**Python – Conditional Statements Assignment Questions**

**# 1.Write a Python program to check if a number is positive, negative, or zero.**

**def check\_number\_positive\_negative\_zero(number: int) -> str:**

**if number > 0:**

**return f"Number is Positive"**

**elif number < 0:**

**return f"Number is negative"**

**else:**

**return f"Number is zero "**

**number = check\_number\_positive\_negative\_zero(0)**

**print(number)**

**# 2.Given a student’s score, write a program to print “Pass” if score ≥ 40, otherwise print “Fail”.**

**def print\_student\_result(score: int) -> str:**

**# if score >= 40:**

**#     return "Pass"**

**# else:**

**#     return "Fail"**

**return "Pass" if score >= 40 else "Fail"**

**result = print\_student\_result(12)**

**print(result)**

**# 3.Write a program that checks if a given year is a leap year.**

**def check\_leap\_year(year: int) -> str:**

**if (year % 400 == 0) or (year % 4 == 0 and year % 100 != 0):**

**return f"{year} is a Leap Year"**

**return f"{year} is not a Leap Year"**

**year = check\_leap\_year(1900)  # check cases for 1900, 2000, 2024**

**print(year)**

**# 4.Given 3 integers, write a program to find the largest number using if-elif-else.**

**def find\_largest\_number(number1: int, number2: int, number3: int) -> str:**

**if number1 > number2 and number1 > number3:**

**return f"{number1} is > both {(number2)} and {(number3)}"**

**elif number2 > number1 and number2 > number3:**

**return f"{number2} is > both {(number1)} and {(number3)}"**

**else:**

**return f"{number3} is > both {(number1)} and {(number2)}"**

**largest\_number = find\_largest\_number(67, 56, 1)**

**print(f"Largest