**Interview Questions  
python**

31. Write a Python function to generate a random password of a given length.

Solution- import random

import string

def generate\_random\_password(length=12):

characters = string.ascii\_letters + string.digits + string.punctuation

password = ''.join(random.choice(characters) for \_ in range(length))

return password

# Example usage:

generated\_password = generate\_random\_password(16) # Generate a 16-character password

print(f"Generated Password: {generated\_password}")

32. Implement a program to convert a decimal number to binary and vice versa.

Solution- def decimal\_to\_binary(decimal):

return bin(decimal)[2:]

def binary\_to\_decimal(binary):

return int(binary, 2)

def main():

while True:

print("Choose an option:")

print("1. Decimal to Binary")

print("2. Binary to Decimal")

print("3. Exit")

choice = input("Enter your choice (1/2/3): ")

if choice == '1':

decimal = int(input("Enter a decimal number: "))

print(f"Binary equivalent: {decimal\_to\_binary(decimal)}")

elif choice == '2':

binary = input("Enter a binary number: ")

print(f"Decimal equivalent: {binary\_to\_decimal(binary)}")

elif choice == '3':

print("Exiting the program.")

break

else:

print("Invalid choice. Please enter 1, 2, or 3.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

33. Create a Python function to find the sum of all even numbers in a given list.

Solution- def even(e):

for i in e:

if i%2==0:

print(i)

a=[2,3,4,5,6,7,8]

even(a)

34. Write a program to find the ASCII value of a character.

Solution- # Function to find the ASCII value of a character

def find\_ascii\_value(char):

return ord(char)

# Input character

char = input("Enter a character: ")

# Ensure the input is a single character

if len(char) == 1:

ascii\_value = find\_ascii\_value(char)

print(f"The ASCII value of '{char}' is {ascii\_value}")

else:

print("Please enter a single character.")

35. Implement a function to check if a given string is a valid email address or not.

Solution- import re

def is\_valid\_email(email):

# Define a regex pattern for a valid email address

email\_pattern = re.compile(

r"(^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$)"

)

# Use the pattern to match the given email string

return re.match(email\_pattern, email) is not None

# Given email address to check

email = "santanumohanty444@gmail.com"

# Check if the email is valid

is\_valid = is\_valid\_email(email)

print(f"{email}: {is\_valid}")