



---

# DOMANDE MUSIMATICA

---

Domande di tutti i capitoli della dispensa del Prof. De Prisco



10 OTTOBRE 2024  
UNISA – CORSO DI MUSIMATICA

## MUSIMATICA DOMANDE

Capitolo 1 – Nozioni Basilari.....	2
Capitolo 2 – Temperamento Equilabile .....	3
Capitolo 3 - MIDI .....	11
Capitolo 4 – Rappresentazione e Analisi della Musica.....	13
Capitolo 5 – Similarità e plagio musicale .....	14
Capitolo 6 – Composizione Automatica Semplice .....	16

## Capitolo 1 – Nozioni Basilari

1. Quali sono le principali caratteristiche di un suono? Cosa determina ognuna di esse?

1. Le principali caratteristiche sono l'altezza determinata dalla frequenza, il volume determinato dall'intensità e la forma delle onde che determina il timbro.

2. Si scriva una funzione con dominio  $Z$  e codominio  $R$  che fornisca tutte le ottave di una data frequenza di riferimento  $fr$ .

2.  $f_n = fr \cdot 2^n$ ,  $n$  appartiene a  $Z$

3. Si scriva una funzione con dominio  $\{1, 2, \dots, 12\}$  e codominio  $R$  che fornisca i 12 semitoni di un'ottava a partire da una frequenza di riferimento  $fr$  nel sistema ben temperato.

3.  $f_k = fr \left(2^{\frac{1}{12}}\right)^k$ ,  $k \in 0, 1, 2 \dots 12$

4. Siano  $n_1, n_2, \dots, n_i$  gli intervalli, misurati in semitoni, di una scala musicale che copre un'ottava. Quanto vale la loro somma  $\sum_{j=1}^i n_j$ ? Perché?

4. 12, perché una scala ci deve far salire da una nota alla sua ottava che dista esattamente 12 semitoni.

5. Che cosa è la scala cromatica?

5. La scala cromatica è una scala che è formata da tutti i suoni dell'ottava in sequenza di frequenza crescente. Corrisponde alla sequenza di intervalli 1,1,1,1,1,1,1,1,1,1,1,1. In una scala cromatica due note successive sono sempre a distanza di un semitono.

6. Siano  $f_1$  e  $f_2$  le frequenze di due suoni. Supponendo che  $f_1$  e  $f_2$  differiscono di pochi Hz, quando è più facile distinguere le due note? Quando le due frequenze sono basse o quando le due frequenze sono alte? Perché?

6. È più facile distinguere 2 note quando le loro frequenze sono basse, siccome l'orecchio umano percepisce meglio le frequenze basse rispetto alle alte.

7. In una rappresentazione della musica su un sistema di assi cartesiani cosa viene rappresentato sugli assi? E a cosa corrisponde una nota in una tale rappresentazione?

7. Sulle ordinate il tempo e sulle ascisse la frequenza. Ogni nota corrisponde ad un segmento orizzontale, la cui inizio e fine, la cui durata dipende dalla posizione della nota della partitura e dalla durata.

8. A che velocità (approssimativa) si propaga il suono nell'aria? Se cambia la temperatura cosa succede a tale velocità?

8. 343 m/s con 20 gradi, se diminuisce la temperatura il suono rallenta.

9. Quale è l'estensione in frequenze di un pianoforte? E quali sono i limiti (approssimativi) che l'orecchio umano può percepire?

9. Il pianoforte ha un'estensione che va dai 27 ai 4000Hz. Mentre l'orecchio umano percepisce dai 17 ai 17.000 Hz.

10. Nel circolo delle quinte due scale successive in quante alterazioni differiscono? Perché?

10. 2 scale differiscono di un'alterazione perché è costruito aggiungendo un diesis o un bemolle alla scala per mantenere la corretta sequenza di toni e semitoni della scala.

11. Volendo rappresentare una musica su assi cartesiani in cui l'asse (orizzontale) delle ascisse rappresenta il tempo, a quale asse assoceresti la melodia e a quale asse l'armonia?

11. **Melodia** → **Asse verticale (ordinate)**, poiché la melodia implica cambiamenti di altezza nel tempo.

