

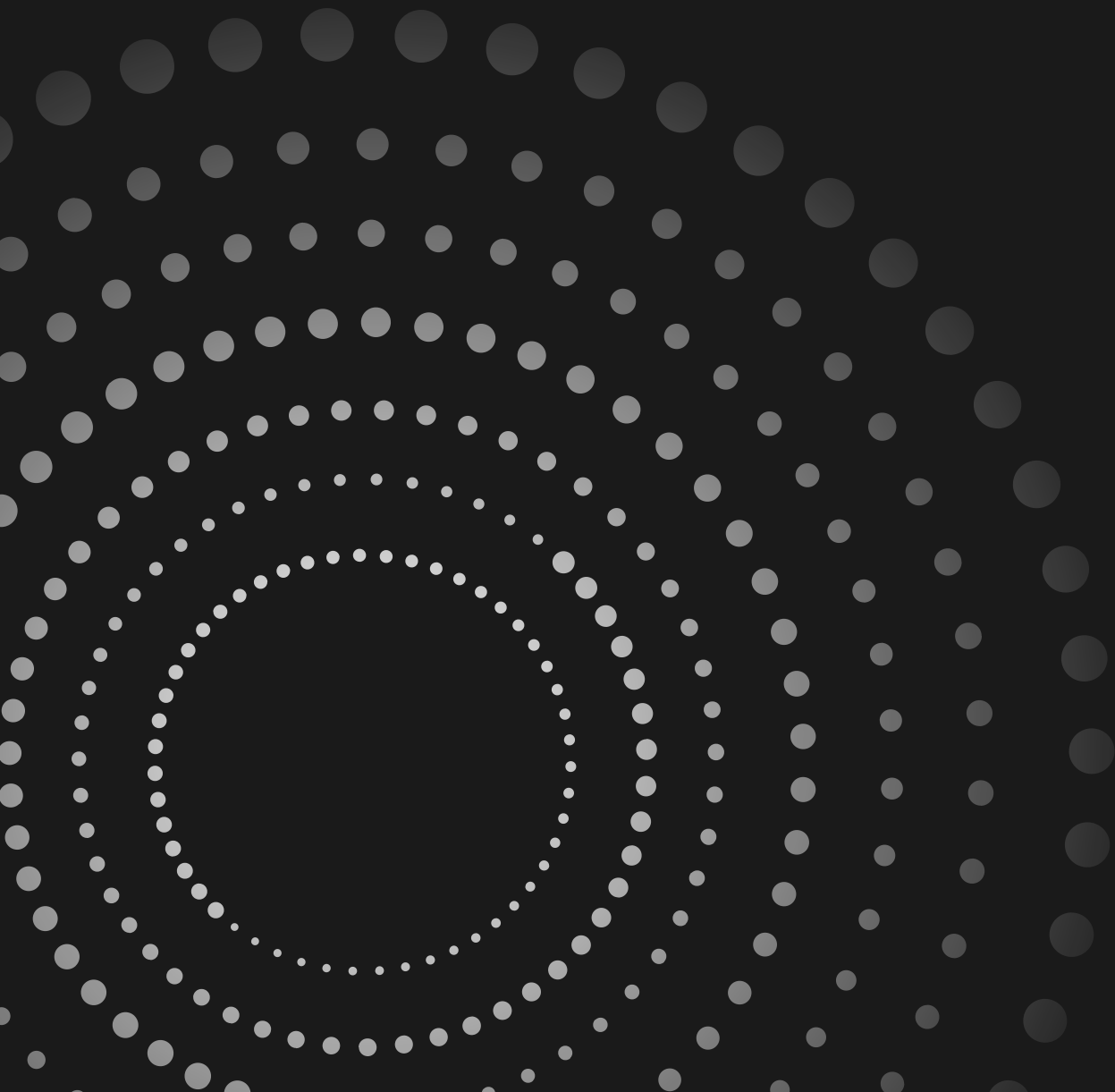
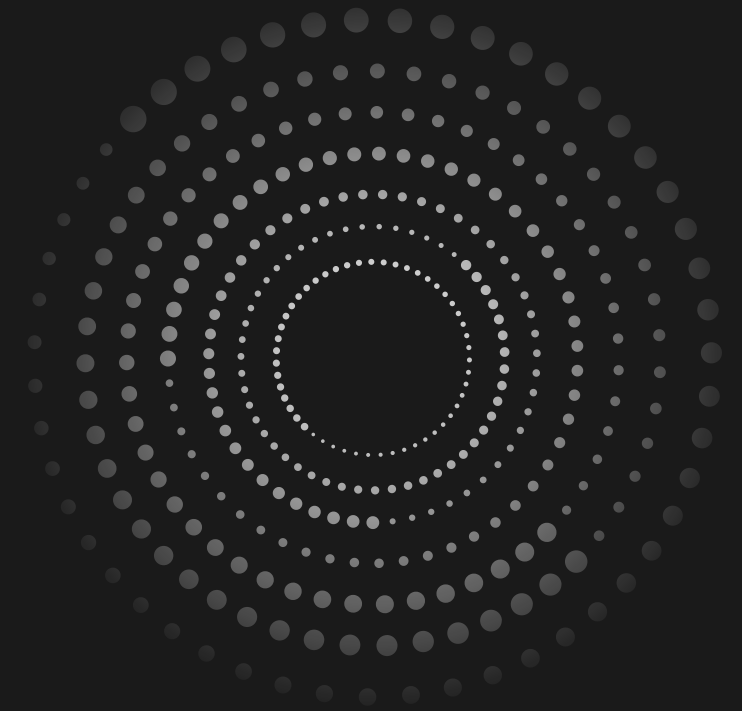
S10 - L3

