

```

1 # Conteggio del numero di occorrenze di un numero in un array
2 # Differenze rispetto al programma fatto con Zippo:
3 # - Indico in ingresso gli elementi dell'array
4 # - Indico in ingresso il numero di cui voglio verificare le occorrenze
5 # Il numero di elementi da controllare è indicato nella costante numero_elementi
6
7 .GLOBAL _main
8 .INCLUDE "C:/amb_GAS/utility"
9
10 .DATA
11 .SET numero_elementi, 9
12 array:      .FILL numero_elementi, 2 # Allocazione array: elementi da 2byte
13 conteggio:  .BYTE 0x00 # Conteggio delle corrispondenze
14
15 .TEXT
16 # Inizializzo CL, che sarà il mio contatore.
17 # AX è il registro che conterrà i numeri decimali inseriti in ingresso con la
  indecimal_word
18 # ESI lo utilizzo per scorrere l'array e individuare se ho visitato tutte le
  posizioni
19
20 _main:      NOP
21             MOV $0, %CL
22             MOV $0, %ESI      # esi = 0
23
24 popolamento:  CMP $numero_elementi, %ESI
25               JE numdacontrollare
26
27               CALL indecimal_word
28               # Caldamente consigliata, situa orrenda senza
29               CALL newline
30
31               MOV %AX, array(, %ESI, 2)
32
33               INC %ESI
34               JMP popolamento
35
36 # Si estrae il numero di cui si vogliono trovare le corrispondenze. Anche in questo
  caso si guarda AX
37 # Confronto AX, dove è presente il numero di cui vogliamo le occorrenze, con
  l'elemento dell'array in posizione ESI
38 # Se non c'è uguaglianza salto l'incremento di CL
39 # NB: La ESI è adeguatamente resettata dopo l'uso precedente, idem la AX che viene
  sovrascritta con la indecimal_word
40
41 numdacontrollare:  CALL indecimal_word
42                   CALL newline
43 reset:            MOV $0, %ESI
44 comp:
45                   CMP $numero_elementi, %ESI
46                   JE fine
47
48                   CMPW array(, %ESI, 2), %AX
49                   JNE poi
50
51                   INC %CL
52
53 poi:             INC %ESI
54                   JMP comp
55

```

```
56 # Sposto il risultato in AL per poter effettuare la stampa del risultato con il
   sottoprogramma
57 fine:      MOV %CL, %AL
58           CALL outdecimal_byte
59
60           # Pongo per evitare la stampa di "Program exited with code X"
61           XOR %EAX, %EAX
62
63           RET
64
```