

# Prova Esame PSD 2023-2024

## ● PRIMA COMBINAZIONE

1. Progetto in C che implementa una struttura dati (workhome assignment - consegna piattaforma e-learning)
  - a. Gruppo/i da 3/2/1 persone/a
  - b. Codice Eseguitibile
  - c. Codice Casi di Test
  - d. Report dove viene spiegato:
    - i. La motivazione della scelta dell'ADT
    - ii. Progettazione
    - iii. Specifica sintattica e semantica
    - iv. Razionale casi di test

**Deadline 31 maggio (40% voto finale) - Deadline 15 giugno (30% voto finale)**

2. Prova Orale (3 domande 10-10-10=30)
  - a. Esercizio
  - b. Teorica
  - c. Teorica

**60% (31 maggio) - 70% (15 giugno) voto finale**

### **Consegna Progetto include:**

1. File sorgenti (Codici più Casi di Test - Il codice deve essere commentato)
2. File eseguibile con Make file
3. Report in **pdf**

## ● SECONDA COMBINAZIONE (SENZA PROGETTO)

1. Prova Orale (30)
  - a. Esercizio (5)
  - b. Domanda Teorica (5)
  - c. Domanda Teorica (5)
  - d. Esercizio con Pseudo Codice (15)

### **INFO UTILI (Work in Progress):**

1. Il voto del progetto è mantenuto fino a fine anno solare (31 Dicembre 2024).
2. Se il progetto non viene consegnato in nessuna delle due date, lo studente può svolgere solo la seconda combinazione.
3. Se lo studente fallisce il progetto può svolgere solo la seconda combinazione.
4. Il progetto ha il voto in trentesimi ed è considerato superato da 18 (incluso) a salire.
5. Non esistono mezzi voti, il  $\leq .5$  è arrotondato per difetto.
6. Una volta consegnato il progetto nelle date indicate (31 maggio/15 giugno), lo studente può svolgere l'orale non necessariamente al primo appello.

7. Dal momento che lo studente si prenota all'esame, i docenti sapranno già se lo studente ha sviluppato/superato/fallito/saltato il progetto, di conseguenza la prova orale sarà svolta secondo la combinazione associata.
8. La consegna del progetto, se in gruppo, andrà fatta da un solo membro del gruppo.

**Form da compilare (un solo membro per gruppo nel caso si scelga di fare il progetto):**

<https://forms.gle/HqNNjbE7Z3tx5j9v5>

# Tips per La Documentazione

## Layout e Design

### **Orientamento e Margine:**

- Per documenti testuali come rapporti o saggi, l'orientamento verticale è standard.
- Imposta i margini a un minimo di 2,5 cm su tutti i lati per garantire una buona leggibilità e spazio per eventuali note o commenti.

## Testo e Tipografia

### **Scelta del Font:**

- Preferisci font sans-serif come Arial, Helvetica, Serif o Times New Roman.
- Usa un corpo di testo di 11 o 12 punti per una facile leggibilità.

### **Intestazioni e Sottointestazioni:**

- Differenziale visivamente (dimensione, grassetto) e mantieni uno stile consistente per ciascun livello.
- Non saltare i livelli di intestazione (es. passare da un'intestazione di primo livello direttamente a una di terzo livello).

### **Enfasi:**

- Usa il grassetto o l'italico per enfatizzare, ma con parsimonia per evitare di sovraccaricare il lettore.

## Elementi di Lista

### **Bullet Points e Liste Numerate:**

- Utilizza bullet points per elenchi non ordinati e numeri per elenchi ordinati o istruzioni passo-passo.
- Mantieni i punti brevi; se necessario, fornisci ulteriori dettagli nel paragrafo seguente.

## Grafici, Tabelle e Immagini

### **Inserimento:**

- Inserisci elementi visivi vicino al testo di riferimento per mantenere la coerenza narrativa.
- Assicurati che ogni elemento grafico sia di alta qualità e non pixelato.

### **Didascalie e Riferimenti:**

- Ogni grafico o immagine deve avere una didascalia esplicativa e, se necessario, un riferimento alla fonte.

## Spaziatura e Allineamento

### Allineamento del Testo:

- L'allineamento giustificato può offrire un aspetto pulito.

