

Funzioni con numero di argomenti variabili (variadic arguments)

Che cosa sono ?

- Sono funzioni tipo printf, in cui il numero di argomenti che vengono passati alla funzione è variabile (ma ne deve avere almeno 1), es:
 - `printf(“%s %s\n”, “hello”, “world”);`
 - `printf(“%s %s – %d, %d, %d, %d\n”, “hello”, “world” 1,2,3,4);`
- Per indicare un numero di argomenti variabile si utilizza '...' (ellipsis) nella definizione della funzione. L'ellipsis deve essere l'ultimo elemento:
 - `void myfunc(int c, ...);`
 - `long myfunc2(int a, char b, ...)`

Elementi

- `#include <stdarg.h>`
- All'interno della funzione bisogna definire una **va_list** che punta agli elementi aggiuntivi e non specificati della funzione (`va_list va;`)
- La macro **va_start** inizializza la variabile 'va' per puntare al primo elemento della lista avendo specificato l'ultimo valore noto (`va_start(va, arg);`)
- La macro **va_arg** restituisce il prossimo valore della lista castandolo al tipo specificato (`va_arg(va, int);`)
- La macro **va_end** va chiamata prima dell'uscita della funzione per effettuare il cleanup (`va_end(va);`)

Esempi

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

// somma di più argomenti
int vsum(int count, ...) {
    va_list ap;
    va_start (ap, count); // inizializzo la lista

    int sum = 0;
    for (int i=0; i<count; i++)
        sum += va_arg (ap, int); // il prossimo

    va_end (ap); // ho finito di usare la lista
    return sum;
}

int main () {
    printf("%d\n", vsum(3, 1,2,3));
    printf("%d\n", vsum(5, 1,2,3,4,5));
    return 0;
}
```

```
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>

// funzione di utilità per stampe di debugging
int debug(char *str, ...) {
    const char dbg[]="DBG: ";
    va_list argp;
    size_t size = strlen(str)+strlen(dbg)+1024;
    char * p=malloc(size);
    assert(p);
    strncpy(p,dbg,strlen(dbg));
    strcpy(p+strlen(dbg), str);
    va_start(argp, str);
    vsnprintf(p, size, str, argp);
    va_end(argp);
    fprintf(stderr, "%s", p);
    free(p);
}

int main () {
    debug("msg %s %d\n", "MESSAGGIO",1);
    debug("msg diverso %d %d\n", 1,1);
    return 0;
}
```

Tipi degli argomenti

- Attenzione al tipo degli argomenti variabili:
 - *char* e *short* vengono promossi ad *int*
 - *float* vengono promossi a *double*
 - per i vettori va usato il puntatore al tipo base
 - per passare puntatori a funzioni usare *typedef* per definire il tipo funzione ed utilizzare tale tipo