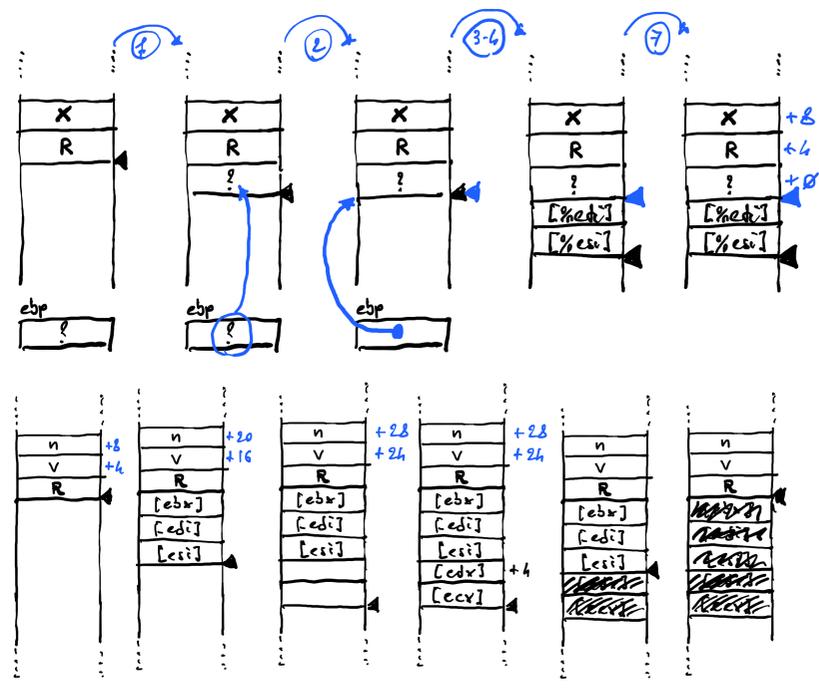


```
.globl f
f:
    pushl %edi
    pushl %ebp
    subl $4, %esp
    movl $1, %edi
    movl $2, %ebp
    movl 4(%esp), %eax
    incl %eax
    movl %eax, (%esp)
    call f
    addl %edi, %eax
    addl %ebp, %eax
    addl $4, %esp
    popl %ebp
    popl %edi
    ret
```

```
int f(int x) {
    int y=1;
    int z=2;
    int a = f(x+1);
    a = a + y + z;
    return a;
}
```

```
.globl f
f:
    1 pushl %ebp
    2 movl %esp, %ebp
    3 pushl %edi
    4 pushl %esi
    5 movl $1, %edi
    6 movl $2, %esi
    7 movl 8(%ebp), %eax
    incl %eax
    pushl %eax
    call f
    addl $4, %esp
    addl %edi, %eax
    addl %esi, %eax
    popl %esi
    popl %edi
    movl %ebp, %esp
    popl %ebp
    ret
```



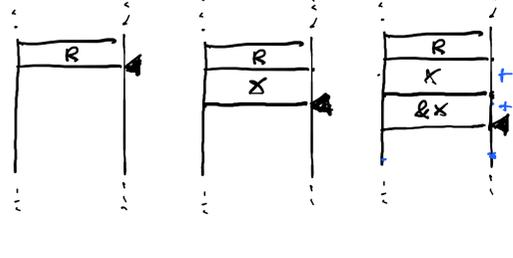
**VARIABILI LOCALI IN STACK**

Una variabile locale dichiarata in C può essere messa dal compilatore nella stack in 3 casi:

- 1 Non ci sono registri disponibili
- 2 È utilizzato l'operatore "address of" (&)
- 3 La variabile è un array o una struct

Esempio

```
int f() {
    1 int x;
    2 load (&x);
    3 return x;
}
```



```
.globl f
f:
    subl $8, %esp
    movl %esp, %eax
    addl $4, %eax
    movl %eax, (%esp)
    call load
    movl 4(%esp), %eax
    addl $8, %esp
    ret
```

leal 4(%esp), %eax  
 ↳ Load Effective Address

cal C(%S), %D → movl %S, %D  
 addl \$C, %D

↑  
 O(B, I, S)  
 B + I \* S + O → D

**CONVERSIONI DI TIPO (CASTING)**

1 OPERAZIONE DI **DOWNCAST** (Troncamento)

```
int c = 0x000CAFE; // 32 bit
char a = c; // 8 bit
```



**ATTENZIONE! PERDITA DI INFORMAZIONE !!!**

a → 0xFE

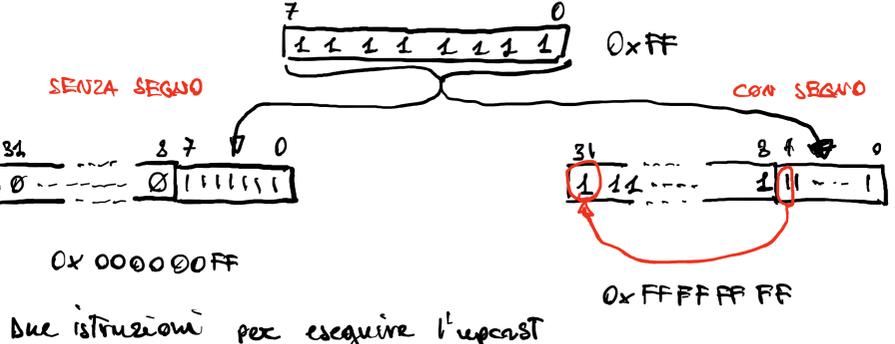
2 OPERAZIONE DI **UPCAST** (promozione)

```
SENZA SEGNO | char c = 's'; // 8 bit
              | int a = c; // 32 bit
SENZA SEGNO | unsigned char c = 's'; // 8 bit
              | unsigned int a = c; // 32 bit
```

**NESSUNA PERDITA DI INFORMAZIONE**

**ATTENZIONE: DOBBIAMO GARANTIRE CHE IL SEGNO ALMANCA VENUTA**

Esempio



Due istruzioni per eseguire l'upcast

```
movsxy S, D → copia x bit meno significativi dalla sorgente ad una destinazione grande y bit con estensione del segno
movzxy S, D → copia x bit meno significativi dalla sorgente ad una destinazione grande y bit senza estensione del segno
```

NOTA BENE: S non può essere un valore immediato  
 D può essere solo un registro

**CON SEGNO**

a = c	char c	short c	int c
char a	movb %cl, %al		
short a	movsbw %cl, %ax	movw %cx, %ax	
int a	movsbl %cl, %eax	movswl %cx, %eax	movl %cx, %eax

**SENZA SEGNO**

a = c	char c	short c	int c
uns-char a	movb %cl, %al		
uns-short a	movzwb %cl, %ax	movw %cx, %ax	
uns-int a	movzbl %cl, %eax	movzwl %cx, %eax	movl %cx, %eax