### Interlinea:

- Usa un'interlinea di 1,15 o 1,5 per migliorare la leggibilità, soprattutto in documenti con molto testo.

## Revisione e Feedback

### Controllo Ortografico e Grammaticale:

- Utilizza strumenti di controllo integrati e, se possibile, esterni per catturare errori che potrebbero essere sfuggiti.

## Alcuni link utili

[Word applicare gli stili: titoli e sottotitoli e loro numerazione](#)

[Strutturazione della pagina con le guide - Supporto tecnico Microsoft](#)

[Inserire un sommario - Supporto tecnico Microsoft](#)

[Inserire una didascalia per un'immagine - Supporto tecnico Microsoft](#)

[Modelli di progettazione di Word | Microsoft Create](#)

# Tracce Progetto

## Traccia 1: Gestore di Eventi per Conferenze (Gruppo di 3 persone)

Obiettivo:

Implementare un sistema in C per gestire la programmazione degli eventi di una conferenza, inclusi i workshop, le sessioni di keynote e i panel di discussione. Il sistema deve consentire agli organizzatori di aggiungere, rimuovere e modificare eventi, nonché di visualizzare l'agenda della conferenza.

Requisiti:

- Gestione Eventi: Aggiungere, eliminare e modificare gli eventi della conferenza.
- Agenda Conferenza: Visualizzare un elenco ordinato degli eventi per data e ora.
- Strutture Dati (ADT - Abstract Data Type): Scegliere e implementare ADT appropriati, per mantenere gli eventi ordinati o gestire la coda di eventi in attesa di essere assegnati a una sala.
- Interfaccia Utente: Implementare una semplice interfaccia da linea di comando per interagire con il sistema.

Casi di Test:

- Aggiunta e rimozione di eventi, verificando che l'agenda si aggiorni correttamente.
- Modifica degli eventi esistenti e controllo degli effetti sul programma della conferenza.
- Test della visualizzazione dell'agenda per assicurarsi che gli eventi siano presentati in ordine cronologico.

Report:

- Motivazione della Scelta dell'ADT: Descrivere le ragioni per cui gli ADT selezionati sono ottimali per la gestione e l'organizzazione degli eventi della conferenza.
- Progettazione: Illustrare la struttura del sistema e come i diversi componenti interagiscono tra loro.
- Specifica Sintattica e Semantica: Elencare e descrivere le operazioni che possono essere eseguite sugli ADT scelti, inclusi pre e post condizioni, i loro input, output e effetti collaterali.
- Razionale dei Casi di Test

## **Traccia 2: Simulatore di Ristorante (Gruppo di 2/3 persone)**

Obiettivo:

Creare un programma in C che simuli il flusso di operazioni in un ristorante, gestendo ordini da tavolo, cucina e consegne. Il sistema dovrebbe permettere l'aggiunta di ordini, la loro elaborazione in cucina e la marcatura degli ordini consegnati.

Requisiti:

- Gestione Ordini: Aggiungere e visualizzare ordini in attesa, ordini in elaborazione in cucina e ordini consegnati.
- Cucina: Elaborare gli ordini in base all'ordine di arrivo e al tempo di preparazione.
- Consegne: Marcare gli ordini come consegnati una volta completati dalla cucina.
- Strutture Dati (ADT - Abstract Data Type): Utilizzare ADT per gli ordini in attesa e in cucina, e per tenere traccia degli ordini consegnati.
- Interfaccia Utente: Implementare una semplice interfaccia da linea di comando per interagire con il sistema.

Casi di Test:

- Aggiunta di ordini al sistema e verifica della loro corretta gestione nella coda di attesa.
- Elaborazione degli ordini in cucina e verifica del rispetto dell'ordine di arrivo e dei tempi di preparazione.
- Marcatura degli ordini come consegnati e verifica della loro rimozione dalla coda di cucina.

