

Esercizio S10-L3

L'esercizio di oggi prevedeva di illustrare e fornire una descrizione di un codice Assembly.

```
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>
```

La prima riga sta a significare che il valore esadecimale 0x20 (32) viene spostato/copiato nel registro EAX.

La seconda riga muove/sposta il valore 0x38 (56) nel registro EDX.

Nella terza viene usato add, ossia una somma, viene quindi aggiunto il contenuto di EDX a EAX.

La quarta prevede lo spostamento del risultato della somma (88) in EBP.

Nella quinta viene usato cmp, compara quindi il valore di EBP con 0xa (10).

Nella sesta viene usato jge che serve a “saltare” delle righe di codice se il valore di EBP (88) è maggiore o uguale a 10. Dato che ciò è vero si salterà direttamente all'indirizzo di memoria segnato.

Nella settima viene spostato il valore 0x0 (0) nel registro EAX.

Nell'ultima viene chiamata la funzione “printf”.