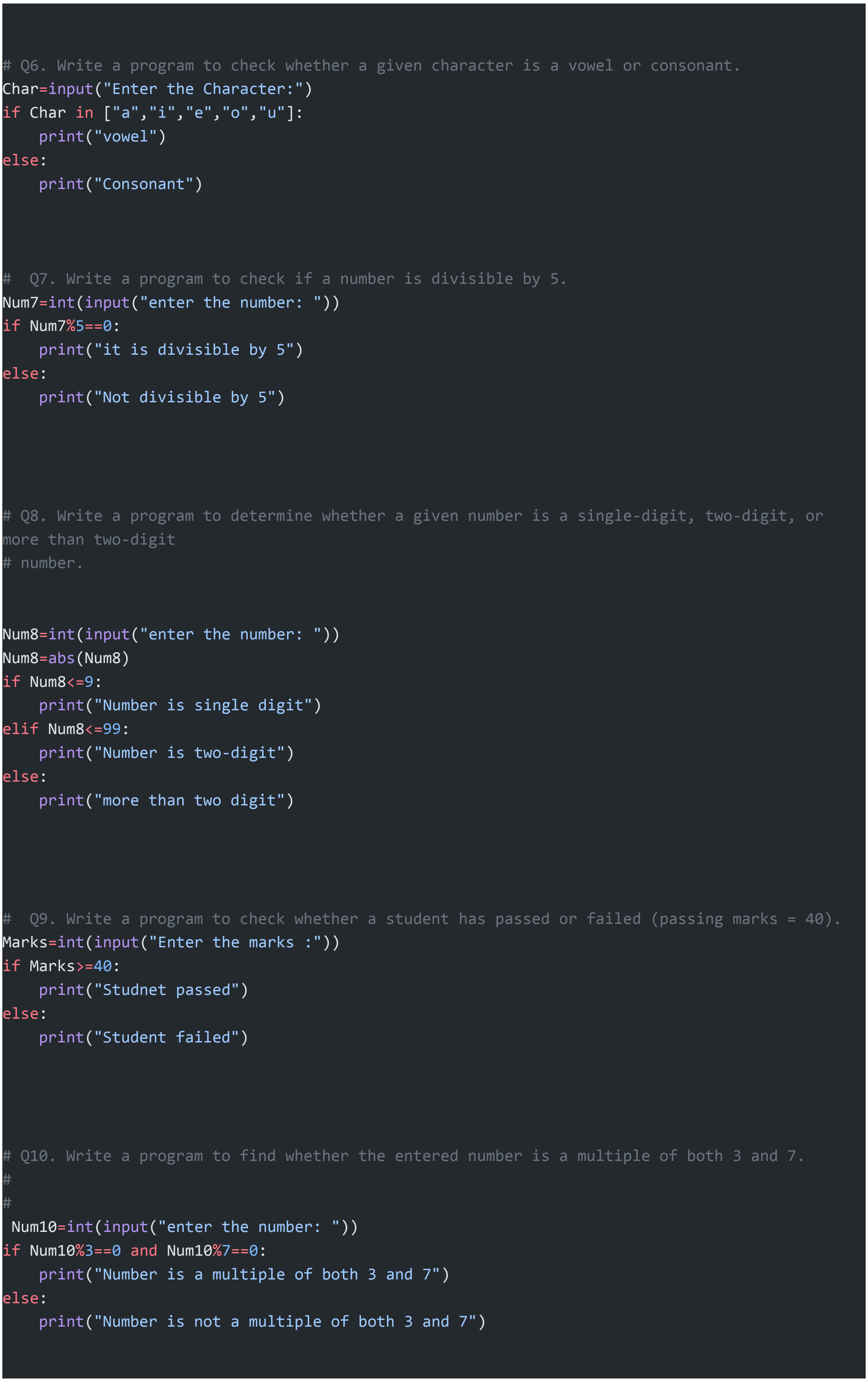
Name: Vedant Wagadre
Enrollment: 0103AL231225
Batch:5
Batch Time: 10:30AM

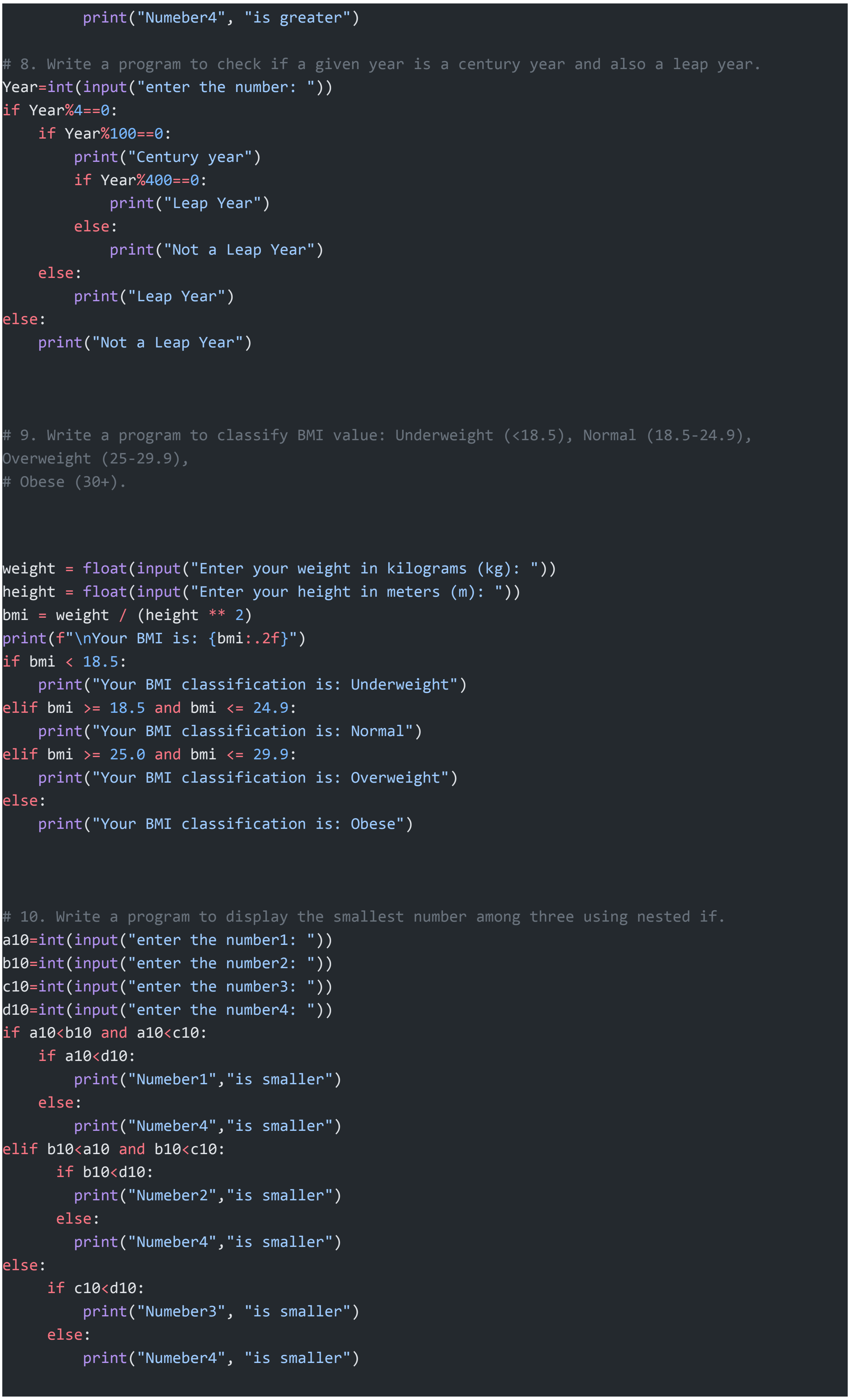
```
# # Name: Vedant Wagadre
# # Enrollment: 0103AL231225
# # Batch:5
# # Batch Time: 10:30


#  Q1. Write a program to check whether a number is positive, negative, or zero.

Num1=int(input("enter the number: "))
if Num1>0:
    print("Number is  positive")
elif Num1<0:
    print("Number is negative")
else:
    print("Number is Zero")


#  Q2. Write a program to check whether a number is even or odd.
Num2=int(input("enter the number: "))
if Num2%2==0:
    print("even")
else:
    print("odd")


#  Q3. Write a program to check if a given year is a leap year or not.
Year=int(input("enter the number: "))
if Year%4==0:
    if Year%100==0:
        if Year%400==0:
            print("Leap Year")
        else:
            print("Not a Leap Year")
    else:
        print("Leap Year")
else:
    print("Not a Leap Year")



#  Q4. Write a program to find the greatest of two numbers.
a=int(input("Enter the Number1:"))
b=int(input("Enter the Number2:"))
if a>b:
    print(a ,"is Greater")
else:
    print(b ,"is Greater")



# Q5. Write a program to check whether a person is eligible to vote (age >= 18).
Age=int(input("Enter your Age :"))
if Age >=18:
    print("You can Vote")
else:
    print("You are Not Eligible to Vote")
```



```
# Q6. Write a program to check whether a given character is a vowel or consonant.
Char=input("Enter the Character:")
if Char in ["a","i","e","o","u"]:
    print("vowel")
else:
    print("Consonant")



#  Q7. Write a program to check if a number is divisible by 5.
Num7=int(input("enter the number: "))
if Num7%5==0:
    print("it is divisible by 5")
else:
    print("Not divisible by 5")




# Q8. Write a program to determine whether a given number is a single-digit, two-digit, or
more than two-digit
# number.


Num8=int(input("enter the number: "))
Num8=abs(Num8)
if Num8<=9:
    print("Number is single digit")
elif Num8<=99:
    print("Number is two-digit")
else:
    print("more than two digit")




#  Q9. Write a program to check whether a student has passed or failed (passing marks = 40).
Marks=int(input("Enter the marks :"))
if Marks>=40:
    print("Studnet passed")
else:
    print("Student failed")




