

Esercizio 1

Scrivi un programma Python che gestisca un elenco di studenti di una scuola. Il programma deve consentire agli utenti di aggiungere nuovi studenti, visualizzare l'elenco degli studenti, cercare uno studente per nome e rimuovere uno studente dall'elenco.

Soluzione

```
# Inizializzazione dell'elenco degli studenti
elenco_studenti = []

# Menu principale
while True:
    print("\nGestione dell'elenco degli studenti")
    print("1. Aggiungi uno studente")
    print("2. Visualizza l'elenco degli studenti")
    print("3. Cerca uno studente per nome")
    print("4. Rimuovi uno studente")
    print("0. Esci")

    scelta = input("Cosa vuoi fare? ")

    if scelta == "1":
        nome = input("Inserisci il nome dello studente: ")
        elenco_studenti.append(nome)
        print(f"{nome} è stato aggiunto all'elenco degli studenti.")
    elif scelta == "2":
        if elenco_studenti:
            print("Elenco degli studenti:")
            for studente in elenco_studenti:
                print(studente)
        else:
            print("Non ci sono studenti nell'elenco.")
    elif scelta == "3":
        nome_ricerca = input("Inserisci il nome dello studente da cercare: ")
        trovato = False
```

```

        for studente in elenco_studenti:
            if nome_ricerca.lower() in studente.lower():
                print(f"{studente} è presente nell'elenco degli
studenti.")
                trovato = True
            if not trovato:
                print("Lo studente cercato non è presente nell'elenco.")
        elif scelta == "4":
            nome_rimozione = input("Inserisci il nome dello studente da
rimuovere: ")
            if nome_rimozione in elenco_studenti:
                elenco_studenti.remove(nome_rimozione)
                print(f"{nome_rimozione} è stato rimosso dall'elenco degli
studenti.")
            else:
                print("Lo studente da rimuovere non è presente nell'elenco.")
        elif scelta == "0":
            print("Grazie per aver utilizzato il programma di gestione degli
studenti. Arrivederci!")
            break
        else:
            print("Scelta non valida. Riprova.")

```

Esercizio 2

Dai in input una parola e controlla se la parola è palindroma

```

def is_palindromo(parola):
    # Verifica se la parola è uguale alla sua inversione
    if parola == parola[::-1]:
        return True
    else:
        return False

```

Esercizio 3

Un numero perfetto è un numero naturale uguale alla somma dei suoi divisori positivi, escluso sé stesso. Scrivi una funzione che verifichi se un numero è perfetto oppure no.

Soluzione

```
def perfetto(n):
    somma_divisori = 0

    # Itera sui numeri da 1 a n/2 (escluso n)
    for i in range(1, n//2 + 1):
        # Se i è un divisore di n, si aggiunge alla somma
        if n % i == 0:
            somma_divisori += i

    # Se la somma dei divisori è uguale a n, allora n è un numero
    perfetto

    if somma_divisori == n:
        return True
    else:
        return False

n = int(input("Inserisci un numero intero positivo: "))
if perfetto(n):
    print(f"Il numero {n} è un numero perfetto.")
else:
    print(f"Il numero {n} non è un numero perfetto.")
```

Esercizi interessanti da guardare

Scrivi un programma Python che simuli un gioco di labirinto testuale. Il labirinto è rappresentato da una griglia rettangolare di dimensioni specificate dall'utente. Il giocatore è rappresentato da un carattere '@' e può muoversi all'interno del labirinto utilizzando i tasti freccia (su, giù, sinistra, destra). Il labirinto contiene anche un punto di arrivo rappresentato dal carattere 'X'. Lo scopo del giocatore è raggiungere il punto di arrivo navigando attraverso il labirinto senza colpire le pareti, rappresentate dal carattere '#'. Se il giocatore tenta di muoversi oltre i confini del labirinto o contro una parete, il movimento non è consentito e il giocatore rimane nella sua posizione corrente.

Soluzione

```
# Definizione del labirinto

labirinto = [
    "#####",
    "#      #",
    "#   ## #",
    "# #   #",
    "# #X  #",
    "#@   #",
    "#####"]

# Posizione iniziale del giocatore
posizione_giocatore = (5, 1)

# Funzione per stampare il labirinto
def stampa_labirinto(labirinto, posizione_giocatore):
    for i in range(len(labirinto)):
        row = ""
        for j in range(len(labirinto[i])):
            if (i, j) == posizione_giocatore:
                row += "@"
            else:
                row += labirinto[i][j]
        print(row)

# Funzione per muovere il giocatore
```

```

def muovi_giocatore(direzione, posizione_giocatore):
    riga, colonna = posizione_giocatore
    if direzione == "su":
        nuova_riga = riga - 1
        nuova_colonna = colonna
    elif direzione == "giù":
        nuova_riga = riga + 1
        nuova_colonna = colonna
    elif direzione == "sinistra":
        nuova_riga = riga
        nuova_colonna = colonna - 1
    elif direzione == "destra":
        nuova_riga = riga
        nuova_colonna = colonna + 1
    else:
        print("Direzione non valida!")
        return posizione_giocatore

    if labirinto[nuova_riga][nuova_colonna] != "#":
        return nuova_riga, nuova_colonna
    else:
        return posizione_giocatore

# Loop di gioco
while True:
    stampa_labirinto(labirinto, posizione_giocatore)
    if labirinto[posizione_giocatore[0]][posizione_giocatore[1]] == "X":
        print("Hai raggiunto il punto di arrivo! Congratulazioni!")
        break

    direzione = input("Dove vuoi muoverti? (su/giù/sinistra/destra):
").lower()

    posizione_giocatore = muovi_giocatore(direzione, posizione_giocatore)

```