Number: {largest\_number}")**

**# 5.Evaluate if a candidate passes technical round using:**

**#       coding\_skill ≥ 4**

**#       problem\_solving ≥ 4**

**#       cs\_fundamentals ≥ 4**

**#       (Use and operator)**

**def check\_candidate\_skills(coding: int, problem: int, cs: int) -> bool:**

**return coding >= 4 and problem >= 4 and cs >= 4**

**Candidate = check\_candidate\_skills(5, 6, 7)**

**# 6.Check if a candidate meets communication and CGPA criteria:**

**#       CGPA ≥ 7.0**

**#       communication ≥ 3**

**def check\_candidate\_criteria(cgpa: float, communication: int) -> bool:**

**return cgpa >= 7 and communication >= 3**

**candidate = check\_candidate\_criteria(8.9, 2)**

**print(candidate)**

**# 7.Based on inputs, decide hiring decision using nested if:**

**#       Check technical first, then CGPA & communication**

**def hiring\_decision(**

**coding\_skill: int,**

**problem\_solving: int,**

**cs\_fundamentals: int,**

**cgpa: float,**

**communication: float,**

**) -> str:**

**if check\_candidate\_skills(coding\_skill, problem\_solving, cs\_fundamentals):**

**if check\_candidate\_criteria(cgpa, communication):**

**return "Candidate is hired!"