# Q10. Write a program to find whether the entered number is a multiple of both 3 and 7.
#
#
 Num10=int(input("enter the number: "))
if Num10%3==0 and Num10%7==0:
    print("Number is a multiple of both 3 and 7")
else:
    print("Number is not a multiple of both 3 and 7")
```

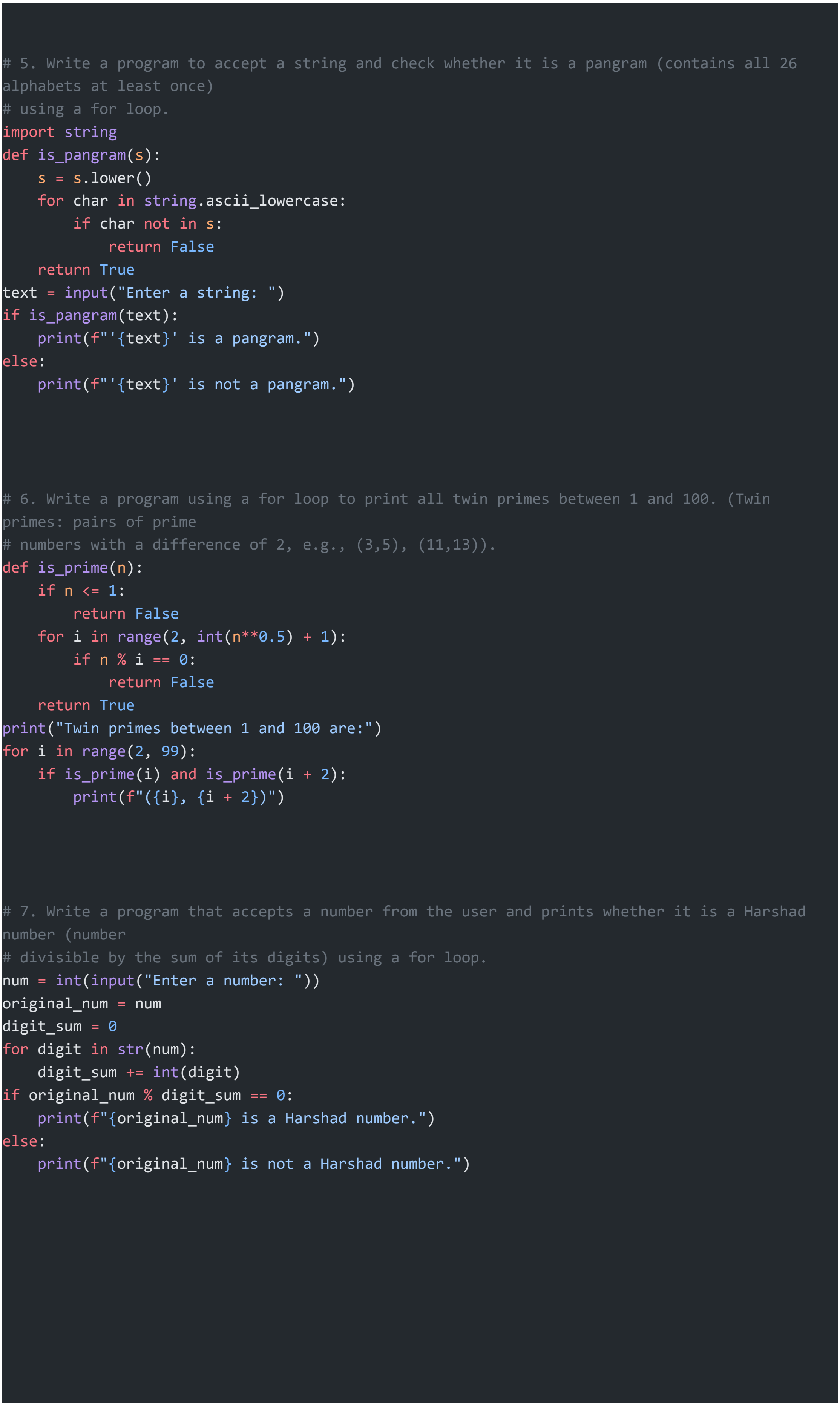
```python
# Ladder If & Nested If:


# 1. Write a program to find the greatest among three numbers.
a1=int(input("enter the numebr: "))
b1=int(input("enter the numebr: "))
c1=int(input("enter the numebr: "))
if a1>b1 and a1>c1:
    print(a1 , "is greater ")
elif b1>a1 and b1 > c1:
    print(b1, "is greater ")
else:
    print(c1 , "is greater")



# 2. Write a program to classify a person based on age: Child (<13), Teenager (13-19), Adult (20-59), Senior (60+).
Age2=int(input("enter the age: "))
if Age2<=13:
    print("child")
elif Age2<=19:
    print("tennager")
elif Age2>=20:
    print("Adult")
else:
    print("Senior")



# 3. Write a program to assign grades based on marks:
# 90-100: A,
# 75-89: B,
# 50-74: C,
# 35-49: D,
# <35: Fail.
Marks3=int(input("enter the marks : "))
if Marks3>=35:
    if Marks3>=50:
        if Marks3>=75:
            if Marks3>=90:
                print("A")
            else:
                print("B")
        else:
            print("C")
    else:
        print("D")
else:
    print("Fail")
```

```
# 4. Write a program to check the type of triangle (equilateral, isosceles, or scalene) based
on sides.
