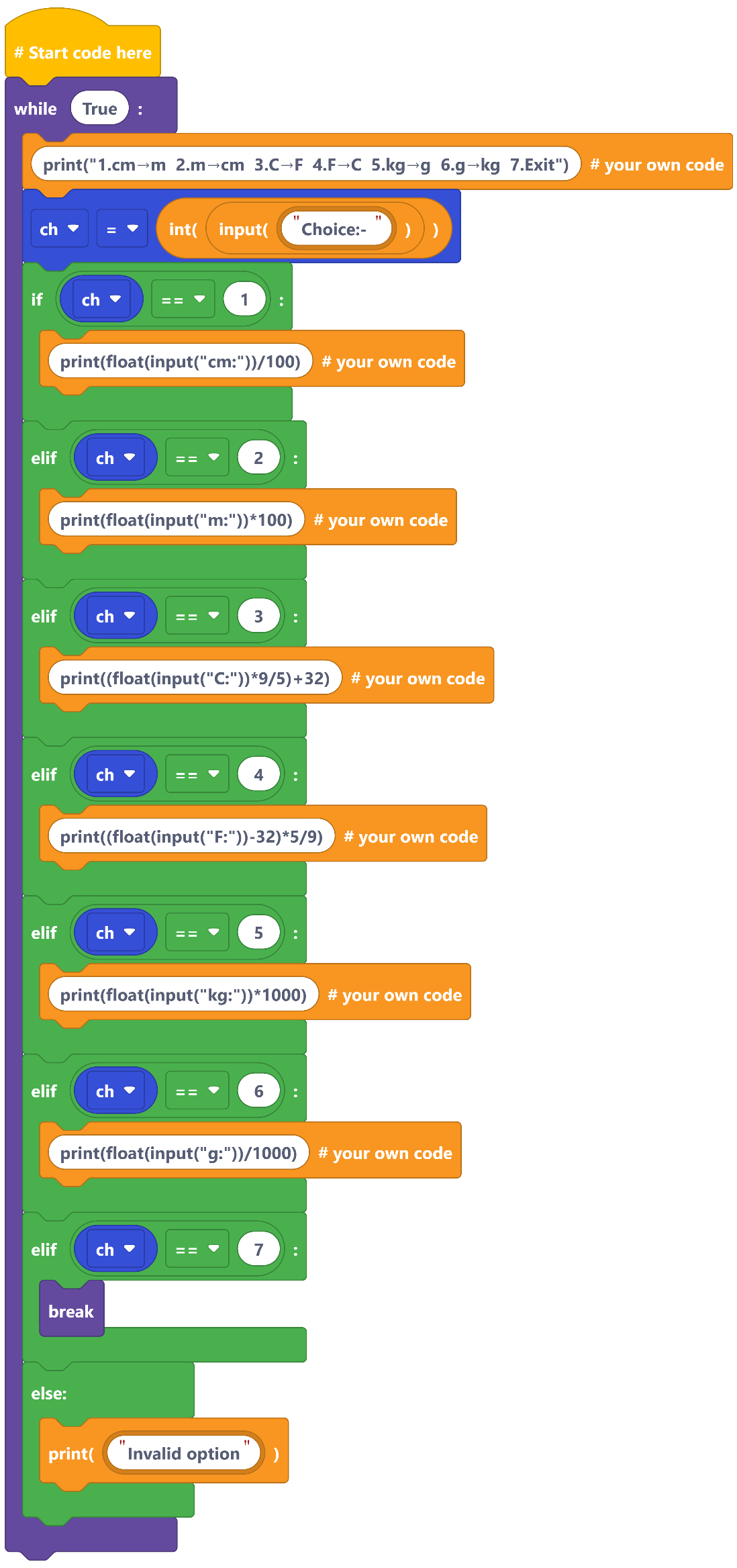
**

*This Python program is a* ***menu-driven calculator****.  
It allows the user to choose an operation (addition, subtraction, multiplication, division, modulus, floor division, exponentiation) from a menu, enter two numbers, and get the result.  
The program runs in a loop until the user selects* ***Exit****.*

