# esercizio 1  
nome\_scuola = "Epicode"  
  
# Inizializza l'indice  
indice = 0  
  
# Esegui il ciclo while fino a quando l'indice è inferiore alla lunghezza della stringa  
while indice < len(nome\_scuola):  
 # Stampa il carattere corrente  
 print(nome\_scuola[indice])  
  
 # Incrementa l'indice per passare al prossimo carattere  
 indice += 1  
  
# esercizio 2  
  
# Inizializza la variabile  
numero = 0  
  
# Esegui il ciclo while fino a quando il numero è minore o uguale a 20  
while numero <= 20:  
 # Stampa il numero corrente  
 print(numero)  
  
 # Incrementa il numero per passare al prossimo  
 numero += 1  
# esercizio 3  
  
# Inizializza la variabile esponente  
esponente = 0  
  
# Inizializza la variabile risultato  
risultato = 1  
  
# Esegui il ciclo while per calcolare e stampare le prime 10 potenze di 2  
while esponente < 10:  
 # Stampa il risultato corrente  
 print(f"2^{esponente} = {risultato}")  
  
 # Incrementa l'esponente  
 esponente += 1  
  
 # Calcola la prossima potenza di 2  
 risultato \*= 2  
  
# esercizio 4  
# Richiedi all'utente di inserire N  
N = int(input("Inserisci il numero N per calcolare le prime N potenze di 2: "))  
  
# Inizializza la variabile esponente  
esponente = 0  
  
# Inizializza la variabile risultato  
risultato = 1  
  
# Esegui il ciclo while per calcolare e stampare le prime N potenze di 2  
while esponente < N:  
 # Stampa il risultato corrente  
 print(f"2^{esponente} = {risultato}")  
  
 # Incrementa l'esponente  
 esponente += 1  
  
 # Calcola la prossima potenza di 2  
 risultato \*= 2  
  
# esercizio 5  
# Inizializza la variabile esponente  
esponente = 0  
  
# Inizializza la variabile risultato  
risultato = 1  
  
# Esegui il ciclo while per calcolare e stampare le potenze di 2 minori di 25000  
while risultato < 25000:  
 # Stampa il risultato corrente  
 print(f"2^{esponente} = {risultato}")  
  
 # Incrementa l'esponente  
 esponente += 1  
  
 # Calcola la prossima potenza di 2  
 risultato \*= 2  
  
# esercizio 6  
  
studenti = ["Alex", "Bob", "Cindy", "Dan", "Emma", "Faith", "Grace", "Henry"]  
corsi = ["Cybersecurity", "Data Analyst", "Backend", "Frontend", "Data Analyst", "Backend"]  
  
# Aggiungi i dati mancanti uno alla volta  
if "Emma" not in studenti:  
 studenti.append("Emma")  
 corsi.append("Data Analyst")  
  
if "Faith" not in studenti:  
 studenti.append("Faith")  
 corsi.append("Backend")  
  
if "Grace" not in studenti:  
 studenti.append("Grace")  
 corsi.append("Frontend")  
  
if "Henry" not in studenti:  
 studenti.append("Henry")  
 corsi.append("Cybersecurity")  
  
# Verifica che le liste siano della stessa lunghezza  
if len(studenti) == len(corsi):  
 # Stampa la lista corsi  
 print(corsi)  
else:  
 print("Le liste non sono della stessa lunghezza.")  
# esercizio 7  
  
# Chiedi all'utente di inserire una stringa  
input\_stringa = input("Inserisci una stringa: ")  
  
# Verifica la lunghezza della stringa  
lunghezza\_stringa = len(input\_stringa)  
  
# Gestisci i casi particolari (lunghezza inferiore a 6 caratteri)  
if lunghezza\_stringa < 6:  
 print("La stringa deve contenere almeno 6 caratteri.")  
else:  
 # Visualizza i primi 3 caratteri, seguiti da 3 punti di sospensione, e infine gli ultimi 3 caratteri  
 risultato = input\_stringa[:3] + "..." + input\_stringa[-3:]  
 print("Risultato:", risultato)  
  
# esercizio 8  
# Richiedi all'utente di inserire un numero  
numero = int(input("Inserisci un numero: "))  
  
# Inizializza una lista per memorizzare i divisori  
divisori = []  
  
# Trova i divisori del numero  
for i in range(2, numero + 1):  
 while numero % i == 0:  
 # Aggiungi il divisore alla lista  
 divisori.append(i)  
 # Riduci il numero dividendo per il divisore  
 numero //= i  
  