side1=int(input("enter the length of side1: "))
side2=int(input("enter the length of side2: "))
side3=int(input("enter the length of side3: "))
if side1==side2 and side2==side3:
    print("triangle is equilateral triangle")
elif side1==side2 or side1==side3 or side2==side3:
    print("triangle is isosceles triangle")
else:
    print("triangle is scalene triangle")
# 5. Write a program to check if a character is uppercase, lowercase, digit, or special
symbol.
ch = input("Enter a character: ")
if ch.isupper():
    print(ch, "is an Uppercase Letter.")
elif ch.islower():
    print(ch, "is a Lowercase Letter.")
elif ch.isdigit():
    print(ch, "is a Digit.")
else:
    print(ch, "is a Special Symbol.")



# 6. Write a program to calculate electricity bill based on units:
# Up to 100 units:  5 per unit,
# 101-200 units:  7 per unit,
# Above 200 units:  10 per unit.
Unit=int(input("enter your unit:"))
if Unit<=100:
    billamount=Unit*5
elif Unit>100:
    billamount=(Unit-100)*7+100*5
else:
    billamount=(Unit-200)*10+100*5+200*7
print(billamount)



# 7. Write a program to determine the largest of four numbers using nested if.
a7=int(input("enter the number1: "))
b7=int(input("enter the number2: "))
c7=int(input("enter the number3: "))
d7=int(input("enter the number4: "))
if a7>b7 and a7>c7:
    if a7>d7:
        print("Numeber1","is greater")
    else:
        print("Numeber4","is greater")
elif b7>a7 and b7>c7:
      if b7>d7:
        print("Numeber2","is greater")
      else:
        print("Numeber4","is greater")
else:
     if c7>d7:
         print("Numeber3", "is greater")
     else:
```

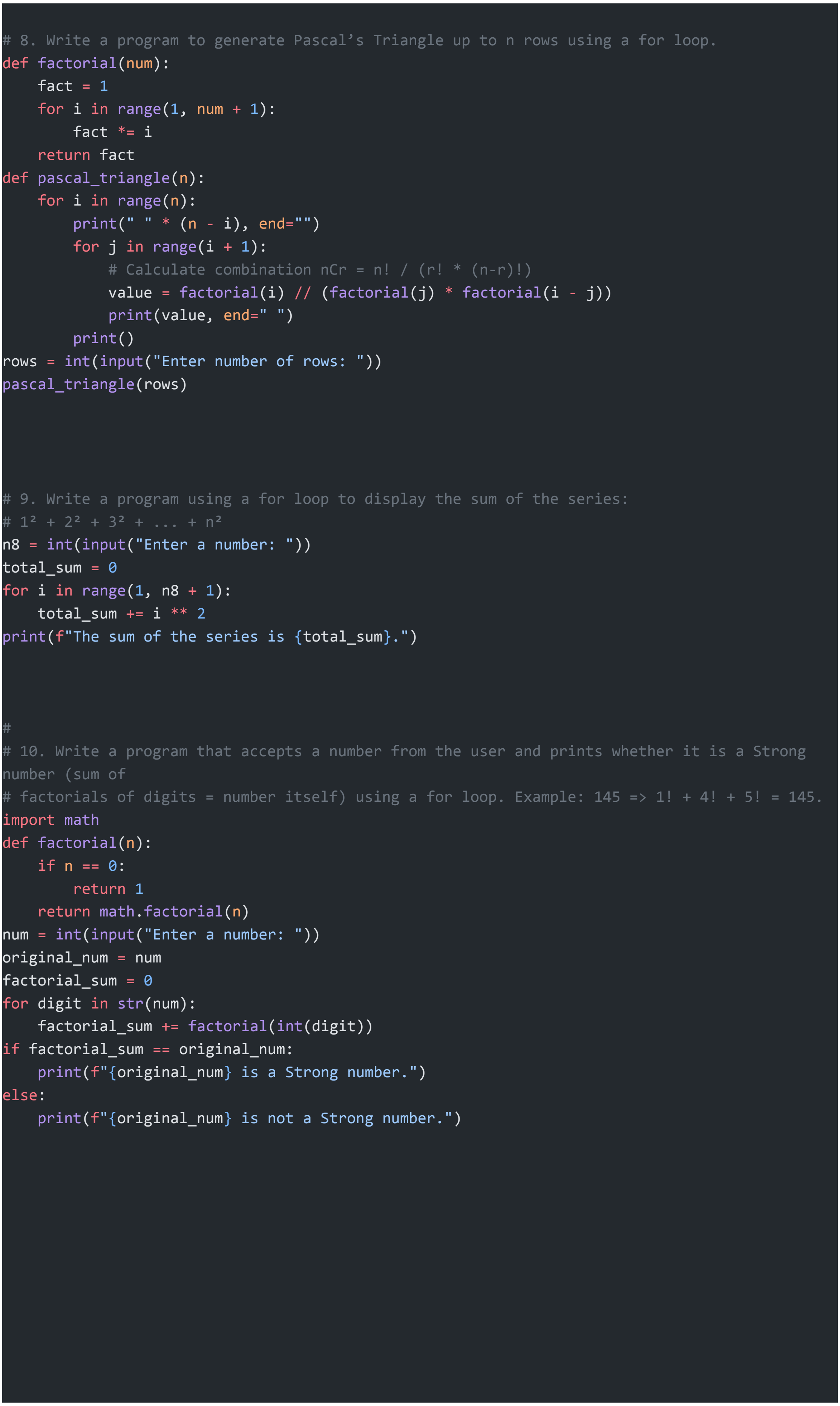


```
         print("Numeber4", "is greater")