**

**else:**

**return "Rejected: Did not meet CGPA/Communication criteria."**

**else:**

**return "Rejected: Did not pass the technical round."**

**hire\_or\_not = hiring\_decision(9, 4, 7, 8.9, 1)**

**print(hire\_or\_not)**

**# 8.Write a program to check if a number is divisible by both 3 and 5 using and.**

**def number\_is\_divisible\_by\_3\_5(number: int) -> str:**

**if (number % 3 == 0) and (number % 5 == 0):**

**return f"{number} is divisible by both 3 and 5"**

**return f"{number} is not divisible by both 3 and 5"**

**print(number\_is\_divisible\_by\_3\_5(15))**

**# 9.Given a list of candidates with CGPAs, print “Fast-Track” if CGPA > 9 or communication ≥ 4**

**def get\_candidates\_track(cgpa: float, communication: int) -> str:**

**if (cgpa >= 9) or (communication >= 4):**

**return "Fast-Track"**

**print(get\_candidates\_track(2, 5))**

**# 10.Write a program using not to check if a person is not eligible for a scholarship (i.e., CGPA < 6).**

**def person\_is\_eligible\_for\_scholarship(cgpa: int) -> str:**

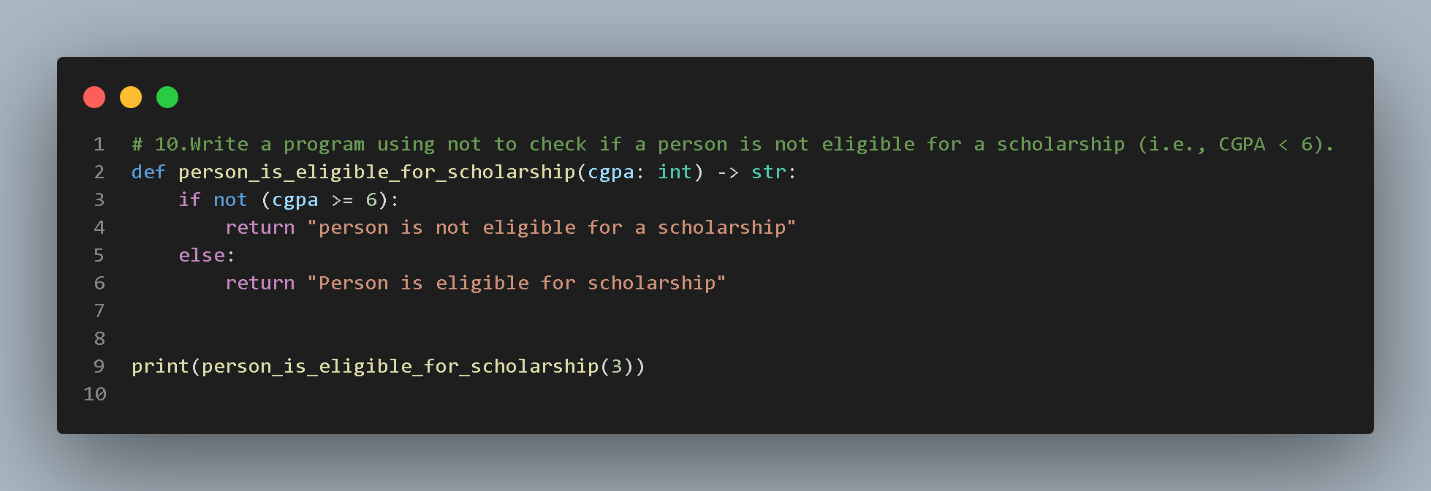