Report:

- Motivazione della Scelta dell'ADT: Illustrare perché specifici ADT sono stati scelti per rappresentare le diverse fasi degli ordini nel ristorante.
- Progettazione: Illustrare la struttura del sistema e come i diversi componenti interagiscono tra loro.
- Specifica Sintattica e Semantica: Elencare e descrivere le operazioni che possono essere eseguite sugli ADT scelti, inclusi pre e post condizioni, i loro input, output e effetti collaterali.
- Razionale dei Casi di Test

### **Traccia 3: Catalogo Musicale Personale (Singola persona)**

Obiettivo:

Creare un programma in C che consenta agli utenti di gestire un catalogo musicale personale. Il sistema deve permettere di aggiungere, rimuovere e cercare album musicali, tenendo traccia di dettagli come artista, titolo dell'album, anno di pubblicazione e genere musicale.

Requisiti:

- Gestione Album: Aggiungere nuovi album al catalogo, rimuovere album esistenti e modificare le informazioni degli album.
- Ricerca Album: Cercare album per artista, titolo, anno di pubblicazione o genere.
- Visualizzazione Catalogo: Mostrare un elenco degli album nel catalogo, opzionalmente ordinati per artista o anno di pubblicazione.
- Strutture Dati (ADT - Abstract Data Type): Utilizzare ADT per memorizzare gli album nel catalogo, consentendo una facile aggiunta, rimozione e ricerca.
- Interfaccia Utente: Implementare una semplice interfaccia da linea di comando per interagire con il sistema.

Casi di Test:

- Aggiunta di album al catalogo e verifica che siano correttamente memorizzati e recuperabili.
- Ricerca di album basata su diversi criteri (artista, titolo, anno, genere) e verifica dell'accuratezza dei risultati.
- Rimozione di album e verifica che siano effettivamente eliminati dal catalogo.

Report:

- Motivazione della Scelta dell'ADT: Illustrare perché specifici ADT sono stati scelti per il catalogo musicale, considerando operazioni come l'aggiunta, la rimozione e la ricerca di album.
- Progettazione: Illustrare la struttura del sistema e come i diversi componenti interagiscono tra loro.
- Specifica Sintattica e Semantica: Elencare e descrivere le operazioni che possono essere eseguite sugli ADT scelti, inclusi pre e post condizioni, i loro input, output e effetti collaterali.
- Razionale dei Casi di Test

# Griglia di Valutazione del Progetto

## 1. Qualità del Codice (Punti Totali: 6)

- **Leggibilità e Organizzazione (3 punti):** Il codice è ben strutturato e organizzato? Sono utilizzati nomi leggibili per variabili e funzioni?
- **Commenti e Documentazione Interna (3 punti):** Il codice è ben commentato, rendendo chiare le funzioni e le logiche implementate?

## 2. Adempimento dei Requisiti (Punti Totali: 6)

- **Completezza (3 punti):** Tutti i requisiti funzionali sono soddisfatti?
- **Correttezza e Test (3 punti):** Il software funziona correttamente come atteso in tutti i casi di test forniti? Sono inclusi casi di test che coprono i vari scenari?

## 3. Implementazione e Utilizzo degli ADT (Punti Totali: 8)

- **Scelta degli ADT (2 punti):** La scelta degli ADT è ben motivata e adeguata alle necessità del progetto?
- **Implementazione degli ADT (2 punti):** Gli ADT sono implementati in modo corretto ed efficiente?
- **Integrazione nel Sistema (2 punti):** Gli ADT sono integrati in modo efficace nel sistema complessivo per gestire e organizzare il tema della traccia?
- **Memoria (2 punti):** Il codice gestisce correttamente l'utilizzo e la de/allocazione della memoria?

## 4. Specifica Sintattica e Semantica (Punti Totali: 5)

- **Chiarezza e Completezza (2.5 punti):** Le specifiche sintattiche e semantiche degli ADT e delle funzioni del sistema sono chiare, complete e correttamente documentate?
- **Corrispondenza con l'Implementazione (2.5 punti):** Le specifiche sintattiche e semantiche corrispondono fedelmente all'implementazione nel codice?

## 5. Documentazione e Report (Punti Totali: 5)

- **Chiarezza e Organizzazione del Report (2 punti):** Il report è ben organizzato, chiaro e facile da seguire?
- **Dettaglio Tecnico e Motivazione (2 punti):** Il report fornisce una spiegazione dettagliata delle scelte tecniche, inclusa la motivazione per la scelta e l'uso degli ADT?
- **Completezza (1 punto):** Il report include tutte le sezioni richieste, coprendo in modo adeguato ogni aspetto del progetto?