# 8. Write a program to check if a given year is a century year and also a leap year.
Year=int(input("enter the number: "))
if Year%4==0:
    if Year%100==0:
        print("Century year")
        if Year%400==0:
            print("Leap Year")
        else:
            print("Not a Leap Year")
    else:
        print("Leap Year")
else:
    print("Not a Leap Year")



# 9. Write a program to classify BMI value: Underweight (<18.5), Normal (18.5-24.9),
Overweight (25-29.9),
# Obese (30+).



weight = float(input("Enter your weight in kilograms (kg): "))
height = float(input("Enter your height in meters (m): "))
bmi = weight / (height ** 2)
print(f"\nYour BMI is: {bmi:.2f}")
if bmi < 18.5:
    print("Your BMI classification is: Underweight")
elif bmi >= 18.5 and bmi <= 24.9:
    print("Your BMI classification is: Normal")
elif bmi >= 25.0 and bmi <= 29.9:
    print("Your BMI classification is: Overweight")
else:
    print("Your BMI classification is: Obese")



# 10. Write a program to display the smallest number among three using nested if.
a10=int(input("enter the number1: "))
b10=int(input("enter the number2: "))
c10=int(input("enter the number3: "))
d10=int(input("enter the number4: "))
if a10<b10 and a10<c10:
    if a10<d10:
        print("Numeber1","is smaller")
    else:
        print("Numeber4","is smaller")
elif b10<a10 and b10<c10:
      if b10<d10:
        print("Numeber2","is smaller")
      else:
        print("Numeber4","is smaller")
else:
     if c10<d10:
         print("Numeber3", "is smaller")
     else:
         print("Numeber4", "is smaller")
```

```
# For Loop Problems:

# 1. Write a program using a for loop to print all Armstrong numbers between 100 and 999.
(Armstrong number:
# sum of cubes of digits equals the number itself. Example: 153 => 1³+5³+3³ = 153).
for i in range (100,1000):
    s=str(i)
    i1=int(int(s[0])**3 + int(s[1])**3 + int(s[2])**3)
    if i1==i:
       print(i1)


# 2. Write a program to generate and display the first n prime numbers using a for loop.
# error
import math
n=int(input("enter the number : "))
am=1
for j in range(2,n):
