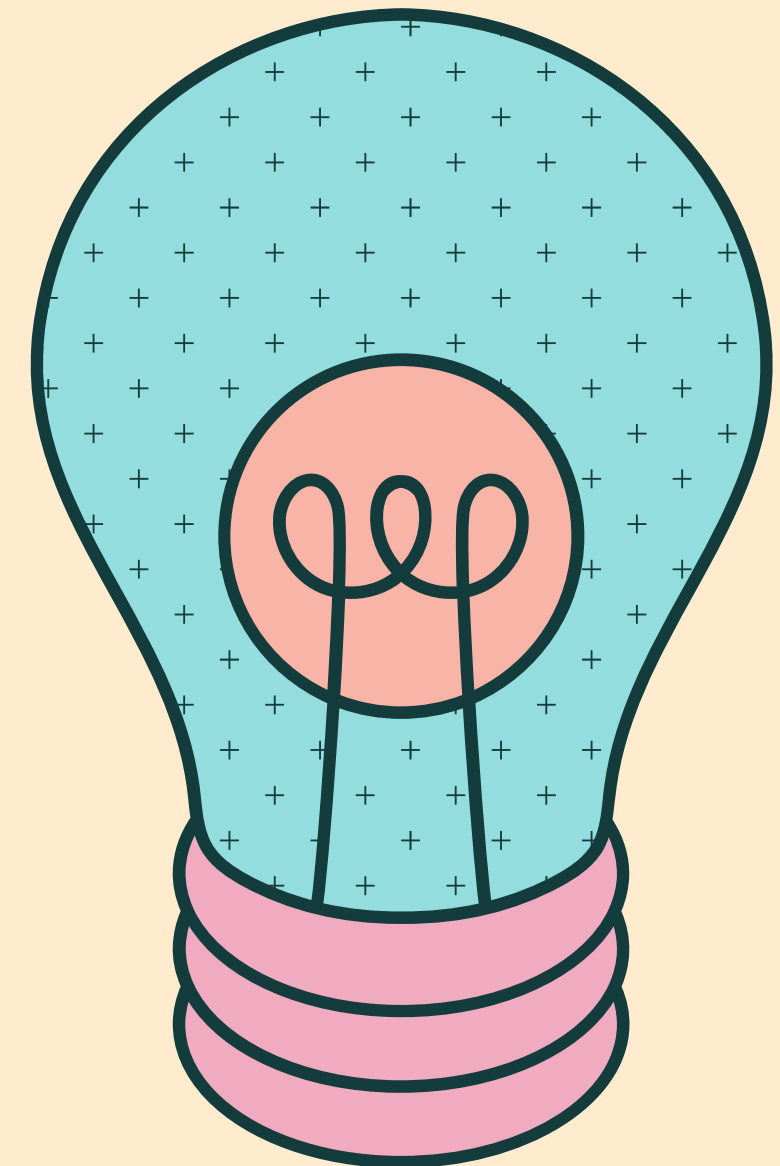


Spiegazione assembly

L'attività di oggi riguarda la conversione dei numeri esadecimali (0xYX) in decimali.

- 0x00001141 <+8>: mov EAX,0x20 = 32 in decimale.

Il valore 32 viene trasferito nel registro EAX, assegnandogli questo valore.

