FUNZIONI DI LIBRERIA

ASSERT

Macro che prende in ingresso una espressione e termina il programma se l'espressione non è soddisfatta

```
#include <assert.h>

assert(exp);
```

FUNZIONI PER LA GESTIONE DI STRINGHE

```
#include <string.h>

size_t strlen(const char * s);          // calcola dimensione stringa
char * strcpy(char * dst, const char * src);   // copia
char * strcat(char * dst, const char * src);   // concatena
```

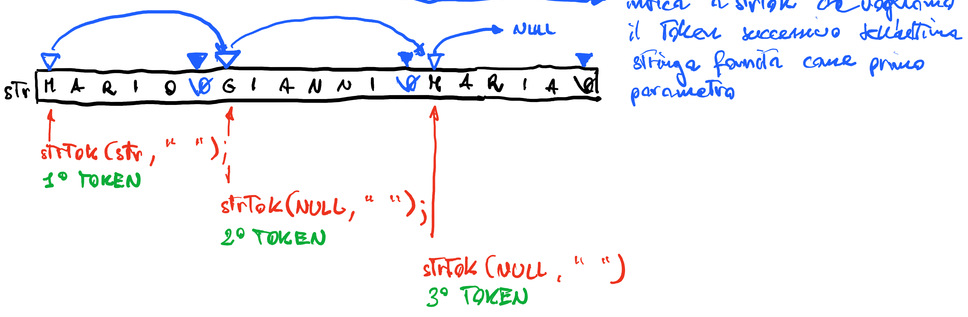
ATTENZIONE: dst deve contenere abbastanza spazio libero per contenere tutta src.

```
char * strtok(char * str, const char * delim);   // divide la stringa in Token
```

UTILIZZO:

```
char * token = strtok(str, delim);
while (token != NULL) {
    ... codice che fa qualcosa con token ...
    token = strtok(NULL, delim)
}
```

- str: stringa da "Tokenizzare"
- delim: stringa che contiene i delimitatori.
- NULL: indica a strtok che vogliamo il Token successivo della stringa fornita come primo parametro




```
int strcmp(const char * s1, const char * s2)
```

confronta s1 con s2 e restituisce
- < 0 se s1 precede s2
- 0 se s1 == s2
- > 0 altrimenti

```
#include <stdlib.h>
int atoi(const char * s)
```

converte s in un valore intero se possibile.

Es.:
- atoi("10"); → 10
- atoi("-10"); → -10
- atoi("10kg"); → 10
- atoi("euro50"); → 0

```
#include <stdio.h>
int sscanf(const char * s, const char * format, ...);
```

Estrae da s valori per popolare le variabili passate come parametri 3, 4, ... secondo format.

```
int sprintf(char * d, const char * format, ...)
```

Scrive in d la stringa format sostituendo ai caratteri jolly "%..." i valori delle variabili passate come parametri 3, 4, ...

```
int getchar();
```

Legge un carattere da tastiera

```
int scanf(const char * format, ...)
```

Legge da tastiera secondo format