      for i in range(2, int(math.sqrt(j)) + 1):
        if j % i == 0:
            print("it is not a prime no.", j)
            am=0
            continue
        else:
            print("it is prime no ", j )
      if am==1:
          print("it is prime no ", j )



# 3. Write a program to display all numbers from 1 to 500 that are divisible by 3, but the sum
of their digits
# should not exceed 10.
for i in range(0,501,3):
    w=str(i)
    total3=0
    for j in w:
       total3+=int(j)
    if total3<=10:
        print(i)



# 4. Write a program using a for loop to print a pyramid of stars (*) of height n. Example for
n=4:
#    *
#   ***
#  *****
# *******
n = 4
for i in range(1, n + 1):
    for j in range(n - i):
        print(" ", end="")
    for k in range(2 * i - 1):
        print("*", end="")
    print()
```

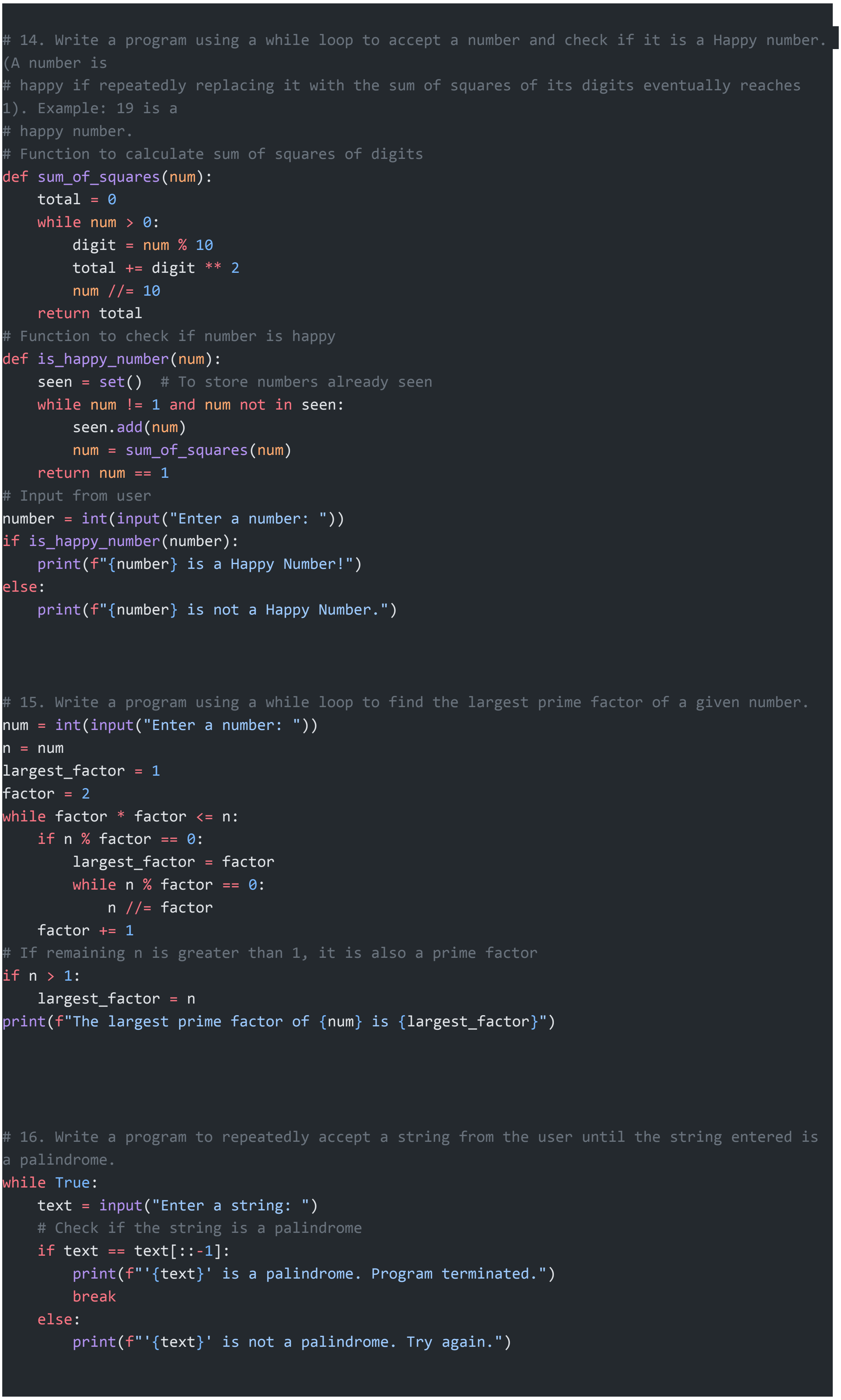


```python
# 5. Write a program to accept a string and check whether it is a pangram (contains all 26
alphabets at least once)
# using a for loop.
import string
def is_pangram(s):
    s = s.lower()
    for char in string.ascii_lowercase:
        if char not in s:
            return False
    return True
text = input("Enter a string: ")
if is_pangram(text):
    print(f"'{text}' is a pangram.")
else:
    print(f"'{text}' is not a pangram.")




# 6. Write a program using a for loop to print all twin primes between 1 and 100. (Twin
primes: pairs of prime
# numbers with a difference of 2, e.g., (3,5), (11,13)).
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
print("Twin primes between 1 and 100 are:")
for i in range(2, 99):
    if is_prime(i) and is_prime(i + 2):
        print(f"({i}, {i + 2})")




# 7. Write a program that accepts a number from the user and prints whether it is a Harshad
number (number
# divisible by the sum of its digits) using a for loop.
num = int(input("Enter a number: "))
original_num = num
digit_sum = 0
for digit in str(num):
    digit_sum += int(digit)
if original_num % digit_sum == 0:
    print(f"{original_num} is a Harshad number.")
else:
    print(f"{original_num} is not a Harshad number.")