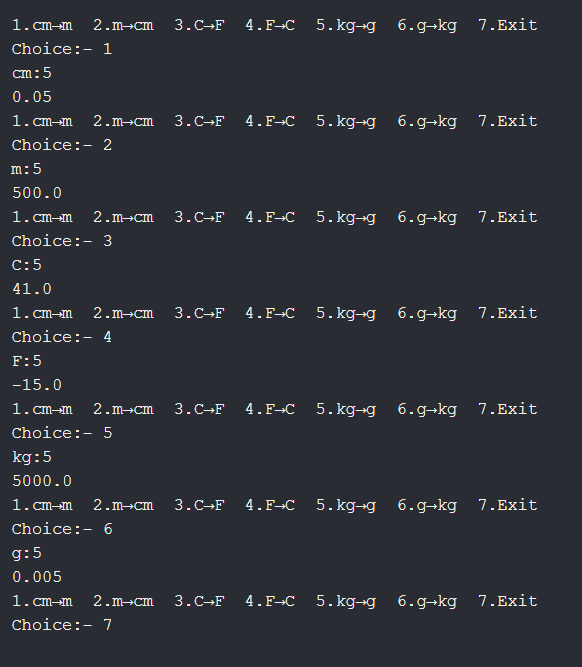
1. **Menu Calculator**

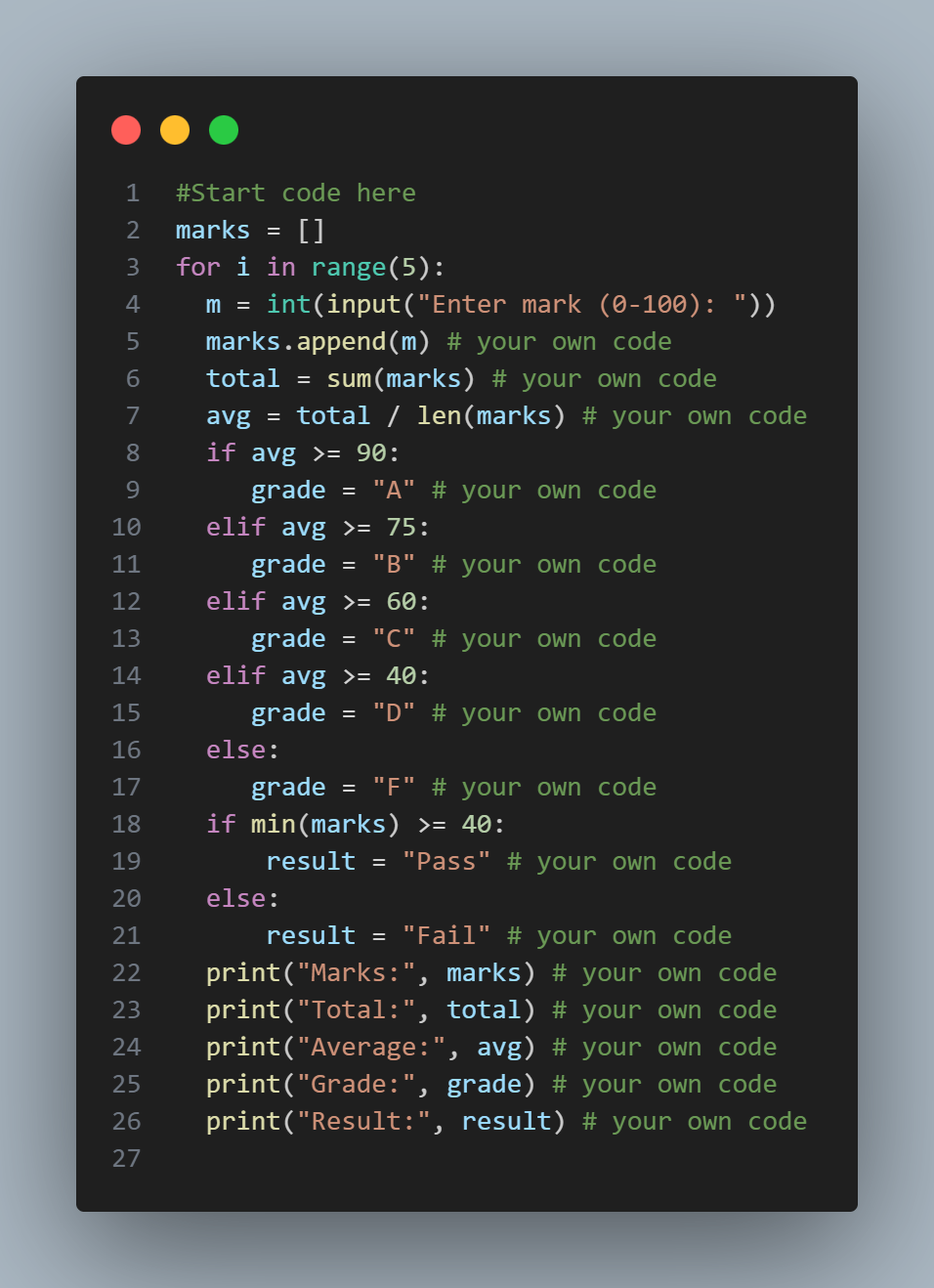
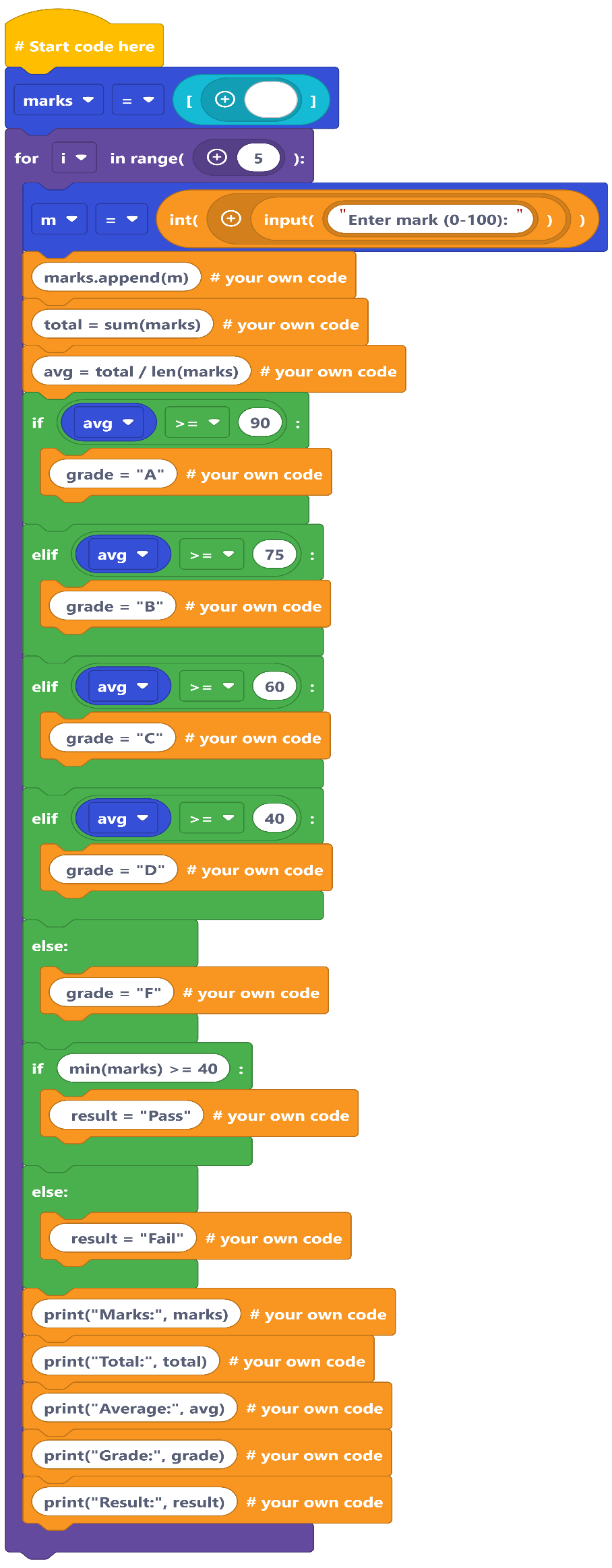
**Task**: Show a menu (+, −, ×, ÷, %, //, \*\*). Read two numbers, perform the chosen operation, repeat until Exit.  
**Blocks**: input/print, while True, if/elif/else, functions with parameters/return, int/float cast.

