

## Esercizi in preparazione della verifica individuale di laboratorio

### Esercizio 1

Si scriva il codice assembleativo per il calcolo dell'espressione:  $x = a + b + 2$

### Esercizio 2

Commentare il seguente programma assembly che verifica se un numero  $n > 1$  è primo (in ambiente Jasmin) e proporre corrispondente algoritmo in *pseudocodice*

```
; Scrivere un programma assembly che verifica se un numero n >1 è primo
; 1 significa numero primo
; 0 non primo
num:
dd 4          ; verificare bug ..... ricordando che 2 è primo e proporre soluzione
primo:
dd 1          ; primo
nonprimo:
dd 0         ; non primo

MOV AX, [num]
MOV BX, AX
SUB BX, 1
uno:
DIV BX
CMP DX,0
JE due
MOV DX,0
MOV AX,[num]
DEC BX
CMP BX,1
JG uno
mov dx, primo          ; Il numero è primo
JMP tre ; Salta alla fine
due:          ; Il numero NON è primo
mov dx, nonprimo
JMP tre ; Salta alla fine
tre:
mov bx, [edx] ; salva in bx il contenuto della cella
                ; puntata da 0x4 se primo cioè 1
                ; puntata da 0x8 se non primo cioè 0
mov [0xC], bx
```

## Soluzione

; Scrivere un programma assembly che verifica se un numero  $n > 1$  è primo  
; 1 significa numero primo  
; 0 non primo

num:  
dd 2 ; verificare bug ..... ricordando che 2 è primo  
; proponendo soluzione (\*)

primo:  
dd 1 ; primo  
nonprimo:  
dd 0 ; non primo .. contenuto della cella con indirizzo 0x8

MOV AX, [num] ; Copia del numero in AX

**CMP AX ,2 ; soluzione bug (\*)**

**JE pari**

MOV BX, AX ; Inizializza BX ....

SUB BX, 1 ; ... ad AX-1

uno:

DIV BX ; Divide AX per BX

CMP DX,0 ; Controlla il resto della divisione

JE due ; Se il resto è zero, salta a due

MOV DX,0

MOV AX,[num] ; Ripristina AX

DEC BX ; Decremento BX

CMP BX,1 ; Confronto BX con 1

JG uno

pari:

mov dx,**primo** ; Il numero è primo

JMP tre ; Salta alla fine

due: ; Il numero NON è primo

mov dx,**nonprimo**

JMP tre ; Salta alla fine

tre:

mov bx, **[edx]** ; salva in bx il contenuto della cella puntata

; da 0x4 se primo cioè 1

; da 0x8 e non primo cioè 0

mov [0xC], bx

Memory					
desc	hex	highlight	8 Bit	16Bit	32Bit
address	signed int	unsigned int	hex		
0x0	2	2	0x00000002		
0x4	1	1	0x00000001		
0x8	0	0	0x00000000		
0xC	1	1	0x00000001		

numero N = 2

Numero primo

EDX: 4

cioè indirizzo 0x4 della cella di memoria che contiene il valore da copiare in 0xC .... 1 perché *numero primo*