# Stampa la lista dei divisori  
print("I divisori di", numero, "sono:", divisori)  
  
# esercizio 9  
  
# Richiedi all'utente di inserire un numero K  
K = int(input("Inserisci un numero K: "))  
  
# Richiedi all'utente di inserire un numero N  
N = int(input("Inserisci un numero N: "))  
  
# Inizializza una lista per memorizzare le potenze di K  
potenze\_di\_K = []  
  
# Calcola e memorizza le prime N potenze di K  
for esponente in range(N):  
 potenza = K \*\* esponente  
 potenze\_di\_K.append(potenza)  
  
# Stampa la lista risultante  
print(f"Le prime {N} potenze di {K} sono:", potenze\_di\_K)  
  
# esercizio 10  
# Lista dei guadagni degli ultimi 12 mesi  
guadagni = [100, 90, 70, 40, 50, 80, 90, 120, 80, 20, 50, 50]  
  
# Inizializza variabili per la somma e la lunghezza della lista  
somma\_guadagni = 0  
lunghezza\_lista = len(guadagni)  
  
# Inizializza l'indice per il ciclo while  
indice = 0  
  
# Calcola la somma dei guadagni utilizzando il ciclo while  
while indice < lunghezza\_lista:  
 somma\_guadagni += guadagni[indice]  
 indice += 1  
  
# Calcola e stampa la media dei guadagni  
if lunghezza\_lista > 0:  
 media\_guadagni = somma\_guadagni / lunghezza\_lista  
 print(f"La media dei guadagni degli ultimi 12 mesi è: {media\_guadagni}")  
else:  
 print("La lista dei guadagni è vuota.")  
  
# esercizio 11  
  
# Lista di codici fiscali  
lista\_cf = {"ABCDEF95G01A123B", "GHIJKL91M02A321C", "MNOPQR89S03A456D", "STUVWX95Z04A654E", "XYZABC01D05A789F",  
 "DEFGHI95J06A987G"}  
  
# Inizializza una lista per i codici fiscali contenenti "95"  
cf\_con\_95 = []  
  
# Trova i codici fiscali che contengono "95" e aggiungili alla lista cf\_con\_95  
for cf in lista\_cf:  
 if "95" in cf:  
 cf\_con\_95.append(cf)  
  
# Stampa la lista dei codici fiscali che contengono "95"  
print("Codici fiscali contenenti '95':", cf\_con\_95)  
  
# Per ogni codice fiscale con "95", stampa i caratteri relativi al nome e al cognome  
for cf in cf\_con\_95:  
 nome = cf[0:6]  
 cognome = cf[6:15]  
 print(f"Codice fiscale: {cf}, Nome: {nome}, Cognome: {cognome}")  
  
# esercizio 12  
  
# Liste degli studenti, dei corsi e delle edizioni  
studenti = ["Alex", "Bob", "Cindy", "Dan", "Emma", "Faith", "Grace", "Henry"]  
corsi = ["Cybersecurity", "Data Analyst", "Backend", "Frontend", "Data Analyst", "Backend", "Frontend", "Cybersecurity"]  
edizioni = [1, 2, 3, 2, 2, 1, 3, 3]  
  
# Stampare gli studenti che frequentano la prima edizione  
print("Studenti che frequentano la prima edizione:")  
  
for i in range(len(studenti)):  
 if edizioni[i] == 1:  
 print(studenti[i])  
# esercizio 13  
  
# Lista di prezzi con il simbolo dell'euro  
prezzi = ["100 €", "200 €", "500 €", "10 €", "50 €", "70 €"]  
  
# Inizializza una nuova lista per i prezzi con il simbolo del dollaro  
prezzi\_in\_dollari = []  
  
# Sostituisci il simbolo dell'euro con quello del dollaro per ogni stringa nella lista  
for prezzo in prezzi:  
 prezzo\_in\_dollari = prezzo.replace("€", "$")  
 prezzi\_in\_dollari.append(prezzo\_in\_dollari)  
  
# Stampa la lista risultante  
print("Prezzi con il simbolo del dollaro:")  
print(prezzi\_in\_dollari)  
  
# esercizio 14  
  
# Lista degli studenti  
studenti = ["Alex", "Bob", "Cindy", "Dan", "Emma", "Faith", "Grace", "Henry", "Isabelle", "John"]  
  
# Inizializza due liste per le squadre  
squadra\_pari = []  
squadra\_dispari = []  
  
# Assegna gli studenti alle squadre in base alla loro posizione  
for i in range(len(studenti)):  
 if i % 2 == 0:  
 squadra\_pari.append(studenti[i])  
 else:  
 squadra\_dispari.append(studenti[i])  
  
# Visualizza le squadre  
print("Squadra Pari:", squadra\_pari)  
print("Squadra Dispari:", squadra\_dispari)