

Varianti della printf()

- Le funzioni nella famiglia printf() producono output in base a una stringa di formato. Le funzioni printf() e vprintf() scrivono l'output su stdout; fprintf() e vfprintf() scrivono l'output nel flusso di output specificato; sprintf(), snprintf(), vsprintf() e vsnprintf() scrivono nella stringa di caratteri str.

```
#include <stdio.h>
int printf(const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
int dprintf(int fd, const char *format, ...);
int sprintf(char *str, const char *format, ...);
int snprintf(char *str, size_t size, const char *format, ...);
```

```
#include <stdarg.h>
int vprintf(const char *format, va_list ap);
int vfprintf(FILE *stream, const char *format, va_list ap);
int vdprintf(int fd, const char *format, va_list ap);
int vsprintf(char *str, const char *format, va_list ap);
int vsnprintf(char *str, size_t size, const char *format, va_list ap);
```

Le corrispondenti funzioni non
unificate

✓
Accettano una
V-List

Come determinare la lunghezza di stringhe?

- A volte non è possibile sapere la lunghezza di una stringa prima di aver generato la stringa stessa
- Esempio: salvare in una stringa in memoria la rappresentazione testuale del valore di un intero
 - Di quanti caratteri abbiamo bisogno?
- Per effettuare questo calcolo, possiamo utilizzare `snprintf()`:

Concerning the return value of `snprintf()`, C99 allows `str` to be `NULL`, and gives the return value (as always) as the number of characters that would have been written in case the output string has been large enough.

Come determinare la lunghezza di stringhe?

```
char *convert_to_ascii(int num)
{
    size_t len = snprintf(NULL, 0, "%d", num) + 1;
    char *ret = malloc(len);
    sprintf(ret, "%d", num);
    return ret;
}
```