**Insegnamento**

**di**

**PROGRAMMAZIONE E LABORATORIO DI PROGRAMMAZIONE**

**a.a. 2018-19**

**Programma**

1. **Un primo programma C**
	1. **Caratteristiche generali di un programma C**
	2. **Modalità di compilazione**
	3. **Modalità di esecuzione**
2. **Cosa vuol dire programmare**
	1. **Il concetto di informazione**
	2. **Il concetto di dato**
	3. **Il processo di codifica/decodifica**
	4. **Informazione implicita vs informazione esplicita**
	5. **Nozione intuitiva di algoritmo**
3. **I diagrammi di flusso**
	1. **Caratteristiche generali**
	2. **La memoria**
		1. **Il modello**
		2. **Lo stato**
	3. **Tipologie di blocchi**
		1. **Blocchi di inizio e di termine**
		2. **Blocchi di acquisizione e di restituzione**
		3. **Blocchi di decisione**
		4. **Blocchi di elaborazione**
	4. **Operatori aritmetici e di confronto**
	5. **Vincoli di validità**
	6. **I Vettori**
		1. **Il tipo di dato**
		2. **Definizione**
		3. **Accesso agli elementi**
	7. **Le Matrici**
		1. **Il tipo di dato**
		2. **Definizione**
		3. **Accesso agli elementi**
4. **Il compilatore e le sue opzioni**
	1. **La generazione dell’eseguibile**
		1. **Preprocessing**
		2. **Compilazione**
		3. **Linking**
	2. **Le opzioni:**
		1. **Individuare la versione del compilatore**
		2. **Assegnare un nome al file eseguibile**
		3. **Invocare il Preprocessore**
		4. **Elevare il grado di trasportabilità**
	3. **I messaggi di errore**
	4. **I warning**
5. **Il modello di memoria del C**
	1. **Il modello**
	2. **Lo stato della memoria**
	3. **Modifiche allo stato della memoria**
		1. **Definizione di una variabile**
		2. **Il comando di assegnamento**
		3. **Rilascio della memoria**
	4. **Accesso alle variabili**
		1. **Tramite nome**
		2. **Tramite indirizzo: i puntatori e gli operatori & e \***
6. **Fondamenti di ingresso/uscita**
	1. **Flussi standard di ingresso/uscita e relativi default**
	2. **Le direttive per il pre-processore**
	3. **Ingresso/uscita da standard input/output**
		1. **La funzione *scanf()***
		2. **La funzione *printf()***
	4. **Gli specificatori di formato**
7. **Le funzioni**
	1. **Le funzioni della matematica vs le funzioni del C**
	2. **Definizione di una funzione**
	3. **Chiamata di una funzione**
	4. **Modalità di passaggio dei parametri**
	5. **Funzioni che modificano lo stato della memoria**
8. **L’aritmetica dei puntatori**
	1. **Gli operatori + e -**
9. **I costrutti di controllo**
	1. **Il blocco di istruzioni**
	2. **I costrutti *if* e *if-else***
	3. **Il costrutto *while***
	4. **Il costrutto *for***
	5. **Il costrutto *break***
	6. **Il costrutto *continue***
10. **I vettori**
	1. **Cosa sono**
	2. **Definizione di un vettore e modifiche allo stato della memoria**
	3. **L’operatore *sizeof()***
	4. **Modalità di accesso agli elementi di un vettore**
		1. **Accesso tramite nome**
		2. **Accesso tramite indirizzo**
	5. **I vettori e le funzioni**
		1. **I vettori come parametri formali**
		2. **i vettori come parametri attuali**
	6. **l’I/O di vettori:**
		1. **acquisizione del contenuto di un vettore**
		2. **restituzione del contenuto di un vettore**
	7. **Limiti del dimensionamento statico di un vettore**
11. **Gestione della memoria**
	1. **L’operatore di casting**
	2. **Le direttive per il preprocessore**
	3. **Allocare un buffer a run-time: la funzione *malloc()***
	4. **Rilasciare un buffer a run-time: la funzione *free()***
	5. **Allocare e inizializzare un buffer a run-time: la funzione *calloc()***
	6. **I memory leak: cosa sono e come evitarli**
	7. **Effettuare la copia di un buffer: la funzione *memcpy()***
12. **Allocazione dinamica di vettori**
	1. **Limiti dell’allocazione statica dei vettori**
	2. **Modalità di allocazione dinamica dei vettori**
13. **Le stringhe**
	1. **Il tipo carattere**
	2. **La tabella dei codici ASCII**
	3. **Le stringhe**
	4. **Definizione di una stringa**
		1. **Statica**
		2. **A run-time**
	5. **Le stringhe costanti**
	6. **Ingresso/uscita di stringhe**
	7. **La libreria di gestione delle stringhe:**
		1. **Direttive per il preprocessore**
		2. **Determinare la lunghezza di una stringa: la funzione *strlen()***
		3. **Copiare stringhe: le funzioni *strcpy()* e *strncpy()***
		4. **Concatenare stringhe: le funzioni *strcat()* e *strncat()***
		5. **Confrontare stringhe: la funzione *strcmp()***
14. **Le matrici**
	1. **Cosa sono**
	2. **Definizione a run-time di una matrice e modifiche allo stato della memoria**
	3. **Modalità di accesso agli elementi di una matrice**
		1. **Accesso tramite nome**
		2. **Accesso tramite indirizzo**
	4. **Modalità di rilascio della memoria allocata per una matrice**
	5. **Le matrici e le funzioni**
	6. **l’I/O di matrici:**
		1. **acquisizione del contenuto di una matrice**
		2. **restituzione del contenuto di una matrice**
15. **I file**
	1. **Apertura di un file**
	2. **Chiusura di un file**
	3. **Lettura del contenuto di un file**
	4. **Scrittura all’interno di un file**