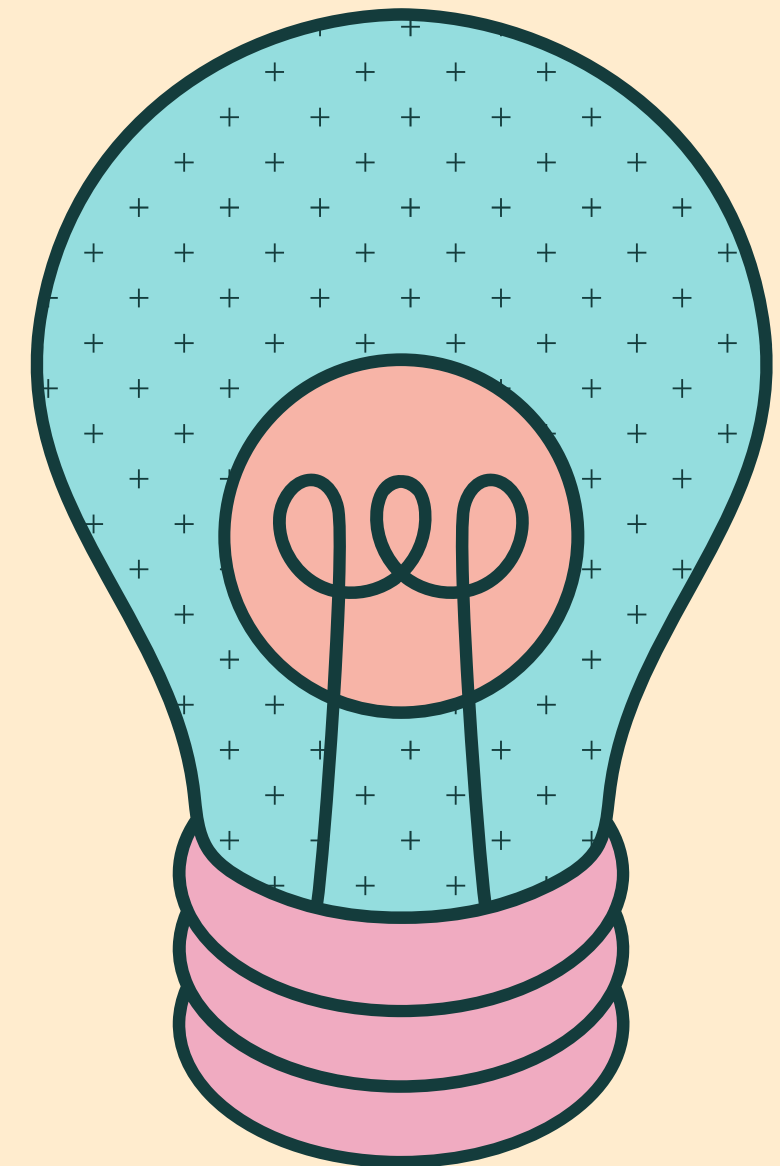


Spiegazione assembly

L'attività odierna consiste nella conversione dei numeri esadecimali (0xYX) in decimali.

- 0x00001148 <+15>: mov EDX,0x38 = 56 in decimale.