```



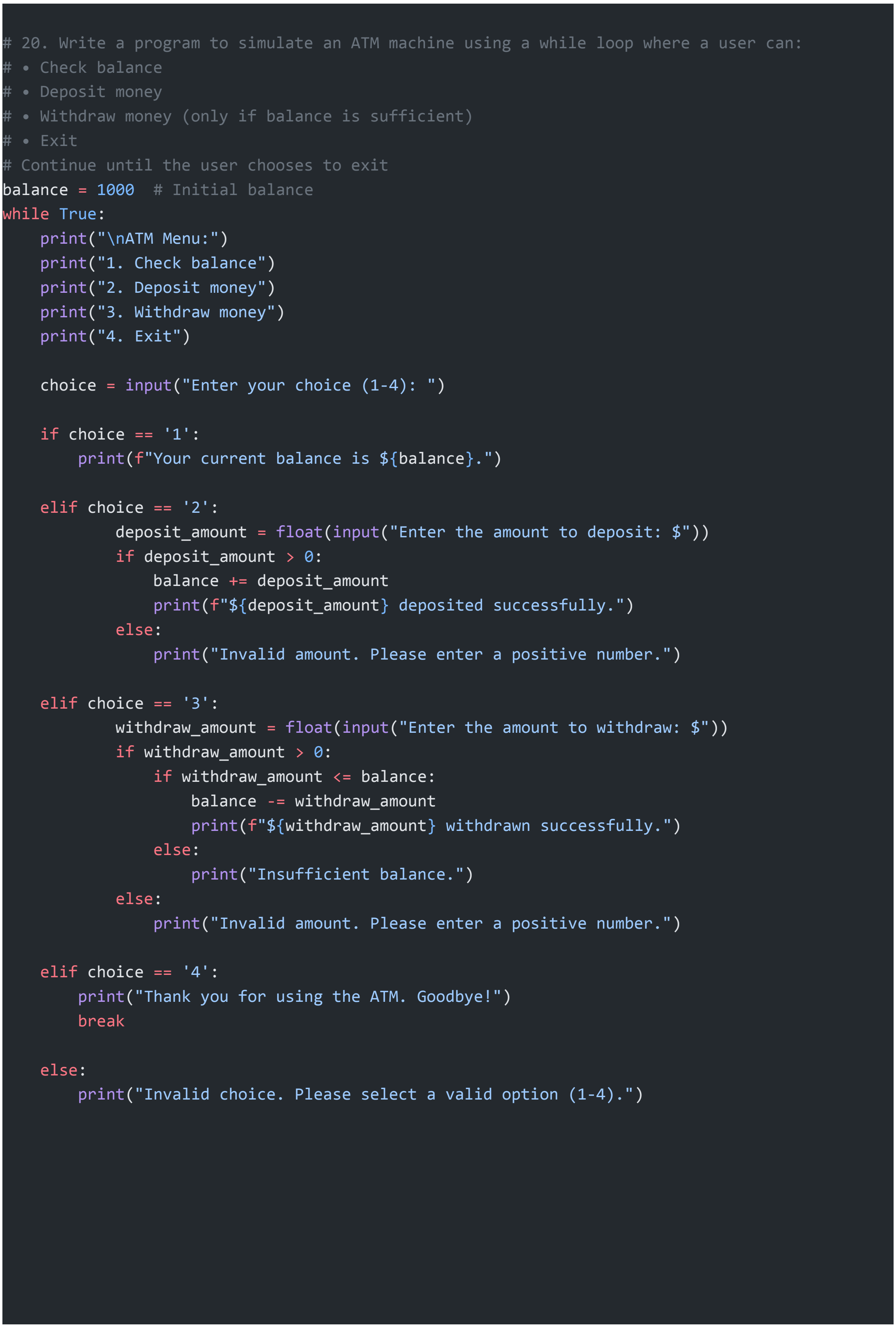
```
# 8. Write a program to generate Pascal’s Triangle up to n rows using a for loop.
def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact
def pascal_triangle(n):
    for i in range(n):
        print(" " * (n - i), end="")
        for j in range(i + 1):
            # Calculate combination nCr = n! / (r! * (n-r)!)
            value = factorial(i) // (factorial(j) * factorial(i - j))
            print(value, end=" ")
        print()
rows = int(input("Enter number of rows: "))
pascal_triangle(rows)




# 9. Write a program using a for loop to display the sum of the series:
# 1² + 2² + 3² + ... + n²
n8 = int(input("Enter a number: "))
total_sum = 0
for i in range(1, n8 + 1):
    total_sum += i ** 2
print(f"The sum of the series is {total_sum}.")




#
# 10. Write a program that accepts a number from the user and prints whether it is a Strong
number (sum of
# factorials of digits = number itself) using a for loop. Example: 145 => 1! + 4! + 5! = 145.
import math
def factorial(n):
    if n == 0:
        return 1
    return math.factorial(n)
num = int(input("Enter a number: "))
original_num = num
factorial_sum = 0
for digit in str(num):
    factorial_sum += factorial(int(digit))
if factorial_sum == original_num:
    print(f"{original_num} is a Strong number.")
else:
    print(f"{original_num} is not a Strong number.")
```

```
# While Loop Problems:

# 11. Write a program using a while loop to find the reverse of a number and check if the
reversed number is
# prime. Example: Input = 73 → Reverse = 37 → Prime.
import math
reverse_num = 0
num11 = int(input("enter the number: "))
# Reverse the number
while num11 > 0:
    digit = num11 % 10
    reverse_num = reverse_num * 10 + digit
    num11 //= 10
is_prime = True
if reverse_num <= 1:
    is_prime = False
else:
    for i in range(2, int(math.sqrt(reverse_num)) + 1):
        if reverse_num % i == 0:
            is_prime = False
            break
if is_prime:
    print(f"The reversed number {reverse_num} is a prime number.")
else:
    print(f"The reversed number {reverse_num} is not a prime number.")



