

Ambiente DEV-CPP

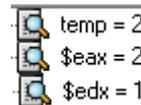
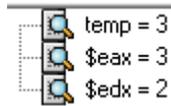
```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    int a = 5;
    int b = 2;
    int temp = 0;

    temp = a+b;
    temp = a-b; // in DX il contenuto di b
    temp = a*b;
    temp = a/b; // contenuto DX 0 senza riporto ed 1 con riporto

    printf("finito .... premi tasto ....");
    getch();

    //system("PAUSE");
    return 0;
}
```



Ambiente JASMIN

; operazioni aritmetiche: somma , sottrazione, moltiplicazione e divisione

inizializzazione:

```
MOV [0x04], 5; ' variabile a '
MOV [0x08], 2; ' variabile b'
MOV [0x0C], 0; ' variabile temp'
```

somma:

```
MOV eax, [0x04]; copio in accumulatore il valore della "variabile a"
ADD eax, [0x08]; sommo
MOV [0x0c], eax; copio in ' variabile temp'
```

sottrazione:

```
MOV edx, [0x08]; copio in D il valore della "variabile b"
MOV eax, [0x04]; copio in accumulatore il valore della "variabile a"
SUB eax, edx; sottrazione
MOV [0x0c], eax; copio in ' variabile temp'
```

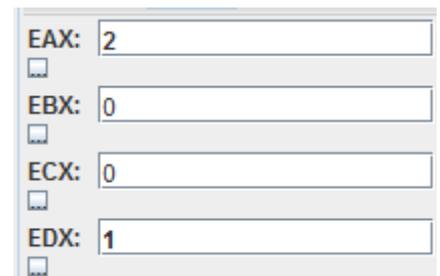
moltiplicazione:

```
MOV eax, [0x04]; copio in accumulatore il valore della "variabile a"
IMUL eax, [0x08]; moltiplicazione ..... anche MUL [0x08]     sintassi: MUL DEST
MOV [0x0c], eax; copio in ' variabile temp'
```

divisione:

```
MOV eax, [0x04]; copio in accumulatore il valore della "variabile a"
IDIV [0x08]; divisione ..... anche DIV [0x08]
MOV [0x0c], eax; copio in ' variabile temp'
MOV [0x0c], eax; copio in ' variabile temp'
```

si noti in EDX il valore 1 che indica riporto



Con **80386/486** (numero a 64 bit), se il divisore (sorgente) ha la dimensione di una doubleword; il risultato (quoziente) è lasciato in **EAX** e il resto in **EDX**.

Ambiente JASMIN

; operazioni aritmetiche

dd 5, 2

MOV EAX,[0x0]

MOV ECX,[0x4]

MOV EBX,[0x0]

SOMMA:

ADD EAX, ECX

MOV [0x8], EAX ; salvo il risultato in memoria

SOTTRAZIONE:

SUB EBX, ECX

MOV [0xC], EBX ; salvo il risultato in memoria

MOLTIPLICAZIONE:

IMUL ECX, [0x0]

MOV [0x10], ECX ; salvo il risultato in memoria

DIVISIONE:

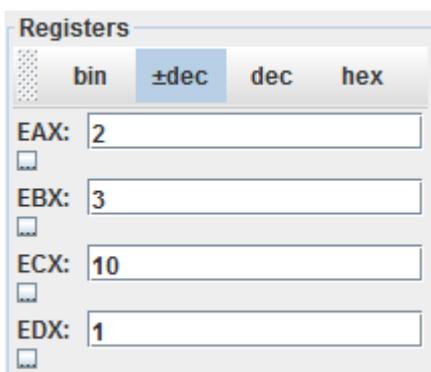
MOV EAX,[0x0]

IDIV [0x4]

MOV [0x14], EAX ; salvo il risultato in memoria

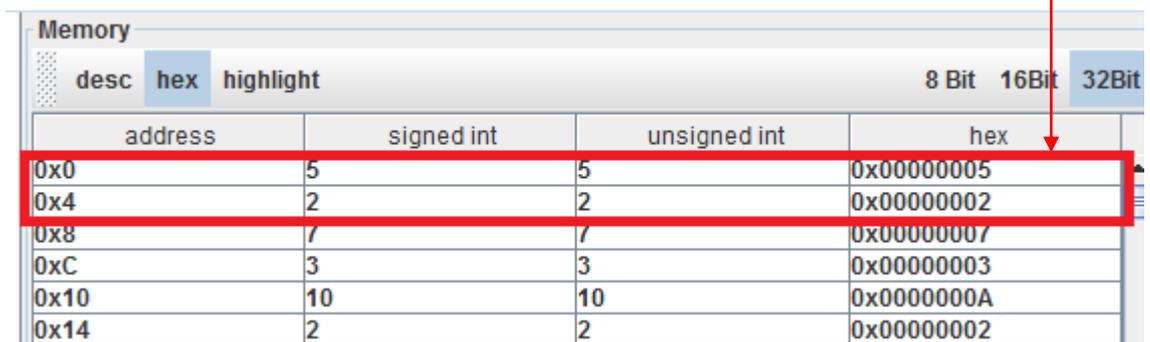
definizione di dati allocati a partire da pagina zero

Al termine dell'esecuzione:



	bin	±dec	dec	hex
EAX:		2		
EBX:		3		
ECX:		10		
EDX:		1		

la divisione è con resto



desc	hex	highlight	8 Bit	16Bit	32Bit
address	signed int	unsigned int	hex		
0x0	5	5	0x00000005		
0x4	2	2	0x00000002		
0x8	7	7	0x00000007		
0xC	3	3	0x00000003		
0x10	10	10	0x0000000A		
0x14	2	2	0x00000002		

Esercizio:

"proporre un programma in assembly per eseguire una divisione col metodo delle sottrazioni successive"

[soluzione](#)