**Armonia** → **Asse orizzontale (ascisse)**, poiché rappresenta la sovrapposizione di più note simultanee in un certo istante temporale.

12. Fornisci una caratterizzazione degli accordi maggiori, minori, diminuiti e aumentati in termini di intervalli di semitoni.

12. Un accordo è un insieme simultaneo di almeno 3 note. Esiste:

- **Accordo maggiore**: formato da terza maggiore e una quinta giusta. Se consideriamo gli intervalli successivi abbiamo una terza maggiore e una terza minore con distanze di 4 e 3 semitoni.
- **Accordo minore**: formato da terza minore e quinta giusta. Se consideriamo gli intervalli successivi abbiamo una terza minore e una terza maggiore con distanze in semitoni di 3 e 4.

13. Da quante voci è formato un corale? E come si chiamano?

13. 4 Voci: Soprano, Contralto, Tenore e Basso.

## Capitolo 2 – Temperamento Equilabile

1. Se un corpo vibra a una frequenza  $f$ , Quali frequenze hanno i suoni armonici prodotti dalla

vibrazione?

1.  $2f$ ,  $3f$ ,  $4f$ ,  $5f$ ....

2. Come si sommano due intervalli di frequenze contigui?

2. Per sommare due intervalli contigui è necessario *moltiplicare* le loro ampiezze

3. Come si costruisce la scala pitagorica a partire da una frequenza  $f$ ?

3. **Parti dalla nota di riferimento  $f$ .**

**Calcola la quinta giusta** moltiplicando per  $\frac{3}{2}$

**Riporta le note nell'ottava corretta:** se la frequenza supera il doppio di  $f$ , dividila per 2; se è inferiore, moltiplicala per 2.

**Ripeti il processo** per altre quinte giuste fino a ottenere le 7 note della scala.

**Aggiusta le note all'interno di un'ottava** per ottenere una scala completa.

4. Un semitono pitagorico è la metà di un tono pitagorico?

4. No, anche se c'è poca differenza, non è la metà esatta.

5. Come si costruisce la scala naturale a partire da una frequenza  $f$ ?

**Parti dalla nota di riferimento  $f$ .**

**Applica gli intervalli** della scala maggiore: **Tono-Tono-Semitono-Tono-Tono-Tono-Semitono (TTsTTTs).**

- Tono = moltiplica la frequenza per  $\frac{9}{8}$ .
- Semitono = moltiplica la frequenza per  $\frac{16}{15}$ .

**Ripeti il processo** per tutte le note della scala, rispettando la sequenza di toni e semitoni.

**Ottieni le 7 note** della scala diatonica maggiore entro

6. Quali sono gli intervalli di tono e semitono nella scala naturale?

TTsTTTs

7. Il semitono della scala naturale,  $\frac{16}{15}$ , è la metà del tono naturale grande,  $\frac{9}{8}$ , o del tono naturale piccolo,  $\frac{10}{9}$  ?

7. Non è la metà esatta di nessuno dei 2.

8. Come si costruisce la scala delle quinte a partire da una frequenza  $f$ ?

8. **Parti dalla frequenza  $f$ .**

**Moltiplica per  $\frac{3}{2}$**  per ottenere la quinta giusta.

**Riporta la nota nell'ottava** (se necessario, dividi per 2).

**Ripeti il processo** fino a ottenere tutte le 12 note.

9. Quale è il problema delle scale pitagoriche, naturali e delle quinte che viene risolto con il temperamento equabile?

9. Il **problema principale** delle scale **pitagoriche, naturali e delle quinte** è che gli intervalli tra le note non sono **ugualmente distribuiti**. Di conseguenza, queste scale non permettono di trasporre la musica liberamente fra le varie tonalità e non permettono di **chiudere perfettamente il ciclo**: ad esempio, nella scala pitagorica, 12 quinte giuste non corrispondono esattamente a 7 ottave, creando un **errore di intonazione** chiamato **comma pitagorico**.

Il **temperamento equabile** risolve questo problema dividendo l'**ottava in 12 semitoni uguali**, cioè ogni semitono ha lo stesso rapporto di frequenza ( $\sqrt[12]{2}$ )

10. Quanto vale un semitono nel temperamento equabile? Si può esprimere come frazione?

$\sqrt[12]{2}$ , no perché gli irrazionali non possono essere espressi come frazioni.