*This program is a* ***unit converter tool*** *that lets the user convert between* ***length (cm, m), temperature (C, F), and mass (kg, g)*** *interactively.*

1. **Unit Converter Hub**

**Task**: Single menu with small converters: cm↔m, °C↔°F, kg↔g. Repeat until Exit.  
**Blocks**: input, arithmetic, while, if/elif.



 .

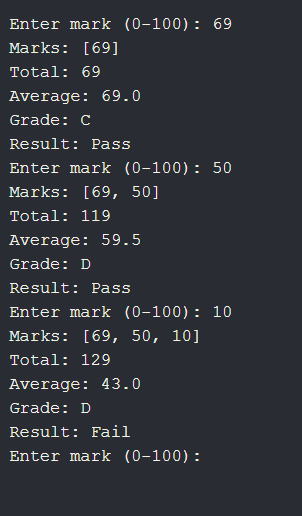
*This program is a* ***unit converter tool*** *that lets the user convert between* ***length (cm, m), temperature (C, F), and mass (kg, g)*** *interactively.*

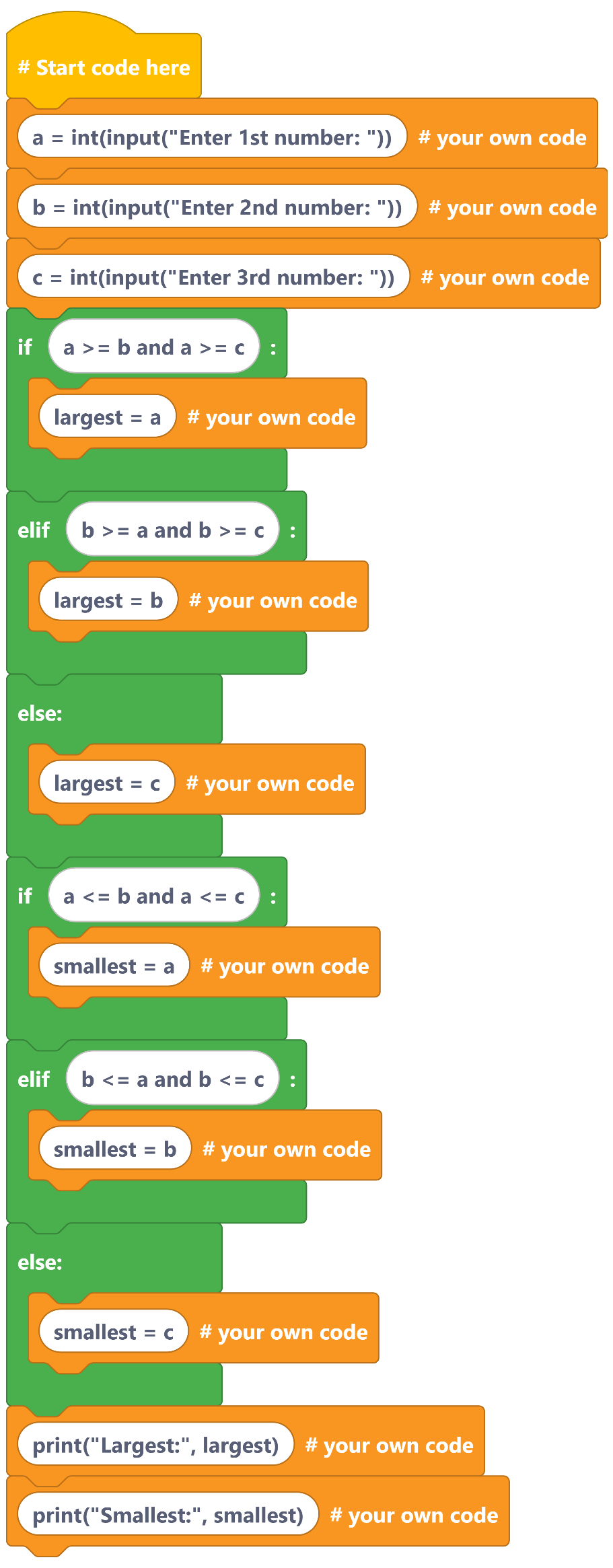