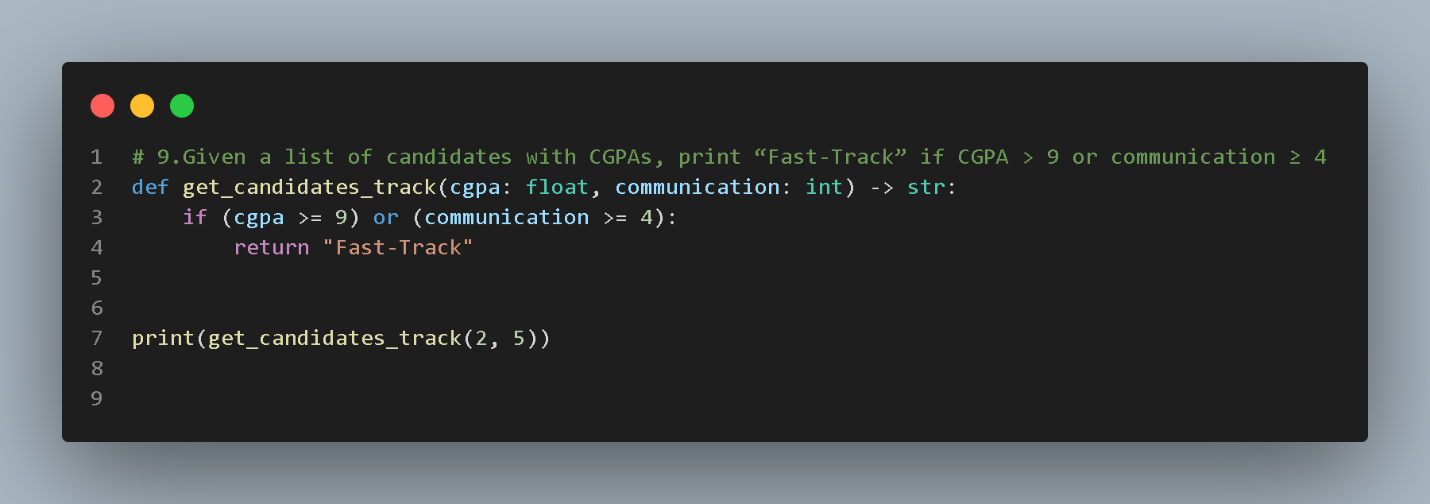
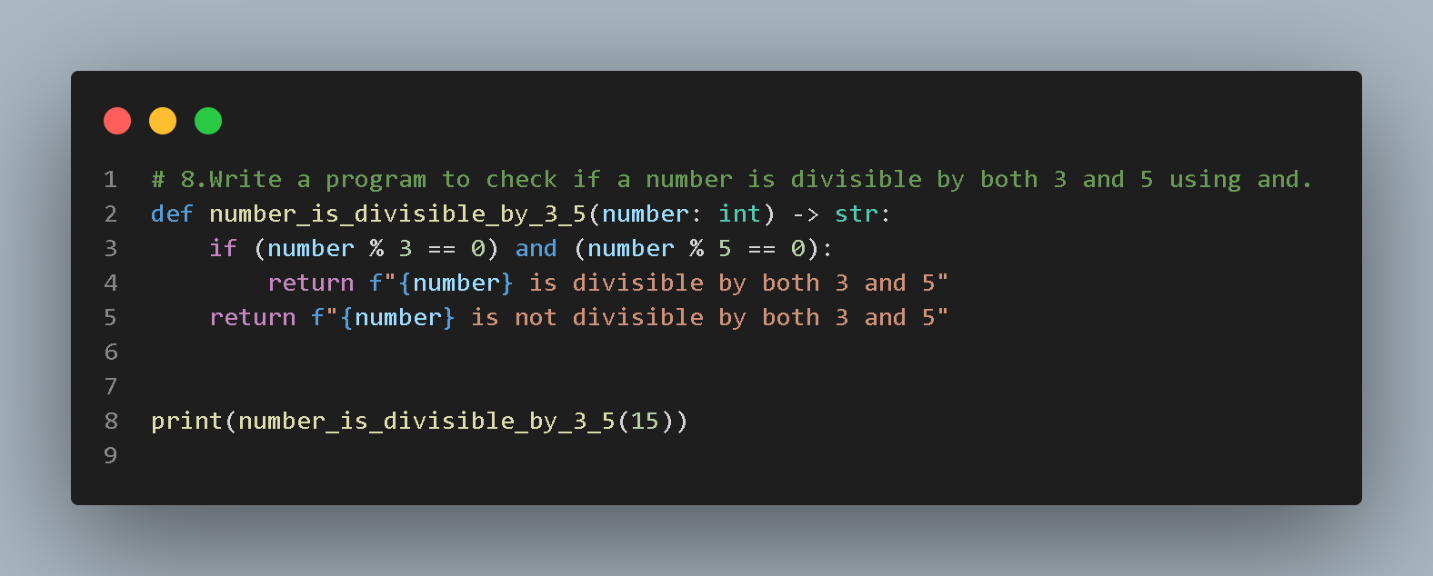
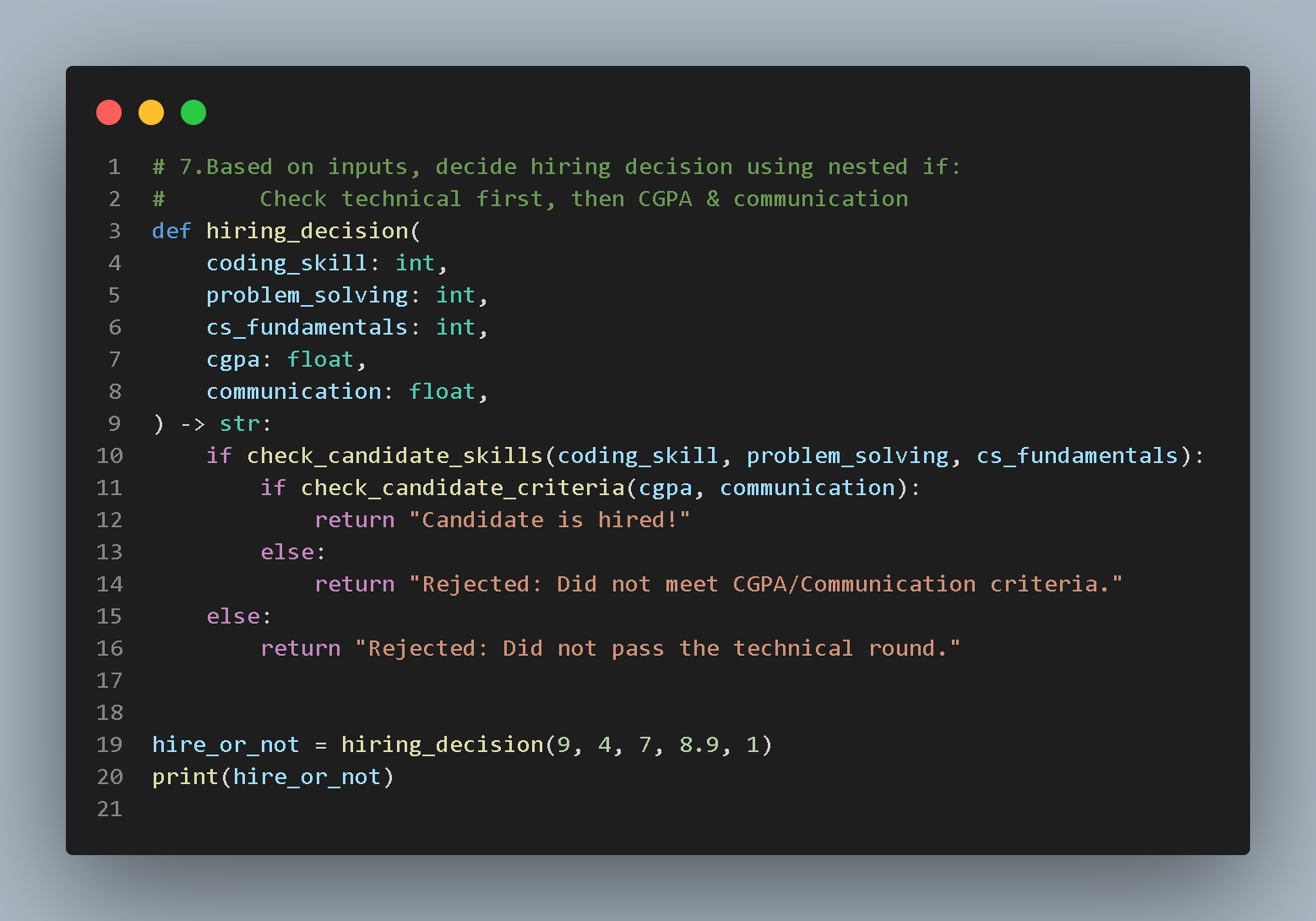
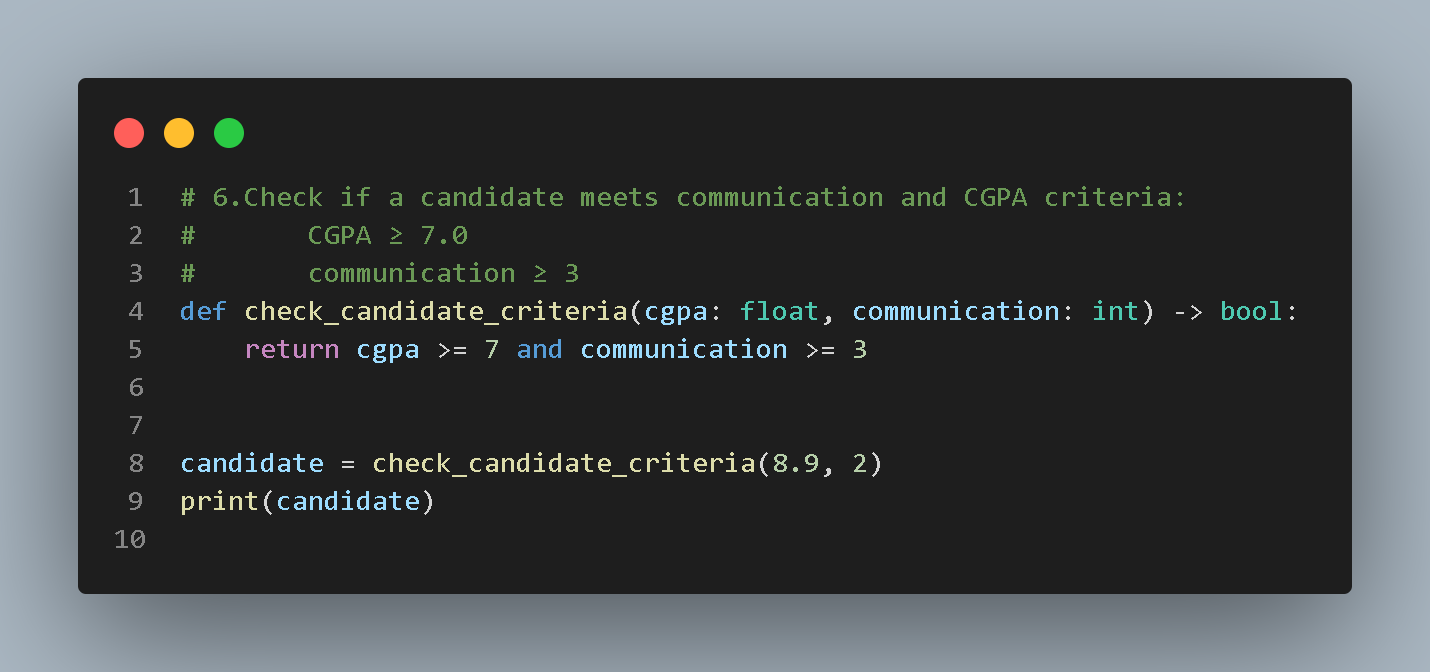
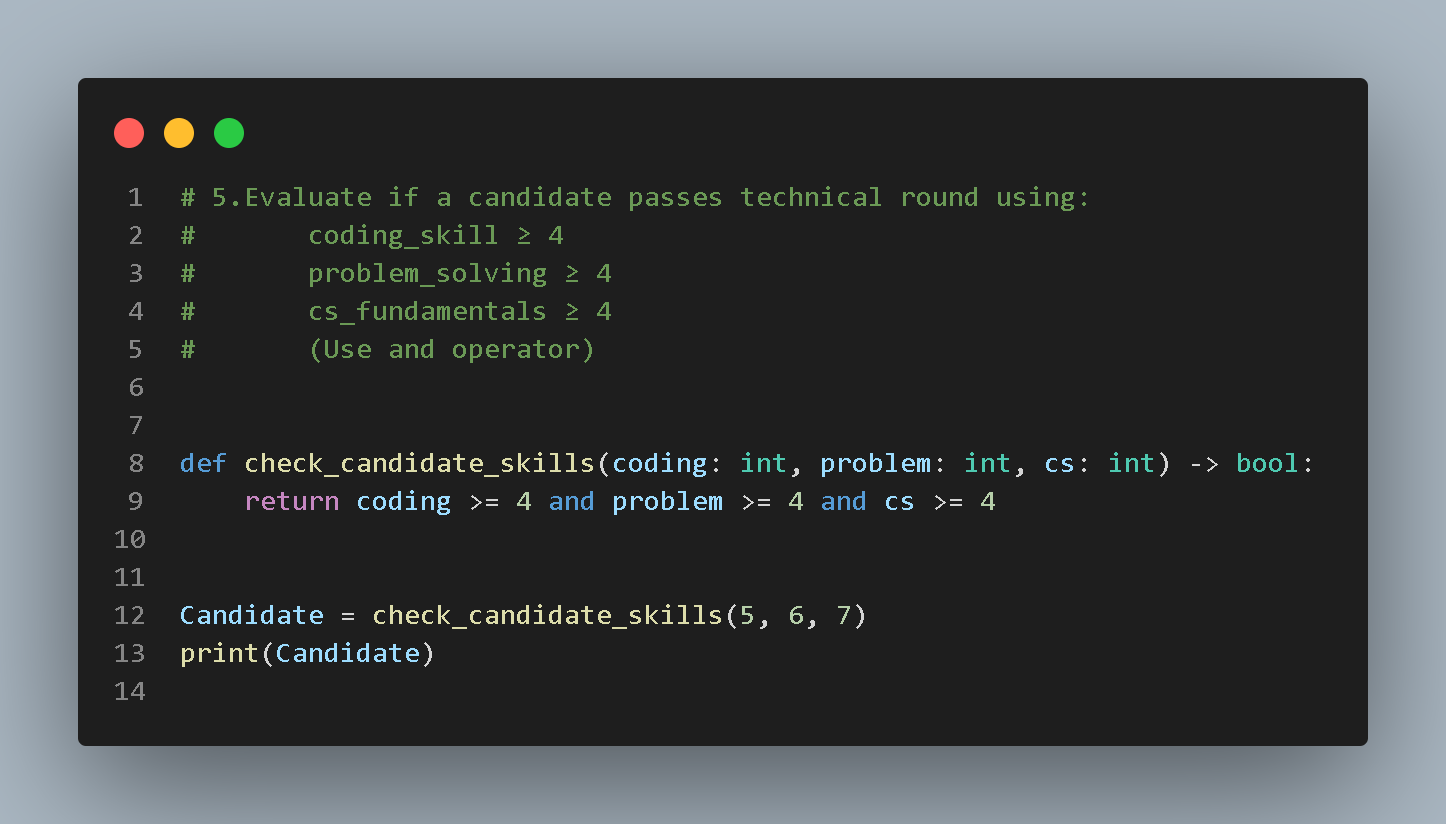
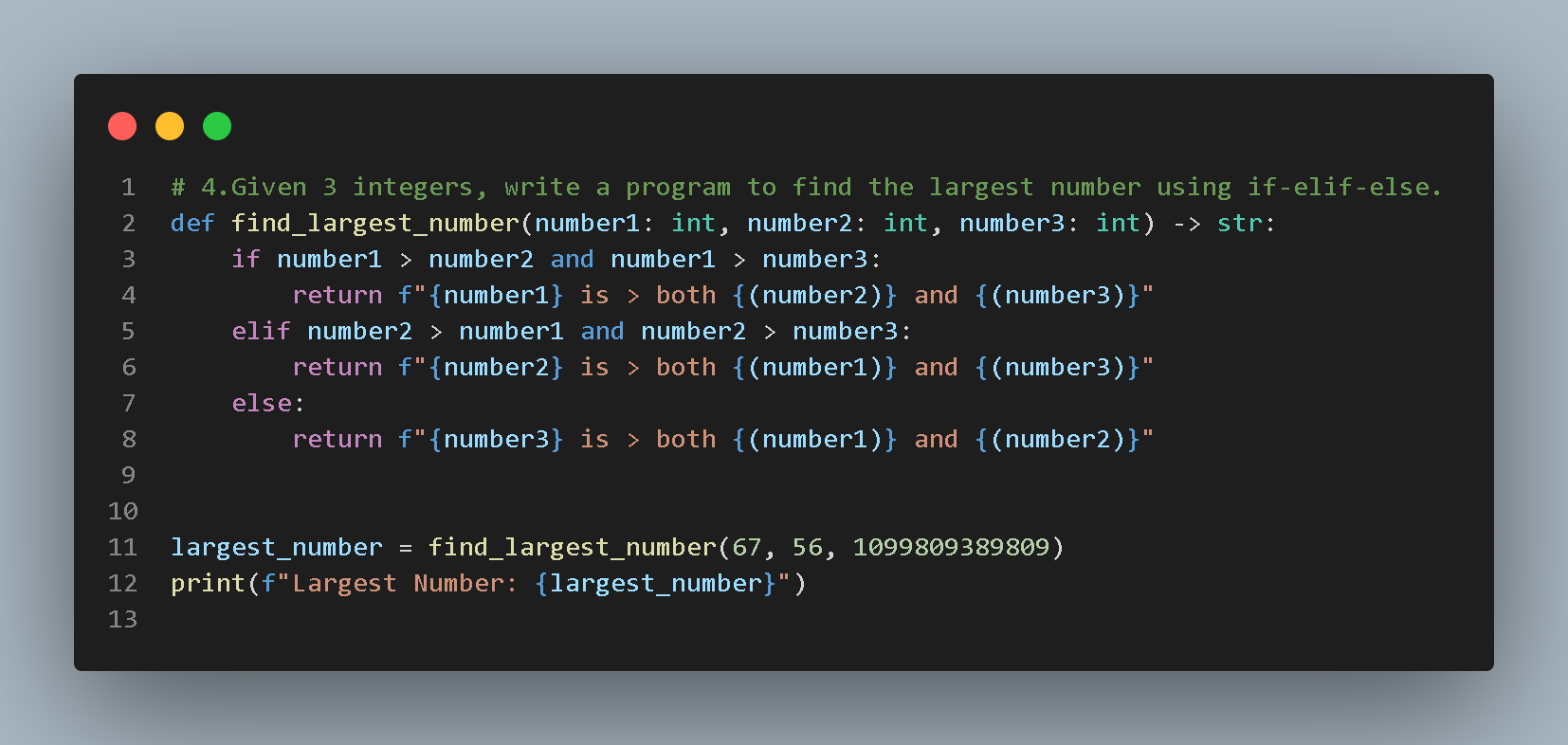
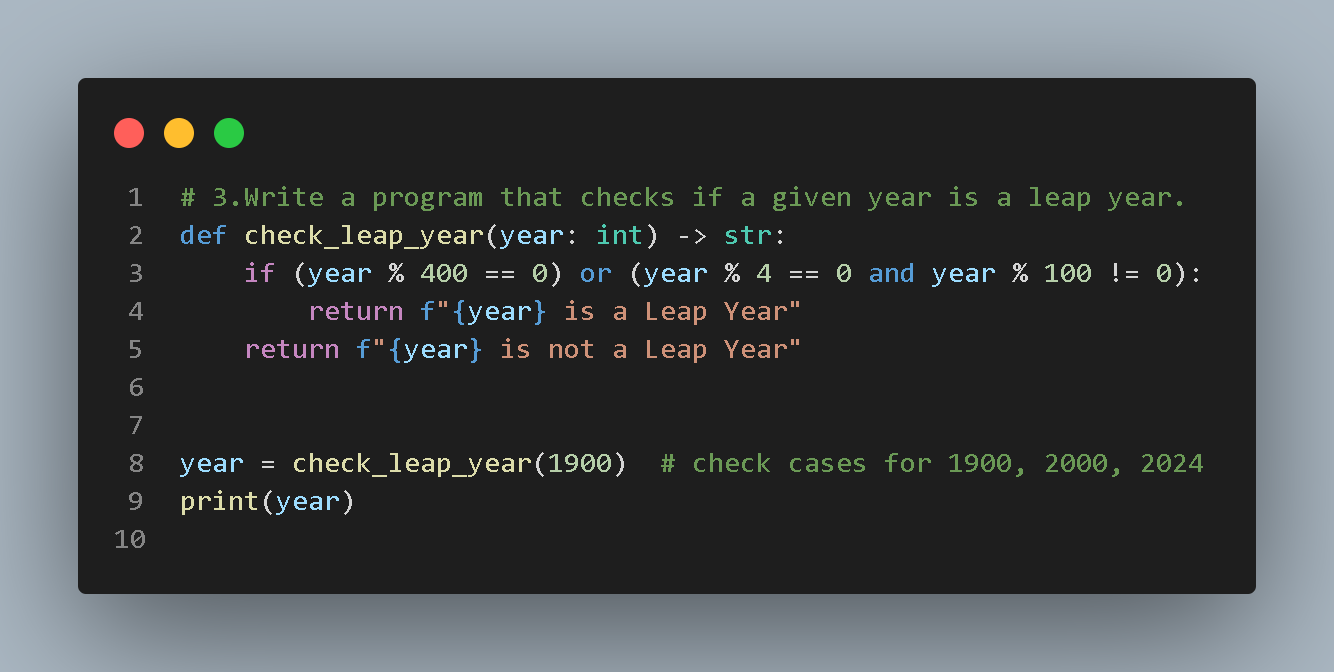
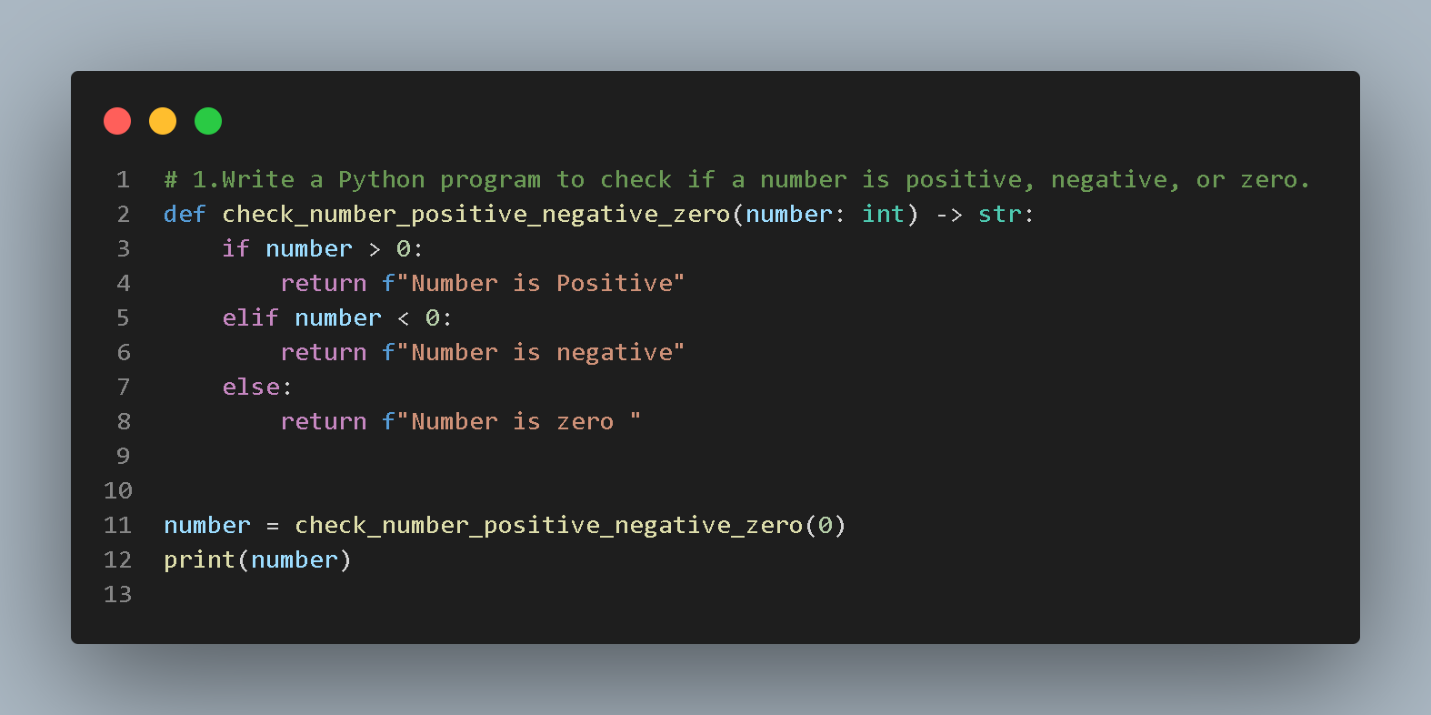
**if not (cgpa >= 6):**

**return "person is not eligible for a scholarship"**

**else:**

**return "Person is eligible for scholarship"**

**print(person\_is\_eligible\_for\_scholarship(3))**

****