

**Insegnamento  
di  
PROGRAMMAZIONE E LABORATORIO DI PROGRAMMAZIONE  
a.a. 2018-19**

**Programma**

- 1. Un primo programma C**
  - 1.1. Caratteristiche generali di un programma C
  - 1.2. Modalità di compilazione
  - 1.3. Modalità di esecuzione
- 2. Cosa vuol dire programmare**
  - 2.1. Il concetto di informazione
  - 2.2. Il concetto di dato
  - 2.3. Il processo di codifica/decodifica
  - 2.4. Informazione implicita vs informazione esplicita
  - 2.5. Nozione intuitiva di algoritmo
- 3. I diagrammi di flusso**
  - 3.1. Caratteristiche generali
  - 3.2. La memoria
    - 3.2.1. Il modello
    - 3.2.2. Lo stato
  - 3.3. Tipologie di blocchi
    - 3.3.1. Blocchi di inizio e di termine
    - 3.3.2. Blocchi di acquisizione e di restituzione
    - 3.3.3. Blocchi di decisione
    - 3.3.4. Blocchi di elaborazione
  - 3.4. Operatori aritmetici e di confronto
  - 3.5. Vincoli di validità
  - 3.6. I Vettori
    - 3.6.1. Il tipo di dato
    - 3.6.2. Definizione
    - 3.6.3. Accesso agli elementi
  - 3.7. Le Matrici
    - 3.7.1. Il tipo di dato
    - 3.7.2. Definizione
    - 3.7.3. Accesso agli elementi
- 4. Il compilatore e le sue opzioni**
  - 4.1. La generazione dell'eseguibile
    - 4.1.1. Preprocessing

- 4.1.2. **Compilazione**
  - 4.1.3. **Linking**
- 4.2. **Le opzioni:**
  - 4.2.1. **Individuare la versione del compilatore**
  - 4.2.2. **Assegnare un nome al file eseguibile**
  - 4.2.3. **Invocare il Preprocessore**
  - 4.2.4. **Elevare il grado di trasportabilità**
- 4.3. **I messaggi di errore**
- 4.4. **I warning**
- 5. **Il modello di memoria del C**
  - 5.1. **Il modello**
  - 5.2. **Lo stato della memoria**
  - 5.3. **Modifiche allo stato della memoria**
    - 5.3.1. **Definizione di una variabile**
    - 5.3.2. **Il comando di assegnamento**
    - 5.3.3. **Rilascio della memoria**
  - 5.4. **Accesso alle variabili**
    - 5.4.1. **Tramite nome**
    - 5.4.2. **Tramite indirizzo: i puntatori e gli operatori & e \***
- 6. **Fondamenti di ingresso/uscita**
  - 6.1. **Flussi standard di ingresso/uscita e relativi default**
  - 6.2. **Le direttive per il pre-processore**
  - 6.3. **Ingresso/uscita da standard input/output**
    - 6.3.1. **La funzione *scanf()***
    - 6.3.2. **La funzione *printf()***
  - 6.4. **Gli specificatori di formato**
- 7. **Le funzioni**
  - 7.1. **Le funzioni della matematica vs le funzioni del C**
  - 7.2. **Definizione di una funzione**
  - 7.3. **Chiamata di una funzione**
  - 7.4. **Modalità di passaggio dei parametri**
  - 7.5. **Funzioni che modificano lo stato della memoria**
- 8. **L'aritmetica dei puntatori**
  - 8.1. **Gli operatori + e -**
- 9. **I costrutti di controllo**
  - 9.1. **Il blocco di istruzioni**
  - 9.2. **I costrutti *if* e *if-else***
  - 9.3. **Il costrutto *while***
  - 9.4. **Il costrutto *for***
  - 9.5. **Il costrutto *break***
  - 9.6. **Il costrutto *continue***

## 10. I vettori

10.1. Cosa sono

10.2. Definizione di un vettore e modifiche allo stato della memoria

10.3. L'operatore *sizeof()*

10.4. Modalità di accesso agli elementi di un vettore

10.4.1. Accesso tramite nome

10.4.2. Accesso tramite indirizzo

10.5. I vettori e le funzioni

10.5.1. I vettori come parametri formali

10.5.2. i vettori come parametri attuali

10.6. I/O di vettori:

10.6.1. acquisizione del contenuto di un vettore

10.6.2. restituzione del contenuto di un vettore

10.7. Limiti del dimensionamento statico di un vettore

## 11. Gestione della memoria

11.1. L'operatore di casting

11.2. Le direttive per il preprocessore

11.3. Allocare un buffer a run-time: la funzione *malloc()*

11.4. Rilasciare un buffer a run-time: la funzione *free()*

11.5. Allocare e inizializzare un buffer a run-time: la funzione *calloc()*

11.6. I memory leak: cosa sono e come evitarli

11.7. Effettuare la copia di un buffer: la funzione *memcpy()*

## 12. Allocazione dinamica di vettori

12.1. Limiti dell'allocazione statica dei vettori

12.2. Modalità di allocazione dinamica dei vettori

## 13. Le stringhe

13.1. Il tipo carattere

13.2. La tabella dei codici ASCII

13.3. Le stringhe

13.4. Definizione di una stringa

13.4.1. Statica

13.4.2. A run-time

13.5. Le stringhe costanti

13.6. Ingresso/uscita di stringhe

13.7. La libreria di gestione delle stringhe:

13.7.1. Direttive per il preprocessore

13.7.2. Determinare la lunghezza di una stringa: la funzione *strlen()*

13.7.3. Copiare stringhe: le funzioni *strcpy()* e *strncpy()*

13.7.4. Concatenare stringhe: le funzioni *strcat()* e *strncat()*

13.7.5. Confrontare stringhe: la funzione *strcmp()*

## **14. Le matrici**

**14.1. Cosa sono**

**14.2. Definizione a run-time di una matrice e modifiche allo stato della memoria**

**14.3. Modalità di accesso agli elementi di una matrice**

**14.3.1. Accesso tramite nome**

**14.3.2. Accesso tramite indirizzo**

**14.4. Modalità di rilascio della memoria allocata per una matrice**

**14.5. Le matrici e le funzioni**

**14.6. I/O di matrici:**

**14.6.1. acquisizione del contenuto di una matrice**

**14.6.2. restituzione del contenuto di una matrice**

## **15. I file**

**15.1. Apertura di un file**

**15.2. Chiusura di un file**

**15.3. Lettura del contenuto di un file**

**15.4. Scrittura all'interno di un file**