11. Costruisci un sistema musicale equabile, basato sull'assioma dell'ottava, ma con 10 note (al posto di 12).

11. Decido una  $f$  di partenza, da questa frequenza per ottenere  $f_2$  moltiplicherò  $f \times \sqrt[10]{2}$ , per ottenere  $f_3$  farò  $f \times (\sqrt[10]{2})^2$  e così via fino a  $f_9$ , con formula generale:  $f_n = f \times (\sqrt[10]{2})^{n-1}$ .

12. Costruisci un sistema musicale equabile, cambiando l'assioma di partenza: non più il rapporto  $\frac{2}{1}$ , ma  $\frac{3}{1}$ . Usa 20 note.

$f_n = f \times (\sqrt[20]{3})^{n-1}$  fino a  $n = 20$

13. Scrivere un programma che partendo dalla scala pitagorica costruisce tutte le altre scale partendo dalle frequenze generate dalle note della scala pitagorica e da tutte quelle che vengono generate di conseguenza. Quando si ferma il programma?

13.

import math

```

def genera_scala_pitagorica(freq_iniziale, num_note):
    frequenze = [freq_iniziale] # Lista delle frequenze
    rapporto_quinta = 3 / 2

    # Genera la scala pitagorica
    for i in range(1, num_note):
        nuova_freq = frequenze[-1] * rapporto_quinta
        # Riporta la frequenza nell'ottava corretta (se > 2 * freq_iniziale, dividila per 2)
        while nuova_freq > 2 * freq_iniziale:
            nuova_freq /= 2
        frequenze.append(nuova_freq)

    return frequenze

def costruisci_scale(frequenze):
    # Implementa qui la logica per costruire altre scale (maggiore, minore, ecc.)
    # usando le frequenze generate dalla scala pitagorica
    scale = {}
    scale['maggiore'] = sorted(frequenze[:7]) # Seleziona 7 note per una scala maggiore (esempio)
    # Altri tipi di scale possono essere generate qui
    return scale

def main():
    freq_iniziale = 440 # La centrale
    num_note = 12 # Numero di note da generare (12 per la scala pitagorica)

    # Genera la scala pitagorica
    frequenze_pitagoriche = genera_scala_pitagorica(freq_iniziale, num_note)

    # Costruisci altre scale
    scale_generate = costruisci_scale(frequenze_pitagoriche)

    # Stampa le scale generate

```

```
for nome_scala, frequenze in scale_generate.items():
    print(f"Scala {nome_scala}: {frequenze}")
```

```
if __name__ == "__main__":
    main()
```

Si ferma quando ha generato tutte e 12 le frequenze.

14. Ripetere l'esercizio 13 partendo dalla scala naturale.

14. import numpy as np

```
def generate_natural_scale(f0=261.63, num_notes=8):
    """Genera la scala naturale (Do maggiore) a partire da una frequenza fondamentale f0."""
    # Intervalli della scala naturale (Do, Re, Mi, Fa, Sol, La, Si, Do)
    intervals = [2, 2, 1, 2, 2, 2, 1] # Tono, Tono, Semitono, Tono, Tono, Tono, Semitono
    scale = [f0] # Inizio con la frequenza fondamentale
    for interval in intervals:
        next_freq = scale[-1] * (2 ** (interval / 12)) # Calcola la frequenza successiva
        scale.append(next_freq)

    return scale
```

```
def generate_scales(starting_scale, max_freq=2000):
    """Genera tutte le scale a partire da una scala iniziale."""
    scales = [starting_scale]
    new_scales = []

    for scale in scales:
        for note in scale:
            # Costruisci una nuova scala usando il rapporto di 3/2 (quinta giusta)
            new_note = note * (3/2)
            if new_note < max_freq:
                # Se la nuova nota è sotto il limite, crea una nuova scala
                new_scale = scale + [new_note]
```



```
new_scales.append(new_scale)
```

```
# Aggiungi le nuove scale generate
```

```
scales.extend(new_scales)
```

```
# Stampa le scale generate
```

```
for idx, scale in enumerate(scales):
```

```
    print(f"Scala {idx + 1}: {[round(note, 2) for note in scale]}")
```

```
return scales
```

```
# Genera la scala naturale
```

```
natural_scale = generate_natural_scale()
```

```
# Genera tutte le scale a partire dalla scala naturale
```

```
all_scales = generate_scales(natural_scale)
```

15. Scrivere un programma che genera le note usando l'intervallo di quinta, cioè  $\frac{3}{2}$ . Quando

si ferma il programma?