Il valore 56 viene caricato nel registro EDX



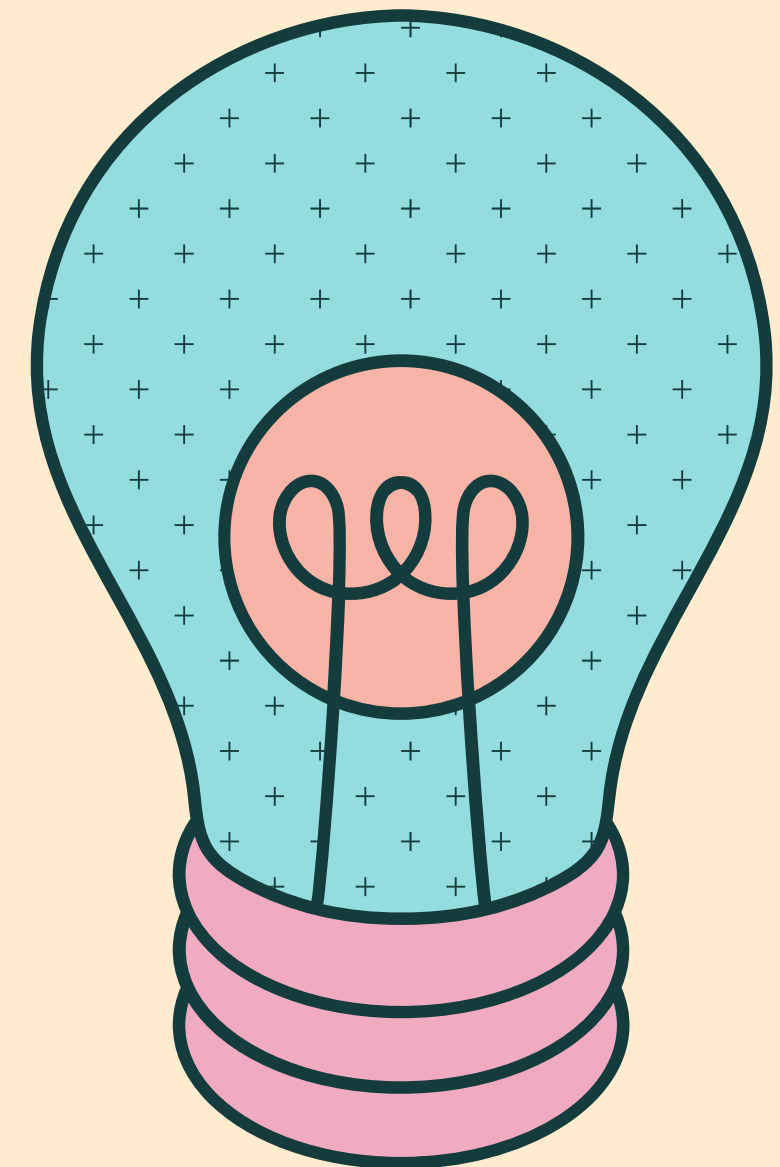
Spiegazione assembly

L'attività odierna consiste nella conversione dei numeri esadecimali (0xYX) in decimali.

- 0x00001155 <+28>: add EAX,EDX

Questa è una procedura di somma del valore assegnato al registro EDX (56) al valore del registro EAX (32).

Il risultato è salvato nel registro EAX

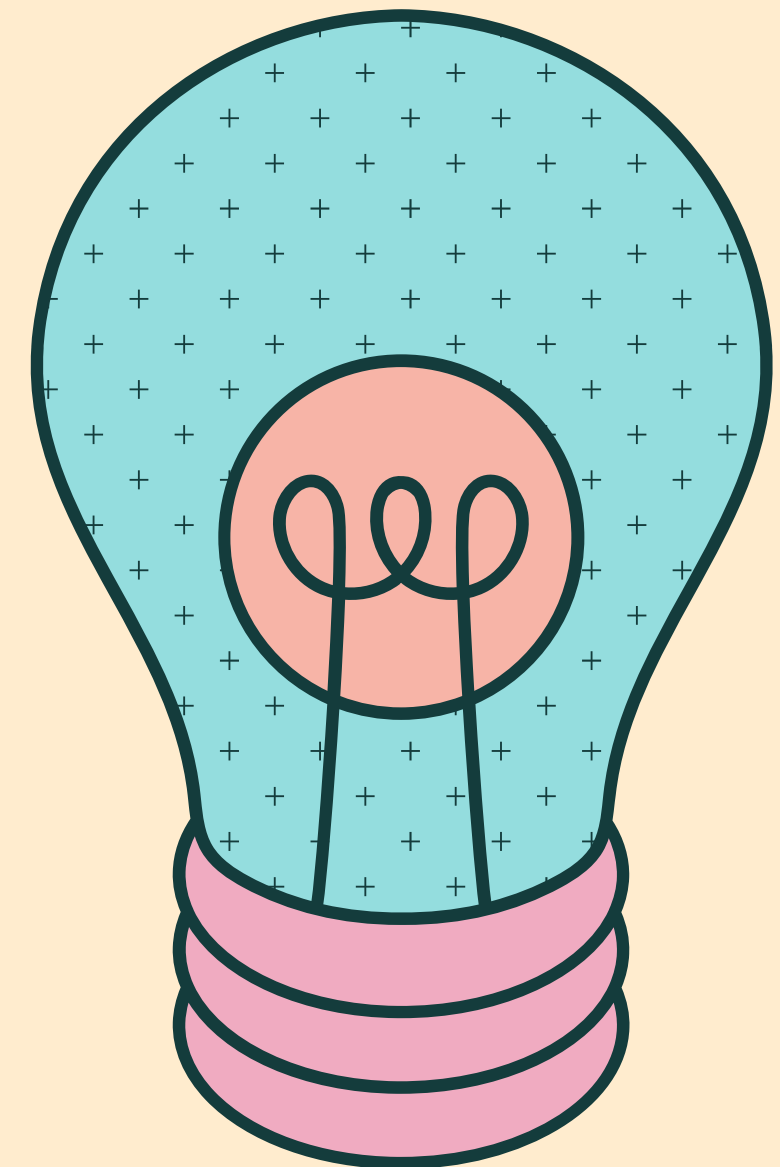


Spiegazione assembly

L'attività odierna consiste nella conversione dei numeri esadecimali (0xYX) in decimali.

- 0x00001157 <+30>: mov EBP, EAX

Questa procedura viene utilizzato per copiare/spostare il valore del registro EAX nel registro EBP.
Quindi assegneremo il risultato della somma precedente al registro EBP.

