



# Laurea triennale in Informatica

*modulo (CFU 6) di*

## Programmazione II e Lab.

**prof. Mariarosaria Rizzardi**

Centro Direzionale di Napoli – Isola C4

stanza: n. 423 – IV piano Lato Nord

tel.: 081 547 6545

email: [mariarosaria.rizzardi@uniparthenope.it](mailto:mariarosaria.rizzardi@uniparthenope.it)

The background features a large, light blue circular seal of the University of Apulia. The seal contains a central shield with a figure, surrounded by the text 'UNIVERSITA' DEGLI STUDI DI APOLOGIA' and 'PARTHENOPE'. Above the shield is '1920 - 2020' and below it is '100° ANNIVERSARIO'.

# ➤ **Approfondimento sulla funzione `fgets()`**

# Esempio 1a: input di una stringa statica

Di solito, si sovradimensiona l'array che conterrà la stringa.

```
#include <stdlib.h>
#include <stdio.h>
#define MAXSIZE 100

int main()
{
    char str[MAXSIZE+1];
    printf("Enter a string: ");
    fgets(str, MAXSIZE, stdin);
    printf("Entered string: [%s]\n", str);
    printf("strlen(string) = %u\n", (unsigned int)strlen(str));

    return 0;
}
```

123456789012  
Enter a string: ciao amico!  
Entered string: [ciao amico!  
] ← new line???  
strlen(string) = 12

# C: funzione `fgets()`

[da: <http://www.cplusplus.com/reference/cstdio/fgets/>]

```
char* fgets(char* str, int num, FILE* stream);
```

Reads characters from stream and stores them as a C string into **str** until

- (**num-1**) characters have been read
- or either
- a newline or the end-of-file is reached, whichever happens first.

A **newline character** makes fgets stop reading, but it is considered a valid character by the function and included in the string copied to **str**.

# Esempio 1b: come risolvere il problema?

```
#include <stdlib.h>
#include <stdio.h>
#define MAXSIZE 100

int main()
{
    char str[MAXSIZE+1];
    printf("Enter a string: ");
    fgets(str, MAXSIZE+1, stdin);
```

```
    if ( strlen(str) < MAXSIZE )
        str[strlen(str)-1] = '\0';
```

mette il terminatore  
al posto giusto!

```
    printf("Entered string: [%s]\n", str);
    printf("strlen(string) = %u\n", (unsigned int)strlen(str));
```

```
    return 0;
```

```
}
```

123456789012

Enter a string: ciao amico!

Entered string: [ciao amico!]

strlen(string) = 11

OK!

## Esempio 2a: input di una stringa dinamica

Se non si conosce prima la dimensione della stringa, di solito, si sovradimensiona l'array che conterrà la stringa.

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

int main()
{
    unsigned int MAXSIZE = 100;
    char* str = (char*)malloc(MAXSIZE);
    printf("Enter a string: ");
    fgets(str, MAXSIZE, stdin);
    printf("Entered string: [%s]\n", str);
    printf("strlen(string) = %u\n", (unsigned int)strlen(str));

    return 0;
}
```

123456789012  
Enter a string: ciao amico!  
Entered string: [ciao amico!  
] ← new line???  
strlen(string) = 12

# Esempio 2b: soluzione al problema

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

```
int main()
{
    unsigned int MAXSIZE = 100;
    char* str = (char*)malloc(MAXSIZE+1);
    printf("Enter a string: ");
    fgets(str, MAXSIZE+1, stdin);
```

```
    if ( strlen(str) < MAXSIZE )      mette il terminatore
        str[strlen(str)-1] = '\0';    al posto giusto!
```

```
    str=(char*)realloc(str, strlen(str)+1); rialloca accorciando
```

```
    printf("Entered string: [%s]\n", str);
    printf("strlen(string) = %u\n", (unsigned int)strlen(str));
```

```
    return 0;
```

```
}
```

123456789012

```
Enter a string: ciao amico!
Entered string: [ciao amico!]
strlen(string) = 11
```

OK!