15. def generate\_notes(start\_freq=440, max\_freq=2000, max\_notes=12):

```
    """Genera note usando l'intervallo di quinta giusta (3/2)."""
```

```
    notes = [start_freq] # Inizia con la frequenza fondamentale
```

```
    current_freq = start_freq
```

```
    while current_freq <= max_freq and len(notes) < max_notes:
```

```
        # Calcola la frequenza della nota successiva usando l'intervallo di quinta
```

```
        current_freq = current_freq * (3 / 2)
```

```
    if current_freq <= max_freq:
```

```
        notes.append(current_freq)
```

```
    return notes
```

```
# Frequenza di partenza (La centrale)
starting_frequency = 440 # Hz
all_notes = generate_notes(starting_frequency)
```

```
# Stampa le note generate
for idx, note in enumerate(all_notes):
    print(f"Nota {idx + 1}: {round(note, 2)} Hz")
```

Il programma si ferma quando:

- La frequenza supera i **2000 Hz**.
- Il numero di note generate raggiunge il **massimo di 12**.

16. Ripetere l'esercizio 15 usando l'intervallo di quarta, cioè  $\frac{3}{2}$ . Quando si ferma il programma?

```
16. def generate_notes_with_fourth(f0=261.63, max_freq=2000):
    """Genera note a partire da una frequenza fondamentale f0 usando l'intervallo di quarta (3/4)."""
    notes = [f0] # Inizio con la frequenza fondamentale
    new_note = f0 * (3 / 4) # Calcola la prima quarta
    while new_note < max_freq:
        notes.append(new_note)
        new_note *= (3 / 4) # Calcola la quarta successiva
```

```
# Stampa le note generate
for idx, note in enumerate(notes):
    print(f"Nota {idx + 1}: {round(note, 2)} Hz")
```

```
return notes
```

```
# Genera le note usando l'intervallo di quarta
generated_notes = generate_notes_with_fourth()
```

17. La scala cromatica delle quinte costruisce le 12 note sfruttando esclusivamente l'intervallo

di una quinta ( $\frac{3}{2}$ ). È possibile fare un procedimento simile con l'intervallo di quarta?

Motivare e discutere la risposta.

17. è teoricamente possibile costruire una scala cromatica delle quarte utilizzando l'intervallo di quarta, il risultato non è equivalente a quello ottenuto con le quinte in termini di chiusura armonica e intonazione. Pertanto, la scala delle quarte potrebbe non essere utilizzata in modo pratico nella musica come lo è la scala delle quinte, a causa di queste limitazioni armoniche e intonative.

## Capitolo 3 - MIDI

1. Come vengono rappresentate le note con il protocollo MIDI? E i diversi timbri? Quante note e quanti timbri diversi possono essere codificati?

1. Le note vengono rappresentate con degli interi che vanno da 0 a 127. Il numero di timbri utilizzabili è sempre da 0 a 127

2. Quali sono i messaggi MIDI che permettono di iniziare a suonare una nota e di terminare il suono?

Note On che ha canale di appartenenza e codice della nota e Note Off che ha canale di appartenenza, codice della nota e velocità di rilascio.

3. Si consideri il seguente valore binario in notazione a lunghezza variabile: 10000010 (primo byte) 00000010 (secondo byte). Quale valore (decimale) codifica?

3. 130, nel primo si omette l'1 e quindi vale 2 ma si combina col secondo valendo  $128 + 2$ .

4. Con quali byte inizia un file MIDI? E con quali byte inizia una traccia all'interno di un file MIDI?

4. Il file MIDI Inizia con la Stringa "MThd" + 4 Byte che contengono la lunghezza del blocco che per quello di intestazione è sempre 6. La traccia inizia con la Stringa "MTrk" + 4 Byte che contengono la lunghezza del blocco

5. Che cosa è il delta time in un file MIDI? In che unità viene specificato?

5. Specifica dopo quanto tempo, rispetto all'evento precedente, l'evento attuale deve accadere, viene specificato in Ticks.