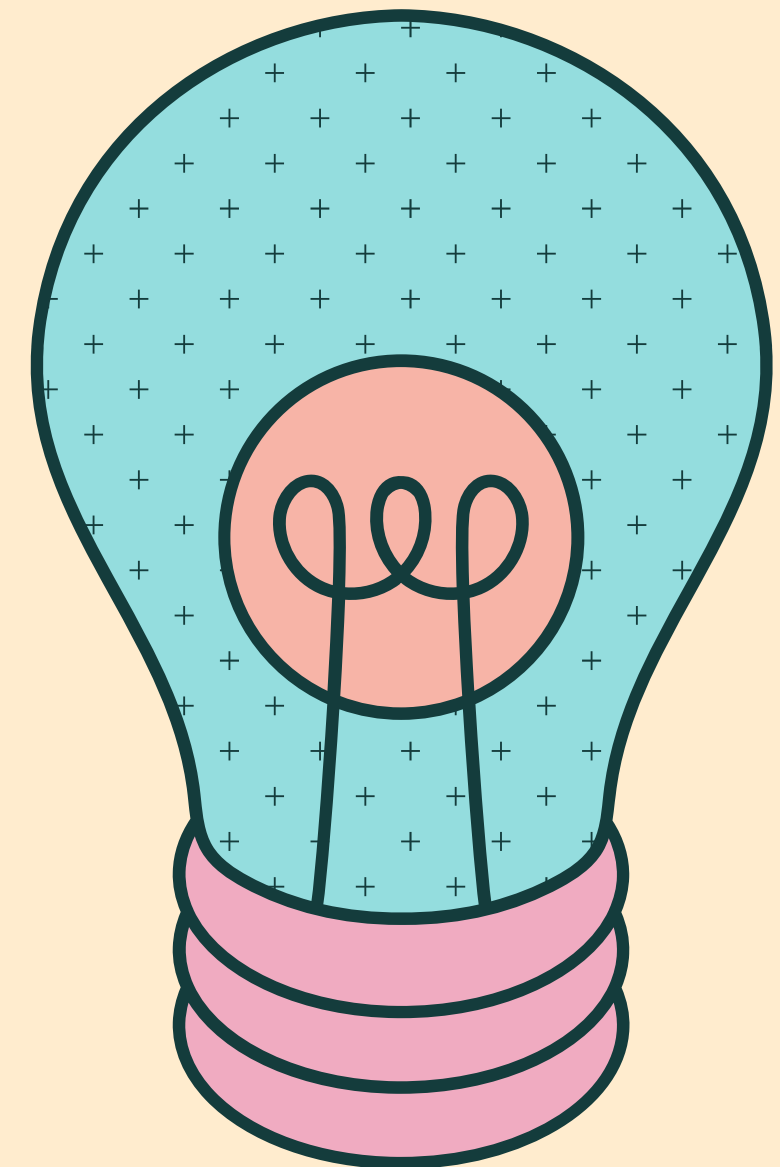


Spiegazione assembly

L'attività odierna consiste nella conversione dei numeri esadecimali (0xYX) in decimali.

- 0x0000115a <+33>: cmp EBP,0xa

Dopo aver attribuito il risultato al registro EBP, esso stesso verrà comparato con il valore 0xa (10)

