

## Ripasso degli argomenti di terza

### RICHIESTE

Per gli argomenti indicati si richiede:

- Un documento in formato word contenente una intestazione (titolo, nome e cognome, classe, data); una esposizione dettagliata dell'argomento, comprendente definizioni, spiegazioni, esempi; almeno 4 domande a risposta aperta su quanto trattato; almeno 4 domande a risposta chiusa (quiz) con ognuna una risposta corretta e tre risposte errate
- Una presentazione in formato PowerPoint dell'argomento che aiuti la spiegazione alla classe;
- Il codice completo di un programma funzionante che illustri l'argomento in tutte le sue parti.
- Saper presentare l'argomento alla classe in una lezione di circa mezz'ora.

Tutto il materiale prodotto deve essere messo a disposizione della classe. I file devono iniziare con il proprio cognome seguito dal titolo.

### Elenco degli argomenti

1. Variabili: tipi di dati primitivi (int, float, double, char...) e loro dimensione; concetto di indirizzo di una variabile; tipi puntatori; costanti in C.
2. \* Visibilità delle variabili. Variabili locali e globali. Condivisione di una variabile tra più file dello stesso progetto.
3. Istruzioni di ingresso ed uscita (printf, scanf, gets, getc, getch, getchar). Vantaggi e svantaggi delle diverse soluzioni;
4. Istruzioni di ingresso e di uscita per i file: scrittura e lettura da file di testo (fprintf, fscanf, fgets...)
5. Istruzioni per il controllo di flusso (if, while, do... while, for, switch case); loro flowchart; if e cicli nidificati; istruzioni break e continue e loro utilizzo nei cicli
6. Istruzioni di assegnazione e di test: operatori aritmetici e logici (AND, OR, NOT). Funzioni matematiche della libreria math.h
7. Funzioni in C: definizione di funzione; sintassi e regole per il loro uso; funzioni void; funzioni con valore di ritorno; funzioni con e senza parametri di ingresso;
8. Funzioni in C: variabili locali e variabili globali; uso delle funzioni per lo sviluppo top down di problemi.
9. Funzioni in C: passaggio di parametri per valore e per riferimento.
10. \* Funzioni che restituiscono un puntatore.
11. Vettori: vettori di tipi primitivi, accesso agli elementi, esplorazione di un vettore. Passaggio di un vettore come parametro di una funzione.
12. Vettori: ordinamento di un vettore (almeno tre algoritmi diversi, da confrontare).
13. Vettori parzialmente riempiti: tipologie, passaggio alle funzioni; gestione della situazione di vettore pieno (overflow).
14. Vettori parzialmente riempiti: confronto degli algoritmi di inserimento di un elemento, ricerca di un elemento, eliminazione di un elemento;
15. Stringhe: definizione, utilizzo, funzioni predefinite per l'input e l'output (puts e gets) per il loro uso;
16. Stringhe: funzioni della libreria string.h
17. \* Vettori multidimensionali (matrici): utilizzo, passaggio di una matrice come parametro ad una funzione.
18. \* Vettori di puntatori: vettori di stringhe, vettori di struct. Schemi di come vengono memorizzati in memoria i vettori e i dati in essi contenuti. Aritmetica dei puntatori.
19. Strutture dati complesse: definizione di struttura, strutture di dati primitivi, strutture contenenti vettori, strutture di strutture. Istruzione typedef. Definizioni di variabili di tipo struct.

## Ripasso degli argomenti di terza

20. Struttura dati complesse: vettori di struct. Passaggio di una struct come parametro di una funzione.
21. File di testo: lettura di file di testo; nomi e posizione dei file; parsing di un file di testo;
22. File di testo: modificatori di accesso in modifica; aggiunta di dati in fondo ad un file; aggiunta di dati all'interno di quelli contenuti in un file
23. \* File binari: differenze con i file di testo; inserimento ed estrazione di dati dai file binari.
24. \* File binari: esempio della gestione di un magazzino (codice bene, descrizione, quantità) con un file binario.
25. \* Allocazione dinamica della memoria: istruzioni malloc, calloc, free, realloc. Esempio di allocazione: array di dimensione infinita.
26. \* Allocazione dinamica della memoria: allocazione di struct; funzioni che restituiscono puntatori a struct.
27. \* Realizzazione di un vettore di struct con l'allocazione dinamica
28. \* Funzioni ricorsive.
29. \* Direttive del preprocessore del C
30. \* Ambiente di sviluppo Visual Code di Microsoft: differenze rispetto a Dev-Cpp. Esempio di realizzazione di un progetto.

## Algoritmi da sapere senza incertezze

### Algoritmi numerici

1. Scomposizione in fattori primi di un numero intero
2. Fattoriale di un numero (eventualmente anche con versione ricorsiva)
3. Elevamento a potenza intera di un numero reale
4. MCD tra due numeri (esistono diverse soluzioni)
5. Calcolo del mcm tra due numeri (esistono diverse soluzioni)
6. Primalità di un numero intero (N.B. 1 non è considerato numero primo, si parte da due)
7. Stampa dei primi N numeri primi
8. Calcolo dell'n-esimo numero primo
9. Calcolo dei primi N termini della serie di Fibonacci (versione iterativa e/o versione ricorsiva)
10. Tabellina pitagorica
11. Calcolo della radice quadrata con il metodo dicotomico
12. Scambio del contenuto di due variabili

### Vettori parzialmente riempiti delle diverse tipologie (sentinella, contatore, valore nullo):

1. Copia di un vettore su un altro
2. Ricerca del max/min tra gli elementi
3. Somma degli elementi di un vettore (accumulazione); calcolo della media
4. Aggiunta di un elemento
5. Ricerca di un elemento (sia per vettore ordinato che non ordinato)
6. Calcolo del numero di occorrenze di un elemento
7. Ricerca del valore o dei valori più frequenti (moda)
8. Ordinamento di un vettore (diversi algoritmi)
9. Inserimento di un elemento in un vettore ordinato
10. Ricerca di un sottovettore in un vettore (esempio: ricerca sottostringa in una stringa)
11. Unione di due vettori ordinati
12. Unione di due vettori non ordinati (aggiunta in fondo)
13. Eliminazione di un elemento
14. Eliminazione dei duplicati
15. Intersezione tra due vettori di elementi distinti (insiemi)
16. Unione tra due vettori di elementi distinti (insiemi)