# 12. Write a program that continues to accept numbers from the user until the sum of digits
of all numbers
# entered becomes greater than 100.
total_digit_sum = 0
while total_digit_sum <= 100:
    num_str = input("Enter a number: ")
    current_digit_sum = 0
    for digit in num_str:
        current_digit_sum += int(digit)
    total_digit_sum += current_digit_sum
    print(f"Current total sum of digits: {total_digit_sum}")
print("The total sum of digits has exceeded 100")



# 13. Write a program using a while loop to check whether a number is a Duck number (a number
containing zero
# but not starting with zero, e.g., 202, 1203).
num = input("Enter a number: ")
is_duck = False
i = 1
while i < len(num):
    if num[i] == '0':
        is_duck = True
        break
    i += 1
if is_duck:
    print(f"{num} is a Duck number.")
else:
    print(f"{num} is not a Duck number.")
```



```
# 14. Write a program using a while loop to accept a number and check if it is a Happy number.
(A number is
# happy if repeatedly replacing it with the sum of squares of its digits eventually reaches
1). Example: 19 is a
# happy number.
# Function to calculate sum of squares of digits
def sum_of_squares(num):
    total = 0
    while num > 0:
        digit = num % 10
        total += digit ** 2
        num //= 10
    return total
# Function to check if number is happy
def is_happy_number(num):
    seen = set()  # To store numbers already seen
    while num != 1 and num not in seen:
        seen.add(num)
        num = sum_of_squares(num)
    return num == 1
# Input from user
number = int(input("Enter a number: "))
if is_happy_number(number):
    print(f"{number} is a Happy Number!")
else:
    print(f"{number} is not a Happy Number.")



# 15. Write a program using a while loop to find the largest prime factor of a given number.
num = int(input("Enter a number: "))
n = num
largest_factor = 1
factor = 2
while factor * factor <= n:
    if n % factor == 0:
        largest_factor = factor
        while n % factor == 0:
            n //= factor
    factor += 1
# If remaining n is greater than 1, it is also a prime factor
if n > 1:
    largest_factor = n
print(f"The largest prime factor of {num} is {largest_factor}")




# 16. Write a program to repeatedly accept a string from the user until the string entered is
a palindrome.
while True:
    text = input("Enter a string: ")
    # Check if the string is a palindrome
    if text == text[::-1]:
        print(f"'{text}' is a palindrome. Program terminated.")
        break
    else:
        print(f"'{text}' is not a palindrome. Try again.")
```

```python
# 17. Write a program using a while loop to compute the sum of digits of a number until the
result becomes a
# single-digit number (Digital root). Example: 9875 => 9+8+7+5=29 => 2+9=11 => 1+1=2.
num_str = input("Enter a number: ")
while len(num_str) > 1:
    digit_sum = 0
    for digit in num_str:
        digit_sum += int(digit)
    num_str = str(digit_sum)
    print(f"Sum of digits: {num_str}")
print(f"The single-digit root is {num_str}.")



# 18. Write a program using a while loop to generate the Collatz sequence for a given number.
(Rule: If n is even
# => n/2, if odd => 3n+1. Continue until n=1).
n = int(input("Enter a positive integer: "))
print(f"Collatz sequence for {n}:")
print(n, end="")
while n != 1:
    if n % 2 == 0:
        n //= 2
    else:
        n = 3 * n + 1
    print(f" -> {n}", end="")
print()



# 19. Write a program using a while loop to accept a number and check whether it is a Kaprekar
number.
# (Kaprekar number: if square of the number can be split into two parts whose sum equals the
number.
# Example: 45²=2025 => 20+25=45).
num = int(input("Enter a number: "))
square = num * num
s_square = str(square)
is_kaprekar = False
i = 1
while i < len(s_square):
    part1_str = s_square[:i]
    part2_str = s_square[i:]
    part1 = int(part1_str) if part1_str else 0
    part2 = int(part2_str) if part2_str else 0
    if part1 + part2 == num:
        is_kaprekar = True
        break
    i += 1
if is_kaprekar:
    print(f"{num} is a Kaprekar number.")
else:
    print(f"{num} is not a Kaprekar number.")
```



```python
# 20. Write a program to simulate an ATM machine using a while loop where a user can:
# • Check balance
# • Deposit money
# • Withdraw money (only if balance is sufficient)
# • Exit
# Continue until the user chooses to exit
balance = 1000  # Initial balance
while True:
    print("\nATM Menu:")
    print("1. Check balance")
    print("2. Deposit money")
    print("3. Withdraw money")
    print("4. Exit")

    choice = input("Enter your choice (1-4): ")

    if choice == '1':
        print(f"Your current balance is ${balance}.")

    elif choice == '2':
            deposit_amount = float(input("Enter the amount to deposit: $"))
            if deposit_amount > 0:
                balance += deposit_amount
                print(f"${deposit_amount} deposited successfully.")
            else:
                print("Invalid amount. Please enter a positive number.")

    elif choice == '3':
            withdraw_amount = float(input("Enter the amount to withdraw: $"))
            if withdraw_amount > 0:
                if withdraw_amount <= balance:
                    balance -= withdraw_amount
                    print(f"${withdraw_amount} withdrawn successfully.")
                else:
                    print("Insufficient balance.")
            else:
                print("Invalid amount. Please enter a positive number.")

    elif choice == '4':
        print("Thank you for using the ATM. Goodbye!")
        break

    else:
        print("Invalid choice. Please select a valid option (1-4).")
```