



UNIVERSITÀ  
DI TRENTO

Dipartimento di Ingegneria e  
Scienza dell'Informazione  
DISI - Trento

# Programmazione 1

## 15 - Esercitazione

Giovanni De Toni

[giovanni.detoni@unitn.it](mailto:giovanni.detoni@unitn.it)

### Attenzione

La presente esercitazione verrà trasmessa via Zoom. Essa verrà anche registrata e successivamente messa a disposizione degli studenti dell'Università degli Studi di Trento. Per gli utenti connessi attraverso Zoom, in caso non desideriate per qualunque motivo essere registrati, siete pregati di effettuare la disconnessione ora. La lezione sarà comunque visionabile in modo asincrono.

Anno Accademico 2021/2022

# Nelle puntate precedenti

- **Stringhe**

```
char stringa [] = "Arancio";
```

- Contiene 5 elementi, i caratteri della stringa e terminatore `'\0'` che viene inserito automaticamente;
- Gli operatori di input e output operano direttamente su stringhe

```
char buffer[256];  
cin >> buffer;  
cout << buffer;
```

# Nelle puntate precedenti

- **Funzioni della libreria `<cstring>`**

- `strlen(s)`: ritorna la lunghezza della stringa `s`;
- `strcpy(t, s)`: copia la stringa `s` all'interno della stringa `t` e restituisce `t`;
- `strcmp(t, s)`: restituisce un valore negativo, nullo o positivo se `s` è alfabeticamente minore, uguale o maggiore di `t`;
- ...

# Nelle puntate precedenti

- **Argomenti da linea di comando**

```
int main( int argc, char * argv [] )
```

- `argc` indica il numero totale delle parole della riga di comando (argomenti + il nome dell'eseguibile)
- `argv` è un array multidimensionale che contiene gli argomenti sotto forma di stringhe;

# Nelle puntate precedenti

- **Argomenti da linea di comando**

```
./a.out input.txt output.txt 56
```

- `argc = 4`
- `argv: ["/a.out", "input.txt", "output.txt", "56"]`

Per convertire eventuali stringhe in numeri, bisogna utilizzare le funzioni `atoi()` o `atof()` della libreria `<cstdlib>`.

# Nelle puntate precedenti

- **Stream di I/O e File di Testo**

- Le funzioni per operare su file sono disponibili tramite la libreria `<fstream>`;
- L'utilizzo di uno stream è analogo a `cin` e `cout` (quindi utilizzando gli operatori `>>` e `<<`);
- Ricordarsi sempre di chiudere uno stream una volta aperto;

# Nelle puntate precedenti

- **Aprire/Chiudere uno stream**

```
1. fstream input, output;  
2. input.open("file.txt", ios::in);  
3. output.open("output.txt", ios::out);  
4.  
5. input.close();  
6. output.close();
```

# Nelle puntate precedenti

- **Leggere/Scrivere un file**

```
1. char buffer[256];  
2. input >> buffer;  
3. while(!input.eof())  
4. {  
5.     input >> buffer;  
6. }  
7.  
8. output << buffer;
```



# Nelle puntate precedenti

- **Funzioni della libreria `<iostream>`**

- `input.get(c)`: legge un carattere (spazi compresi) dallo stream e lo memorizza nella variabile `c`;
- `input.eof()`: indica se abbiamo raggiunto la fine dello stream;
- `input.fail()`: indica se ci sono stati errori nell'apertura dello stream;
- ...

# 1 - Calcolatrice

Scrivere un programma che, dati a linea di comando tre valori, due numeri e un operatore, ritorni a video il risultato dell'operazione matematica definita dall'operatore.

```
./calcolatrice 1 + 3
```

Il risultato è 4

## 2 - Impariamo a leggere

Scrivere un programma che, dato in input da linea di comando il nome di un file, ritorni a video il contenuto del suddetto file. Implementare anche dei controlli che avvisino l'utente in caso il numero di argomenti sia sbagliato (manca il file di input) oppure nel caso in cui il file di input non esista.

Documentazione <fstream>:

<http://www.cplusplus.com/reference/fstream/fstream/>

# 3 - Impariamo a copiare

Scrivere un programma che, dati in input da linea di comando i nomi di due file, A e B, copi il contenuto di A all'interno di B. Implementare anche dei controlli che avvisino l'utente in caso il numero di argomenti sia sbagliato (manca il file di input) oppure nel caso in cui il file di input non esista.

Documentazione <fstream>:

<http://www.cplusplus.com/reference/fstream/fstream/>

## 4 - Sostituzione di lettera

Scrivere un programma che, dato in input il nome di un file F e una lettera L, stampi a video il contenuto di quel file, sostituendo ogni occorrenza della lettera L con il simbolo “?”

```
"Che magnifica giornata", a  
=  
"Che m?gnific? giorn?t?"
```

## 5 - Intersezione di parole

Scrivere un programma che, dati in input due file A e B da riga di comando, generi un terzo file C che contenga tutte le parole presenti in entrambi i file A e B.

E' consentito l'uso delle funzioni `strcmp()` , `strcpy()` della libreria `<cstring>`. Assumete inoltre che i file contengano al massimo 1000 parole e che la lunghezza massima delle singole parole sia 100. **E' consentito aprire al massimo un unico stream per file.**