TEAM

Iosif Castrucci

Donato Tralli

Gianpaolo Miliccia Mendoza



Traccia:

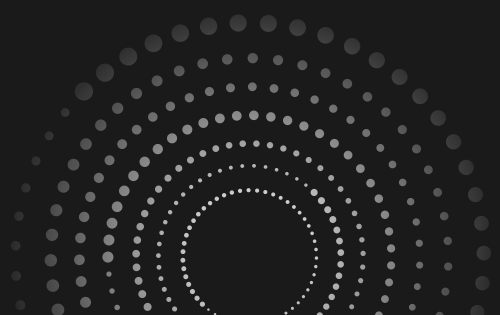
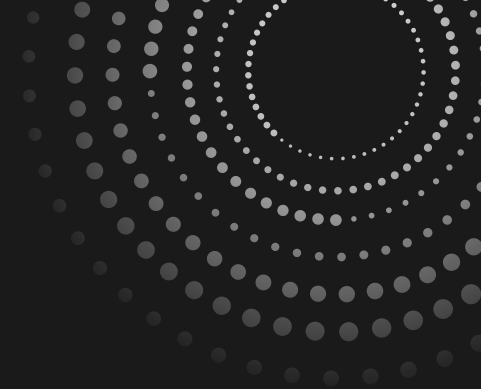
Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly. Dato il codice in Assembly per la CPU x86 allegato qui di seguito, identificare lo scopo di ogni istruzione, inserendo una descrizione per ogni riga di codice. Ricordate che i numeri nel formato 0xYY sono numeri esadecimali.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





0x00001141 <+8>: mov EAX,0x20	EAX = 0x20 (32 in decimale)
0x00001148 <+15>: mov EDX,0x38	EDX = 0x38 (56 in decimale)
0x00001155 <+28>: add EAX,EDX	EAX = EAX + EDX (32 + 56 = 88)
0x00001157 <+30>: mov EBP,EAX	EBP = EAX (EBP = 88)
0x0000115a <+33>: cmp EBP,0xa	Confronta EBP con 0xa (10 in decimale)
0x0000115e <+37>: jge 0x1176 <main+61>	Se EBP >= 10, salta all'indirizzo 0x1176
0x0000116a <+49>: mov eax,0x0	EAX = 0x0 (resetta EAX a 0)
0x0000116f <+54>: call 0x1030 <printf@plt>	Chiama la funzione printf



SPIEGAZIONE CODICE ASSEMBLY

0x00001141 <+8>: mov EAX,0x20

- Carica il valore esadecimale 0x20 (32 in decimale) nel registro EAX.
- Commento: EAX = 0x20 (32 in decimale)

0x00001148 <+15>: mov EDX,0x38

- Carica il valore esadecimale 0x38 (56 in decimale) nel registro EDX.
- Commento: EDX = 0x38 (56 in decimale)

0x00001155 <+28>: add EAX,EDX

- Somma il valore nel registro EDX al valore nel registro EAX e memorizza il risultato in EAX.
- Commento: EAX = EAX + EDX (32 + 56 = 88)

0x00001157 <+30>: mov EBP,EAX

- Copia il valore del registro EAX nel registro EBP.
- Commento: EBP = EAX (EBP = 88)

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

- Confronta il valore nel registro EBP con il valore esadecimale 0xa (10 in decimale).
- Commento: Confronta EBP con 0xa (10 in decimale)

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

- Se il valore di EBP è maggiore o uguale a 10, salta all'indirizzo 0x1176.
- Commento: Se EBP >= 10, salta all'indirizzo 0x1176

0x0000116a <+49>: mov eax,0x0

- Carica il valore 0x0 (0 in decimale) nel registro EAX.
- Commento: EAX = 0x0 (resetta EAX a 0)

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

- Chiama la funzione printf, che si trova all'indirizzo 0x1030.
- Commento: Chiama la funzione printf