Gabriele Di giampietro **Esercitazione 11/01/2023**

Obbiettivo:

Identificare lo scopo di ogni istruzione del codice seguente in Assembly per la CPU x86, inserendo una descrizione per ogni riga del codice.

```
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

Attraverso l'istruzione **mov** andiamo a prendere la sorgente con valore 0x20(esadecimale) o 32(decimale) e viene inserito nella destinazione che è il registro di memoria EAX.

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

Attraverso l'istruzione **mov** andiamo a prendere la sorgente con valore 0x38(esadecimale) o 56(decimale) e viene inserito nella destinazione che è il registro di memoria EDX.

0x00001155<+28>: add EAX, EDX

Attraverso l'istruzione **add** somma il valore della sorgente EDX e della destinazione EAX, che sappiamo dalle righe di istruzione precedenti sia 56+32=88(decimale) oppure 0x38+0x20=58 (esadecimale).

0x00001157<+30>: mov EBP, EAX

Attraverso l'istruzione **mov** andiamo a prendere la sorgente EAX con valore 88 (decimale) o 58 (esadecimale) e viene inserito nella destinazione che è il registro di memoria EBP.

0x0000115a <+33>: cmp EBP,Oxa

Attraverso il comando **cmp** andiamo a fare un'istruzione simile a **sub**, ovvero simile ad una sottrazione, senza aggiornare il valore di EBP. Questo comando serve per verificare o meglio comparare se il numero è 0, ha avuto un riporto oppure nessuno dei due. Nel nostro caso abbiamo: EBP 88 >10 (deciamale) oppure 0xa (esadecimale) che ci setta la Zero Flag a 0 e la Carry Flag a 0.

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

Attraverso il comando **jge** andiamo a fare un salto nell'indirizzo di memoria 0x1176 solo nel caso in cui la destinazione sia di valore maggiore o uguale alla sorgente nell'istruzione cmp.

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

Attraverso l'istruzione **mov** andiamo a prendere la sorgente con valore 0x0(esadecimale) o 0(decimale) e viene inserito nella destinazione che è il registro di memoria EAX.

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

Attraverso l'istruzione **call** andiamo ad effettuare una chiamata sull'allocazione di memoria 0x1030 nella quale è corrispondete alla funzione di print, ovvero di una stampa a schermo di un codice e di una sotto funzione.