**3. Grading & Result Sheet**

**Task**: Take 5 subject marks (0–100). Compute total, average, grade

(A/B/C/D/F), and pass/fail.  
 **Blocks**: lists (append, len, sum), comparisons, if/elif, print

formatting.



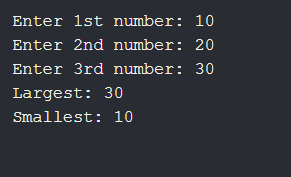
.

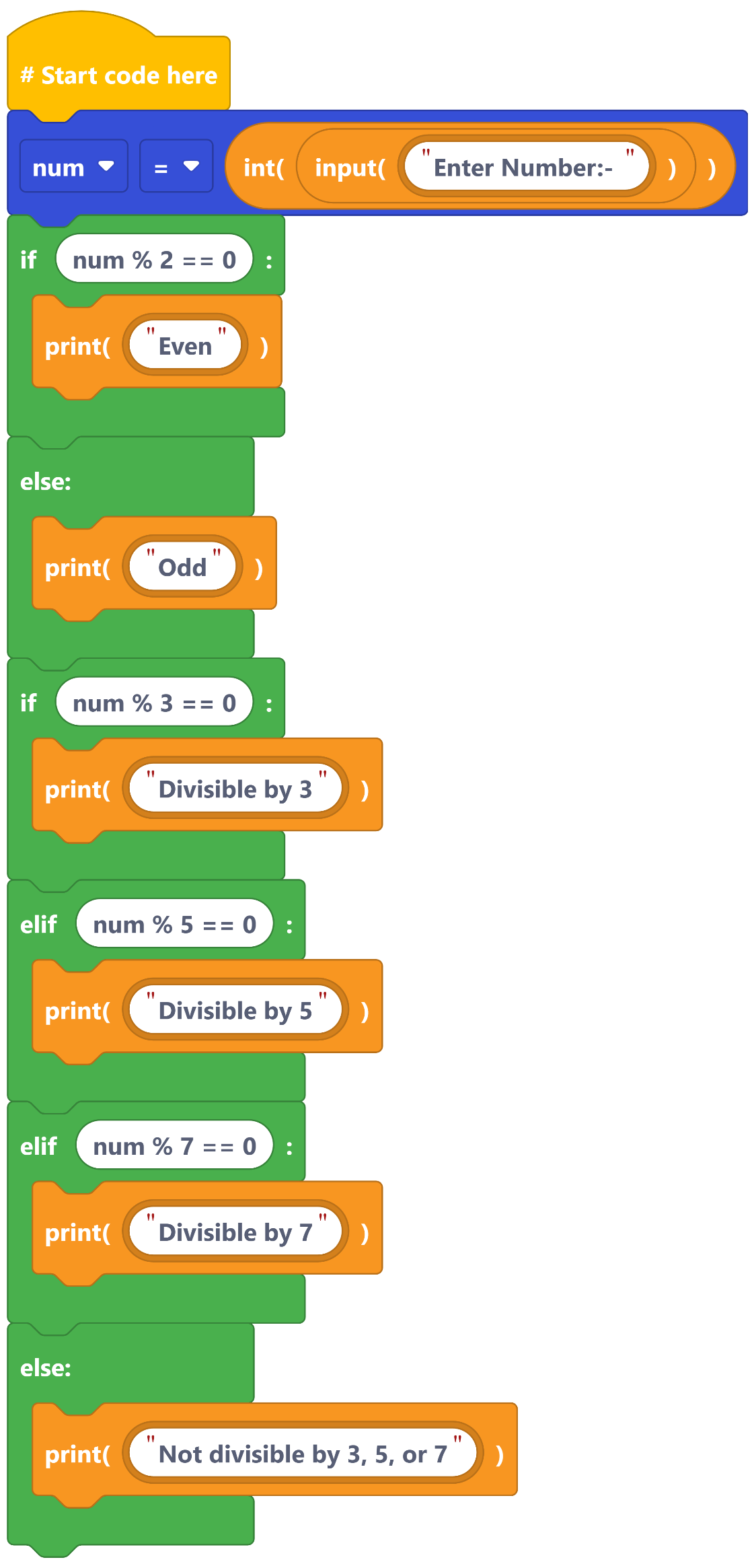
1. *Takes three numbers as input (a, b, c).*
2. *Uses if-elif-else conditions to check:*
3. *Which number is the* ***largest****.Which number is the* ***smallest****.*
4. *Prints both the largest and smallest numbers*
5. ***The program finds and displays the largest and smallest of three numbers entered by the user.***