6. Che cosa è lo status byte nel protocollo MIDI? Si fornisca una spiegazione dettagliata.

6. Permette di distinguere i tipi di eventi fra: MIDI, sistema e meta eventi.

7. Quanti parametri prevedono i vari messaggi MIDI. Si faccia qualche esempio.

7. Al più 2, ad esempio Note Off ha 2 parametri: codice della nota MIDI e velocità di rilascio.

Program Change invece ne ha solo 1: il parametro che specifica lo strumento MIDI.

8. Si spieghi come funziona il metodo detto *running status*.

8. Il running status permette di non spedire il byte di stato nel caso in cui il valore di questo byte sia lo stesso di quello dell'evento precedente. Questa situazione viene segnalata da uno status byte

minore o uguale a 127 se il valore è così allora lo status byte è fittizio e in realtà il suo valore verrà usato come valore del primo parametro dell'evento.

9. In cosa consiste il “trucco” del Note Off per i file MIDI?

9. Il trucchetto consiste nel sostituire i messaggi di Note Off con messaggi di Note On la cui velocità di attacco (secondo parametro) è 0

10. Si faccia qualche esempio di meta-evento MIDI fornendo qualche dettaglio.

10. I meta-eventi servono a inserire informazioni aggiuntive non prettamente collegate alla musica codificata dagli eventi MIDI, come ad esempio il nome della traccia.

La configurazione è: 255 + tipo + lunghezza + dati evento.

Un esempio può essere **Key Signature** Questo meta evento specifica la tonalità. Il formato è 255 89 2 b1 b2 con b1 che specifica il numero di alterazioni e b2 che specifica il modo.

## Capitolo 4 – Rappresentazione e Analisi della Musica

1. In un file MusicXML quale è la differenza fra lo schema partwise e quello timewise?

1. Partwise dà priorità all'organizzazione gerarchica vista orizzontalmente, cioè le battute musicali sono contenute nella parte, mentre timewise dà priorità all'organizzazione gerarchica vista verticalmente, cioè le parti, i singoli righi musicali, sono contenuti nella "battuta".

2. Che cosa è music21?

2. è una libreria Python che fornisce strumenti per la manipolazione della musica in formati simbolici e permette di gestire i file in formato MusicXML.

3. Descrivere brevemente la rappresentazione tinyNotation di music21 e fornire un esempio musicale in tale rappresentazione.

**Nome della nota:** Lettere da "A" a "G"

**Ottava:** Numero che indica l'ottava.

"C4" indica il Do centrale del pianoforte e "C5" è il Do dell'ottava successiva.

## Capitolo 5 – Similarità e plagio musicale

1. Quale è la edit distance fra ababca e cbaca? Indicare anche la sequenza di operazioni che permettono di trasformare una stringa nell'altra usando un numero di operazioni pari alla edit distance.

1. L'edit distance tra "ababca" e "cbaca" è **2**, e la sequenza di operazioni minima è:

- **Sostituzione** di "a" con "c". → cbabca
- **Cancellazione** di "b". → cbaca

2. La edit distance può essere usata come metrica di similitudine fra due stringhe. Rappresentando la musica con delle stringhe si può sfruttare la edit distance per fornire una metrica di similitudine fra due melodie. Tuttavia, per rappresentare la musica è comodo usare più di un carattere per ogni nota. Come si può adattare la edit distance per gestire questo aspetto?

2. Se cambiano poche note nella melodia anche la stringa della rappresentazione cambierà di poco e conseguentemente l'edit distance sarà piccola. Per confrontare 2 melodie è necessario avere entrambe le melodie nella stessa tonalità.

3. Definire una rappresentazione testuale che permetta di descrivere una melodia specificando le note; in particolare la rappresentazione deve considerare l'altezza delle note e la durata permettendo di rappresentare le durate di semibreve, minime, semiminime, ... fino alle semibiscrome.

3. PITCH-ND siccome va a scegliere un minimo comune divisore e ci permette di rappresentare tutte le note con tutte le durate.

