

# Esercizi su

---

# Funzioni

- Tracce extra
  - Sul sito del corso

- *funz\_max.cc*
- *funz\_fattoriale.cc*

# Documentazione

---

- Il codice va documentato (commentato)
  - Leggibilità
    - Riduzione degli errori
    - Manutenibilità
- Documentare gli algoritmi utilizzati

# Documentazione: funzioni

---

- Interfaccia della funzione
  - Valori in ingresso
  - Valori in uscita
- Algoritmo implementato

# Collaudo (testing)

---

- Se non avessimo aggiunto il controllo in *funz\_fattoriale.cc*
  - Cosa accadeva se si inseriva un numero negativo?
- In generale come si va alla ricerca degli errori?
- Innanzitutto occorre provare il programma

# Esautività 1/2

---

- Se un programma funziona correttamente per un valore di ingresso, si può affermare che sia corretto?

# Eshaustività 2/2

---

- Ovviamente no
- Senza entrare in ulteriori dettagli, per questo corso diciamo solo che bisogna cercare di provare il programma per tutti gli ingressi possibili, o almeno per un alta percentuale degli ingressi possibili
- Quale logica e quale approccio usare?



# Testing a scatola aperta

---

- Testing a scatola aperta (*white box*)
  - Mi metto nei panni del compilatore prima e soprattutto dell'esecutore dopo
    - Cerco di capire come vanno le cose al variare dei rami di codice eseguiti
  - I commenti nel programma aiutano

# Testing a scatola chiusa

---

- Testing scatola chiusa (*black box*)
  - Si opera sui valori di ingresso supponendo di non sapere nulla di come funziona il programma
    - Si provano i valori agli estremi, nel mezzo, fuori dagli estremi degli intervalli consentiti

# Somma di quadrati

---

- *somma\_quadrati.cc*
  - Prima di vedere la soluzione faremo qualche commento sul **tracing**

# Tracing 1/3

---

- Come faccio a capire dove e perché fallisce un programma?
- Cosa accade all'esecuzione di ciascuna istruzione?
  - Eventuale cambio del flusso di esecuzione in conseguenza di
    - **Lettura** di un qualche valore
    - **Scrittura** di un valore

- Cosa guida l'esecuzione di un programma?
  - Il **valore** delle variabili
- Come posso guardare il valore delle variabili mentre il programma è in esecuzione?
- Stampandolo (tracing)!

- Inserire una *cout*<< in un ciclo può creare problemi?
- Cosa succede se il ciclo non termina più?
- Possibili soluzioni?
  - Inserire delle letture da tastiera per controllare il ritmo delle iterazioni durante l'esecuzione

# Valori di ritorno ed eccezioni

---

- Ritornare
  - $-1$  oppure in generale
  - un valore fuori dall'intervallo di valori di output attesi
- in caso di errore è una buona norma?
- Soluzione migliore: meccanismo delle eccezioni del C++

- Abbiamo visto
  - Chiamata di funzione con due parametri
  - Suddivisione di un doppio ciclo tra il *main* ed una funzione
  - Utilizzo delle *cout*<< per il tracing



# Generazione numeri primi

---

- *gen\_primi.cc*
- Nella soluzione vedremo:
  - Invocazione di funzioni all'interno delle funzioni
  - Uso dell'istruzione vuota

# Compiti per casa

---

- Ordine di difficoltà:
  - *gen\_primi\_gemelli.cc*
  - *funz\_quadrato\_pieno.cc*
  - *verifica\_data.cc*
  - *funz\_pot\_pos\_overflow.txt*
  - *ricevimento\_iter.cc*