**4. Max–Min of Three**

**Task**: Read three numbers; print largest and smallest without using built-ins like max/min.  
**Blocks**: if/elif/else, comparison operators.





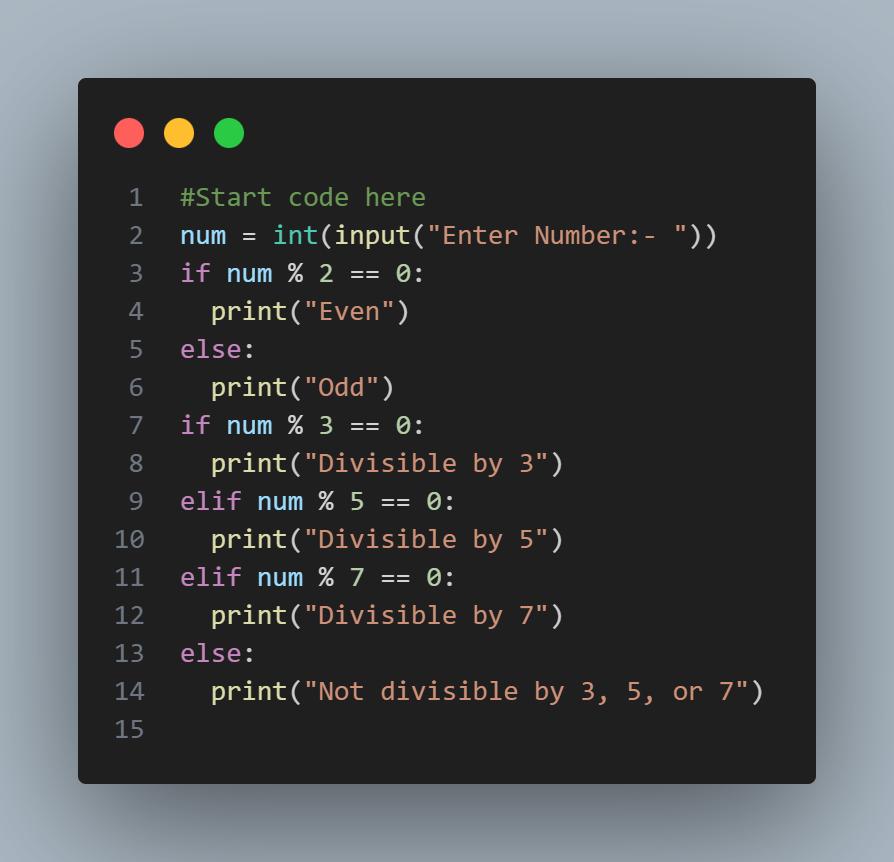
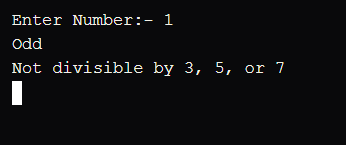
.