4. Si consideri la seguente rappresentazione testuale per descrivere delle melodie: una stringa di metacaratteri in cui ogni metacarattere ha una lettera per il nome della nota e un numero per l'ottava. Quali sono i vantaggi e gli svantaggi di una tale rappresentazione?

4. Questa rappresentazione è **semplice ed efficiente** per rappresentare una sequenza di note in un contesto di altezza e ottava, rendendola utile per confronti rapidi o analisi base di melodie.

Tuttavia, la sua **incompletezza** in termini di altre caratteristiche musicali come durata, intensità e articolazione la rende **inadatta** a rappresentazioni musicali più dettagliate e precise, necessarie per una descrizione completa di una melodia.

5. Definire una rappresentazione testuale che evita il problema della trasposizione.

5. Per aggirare questo problema usiamo la rappresentazione PITCH che dà importanza agli intervalli fra note successive. Al posto di rappresentare la singola nota rappresenteremo gli intervalli fra due note successive misurandolo in semitoni.



## Capitolo 6 – Composizione Automatica Semplice

1. Descrivere il metodo di composizione automatico ideato da Guido d'Arezzo.

1. prima associa un simbolo ad ognuna delle note in un intervallo di 2 ottave. Quindi, utilizza una ulteriore associazione di questi simboli con le vocali *a, e, i, o, u*. Avendo solo 5 vocali, Guido crea un'associazione ciclica per la quale alla stessa vocale corrispondono più note. Partendo da un testo si estraggono le vocali e da queste si crea una sequenza di note. Per ogni vocale in realtà c'è da scegliere una fra 3 (o 4) note e la scelta viene fatta in modo casuale.

2. Descrivere il gioco dei dadi per la creazione automatica di musica.

2. è un sistema di composizione basato sulla generazione casuale di un numero fra 2 e 12 ripetuta per un prefissato numero di volte. Ogni numero casuale determina la selezione di una battuta musicale fra 11 disponibili. Il tutto si basa sulla creazione a priori delle battute musicali disponibili per la selezione.

3. Si consideri il gioco dei dadi per una composizione lunga 10 battute e con il lancio di un solo dado. Quante varianti occorre pre-comporre per ogni battuta? Quante sono le possibili diverse composizioni che possono essere composte lanciando il dado?

3.  $6^{10}$  con 6 varianti precomposte per ogni battuta.

4. Nel gioco dei dadi per la composizione automatica quale è l'elemento chiave (dovuto alla bravura di chi pre-compone le battute) per il quale la musica composta risulta gradevole?

4. Ogni singola possibilità deve legarsi bene ad ogni possibile battuta precedente e ogni possibile battuta successiva.

5. Descrivere l'automa cellulare denominato *Game of life*.

5. l'universo delle celle è bidimensionale, il vicinato di una cella è costituito dalle 8 celle che condividono o un angolo o un lato con la cella stessa, vi sono 2 stati, cellula viva o morta con le seguenti regole di evoluzione.

- Una cella morta che al tempo  $t$  ha esattamente 3 vicini vivi diventa viva (nascita).
- Una cella viva che al tempo  $t$  ha 2 o 3 vicini vivi rimane viva (sopravvivenza).
- In tutti gli altri casi la cella muore o rimane morta (sovraffollamento o solitudine)

6. Descrivere l'automa cellulare denominato *Cyclic Cellular Automaton*.

6. universo è bidimensionale ma gli stati sono  $p$ , dove  $p$  è un parametro, e sono numerati da 0 a  $p - 1$ . La regola è la seguente: se per una cella  $(i, j)$  che si trova nello stato  $k$  ci sono almeno  $t$ (soglia) celle adiacenti nello stato  $k + 1 \bmod p$  allora la cella passa anch'essa nello stato  $k + 1 \bmod p$ .

7. Si faccia un esempio di un procedimento di composizione automatica che può essere considerato *frattale*.

Prendiamo 2 note Do4 e Fa4, ognuna delle due note genera una coppia di note con lo stesso intervallo.

Il Do4 genererebbe la stessa coppia di note originali, mentre il Fa4 genererebbe la coppia Fa4 e Sib, il procedimento potrebbe continuare all'infinito, magari definendo dei limiti superiori e inferiori entro i quali devono rimanere le note.