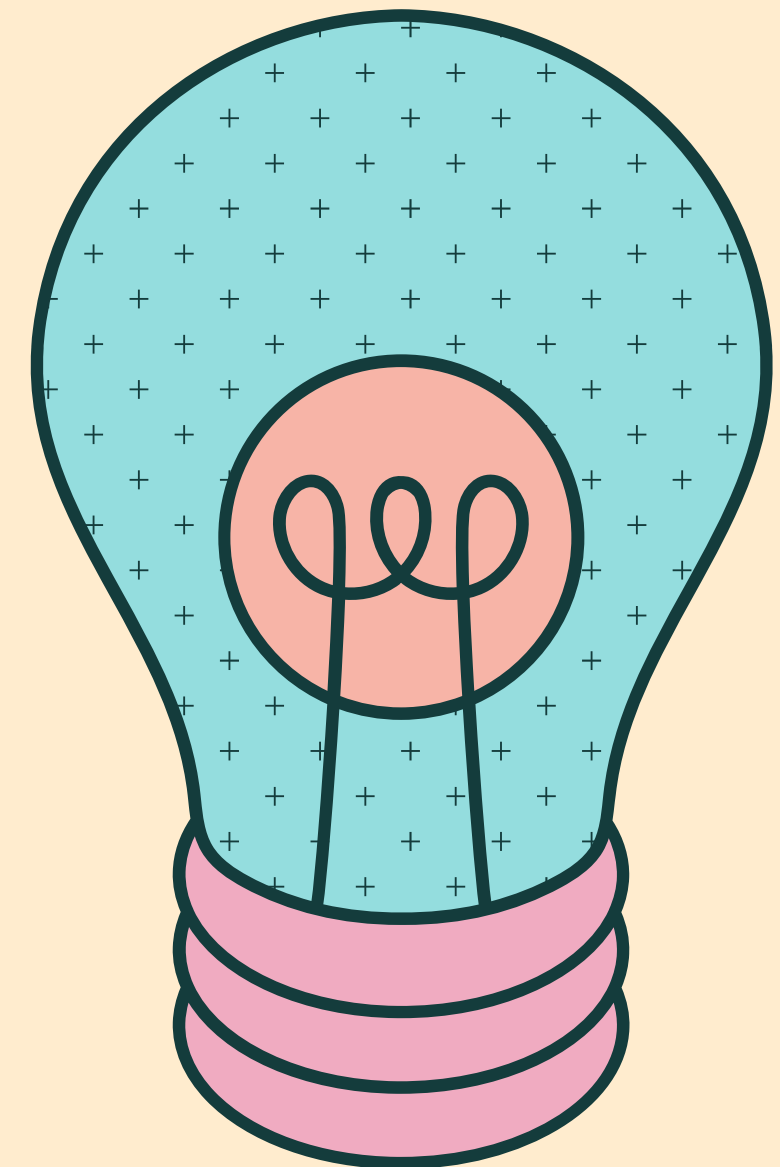


Spiegazione assembly

L'attività odierna consiste nella conversione dei numeri esadecimali (0xYX) in decimali.

- 0x0000115e <+37>: jge 0x1176 <main+61>

Questa è un'istruzione di salto, ovvero, se il risultato della somma EBP è maggiore o uguale a 10, il programma salta all'indirizzo 0x1176, altrimenti continua normalmente.