1. *Takes a number as input from the user.*
2. *Checks if the number is* ***Even or Odd****.*
3. *Then checks if the number is divisible by* ***3, 5, or 7****.*
4. *Prints which number it is divisible by.*
5. *If not divisible by any, it prints "Not divisible by 3, 5, or 7".*

*This program checks whether a number is* ***even or odd*** *and tests its* ***divisibility by 3, 5, or 7****.*

**5. Even/Odd & Divisibility Checker**

**Task**: Read a number; report even/odd, and divisibility by 3, 5, and 7.  
**Blocks**: modulus %, chained if/elif, logical operators.

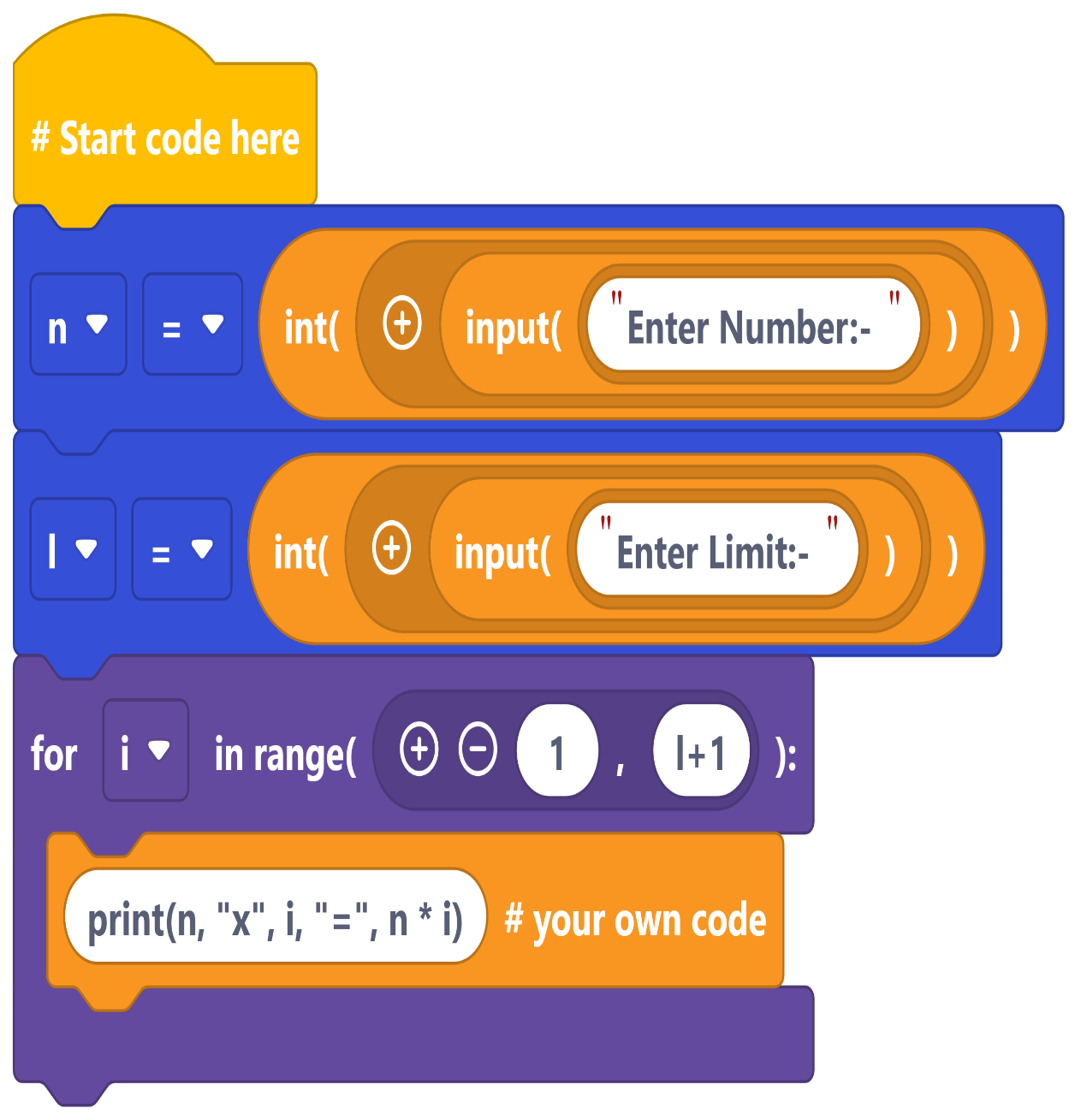
.