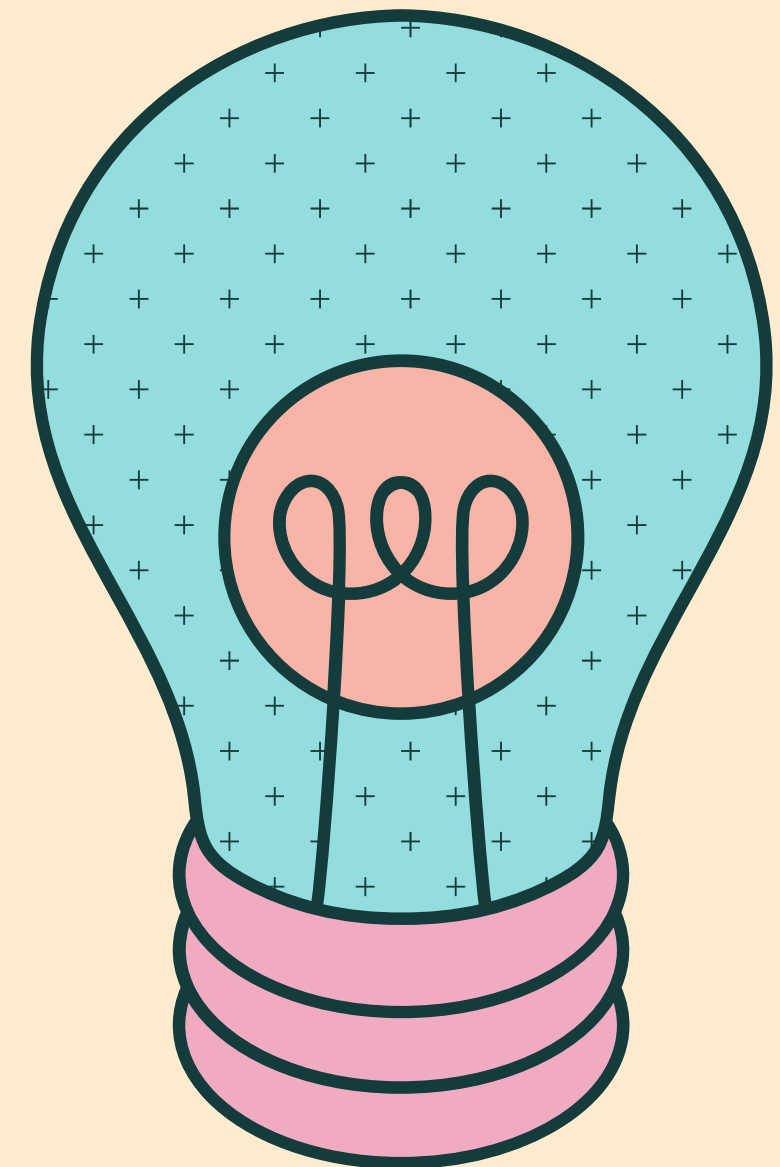


Spiegazione assembly

L'attività odierna consiste nella conversione dei numeri esadecimali (0xYX) in decimali.

- 0x0000116a <+49>: mov EAX,0x0

Qui il valore del registro di EAX viene portato a 0

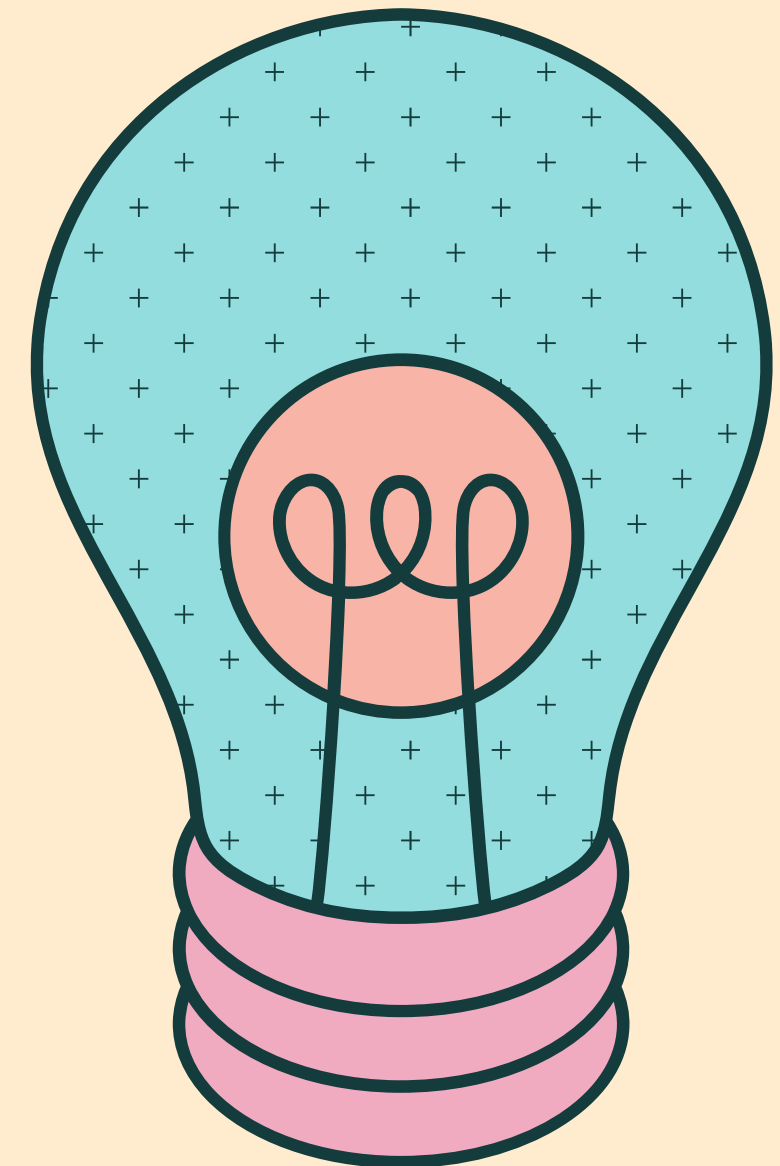


Spiegazione assembly

L'attività odierna consiste nella conversione dei numeri esadecimali (0xYX) in decimali.

- 0x0000116f <+54>: call 0x1030 <printf@plt>

Al termine del codice verrà stampato a schermo il risultato



RIASSUNTO

Il seguente codice assembly esegue l'addizione dei numeri 32 e 56, salvando il risultato in un registro (EBP), confrontandolo con 0xA e, se il risultato è inferiore a 0xA (10), stampa il risultato usando la funzione printf.