**6. Times Table Generator**

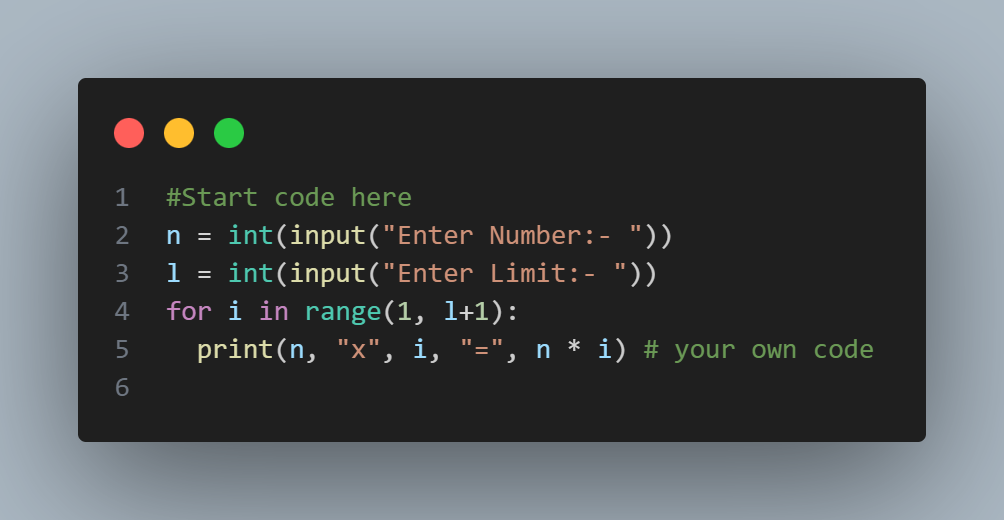
**Task**: Read n and limit L; print the multiplication table of n from 1…L.  
**Blocks**: for loop, multiplication, string formatting.

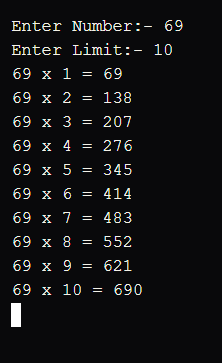
1. *Asks the user for a number n.*
2. *Asks the user for a limit l.*
3. *Uses a loop to print the multiplication table of n from 1 to l.*

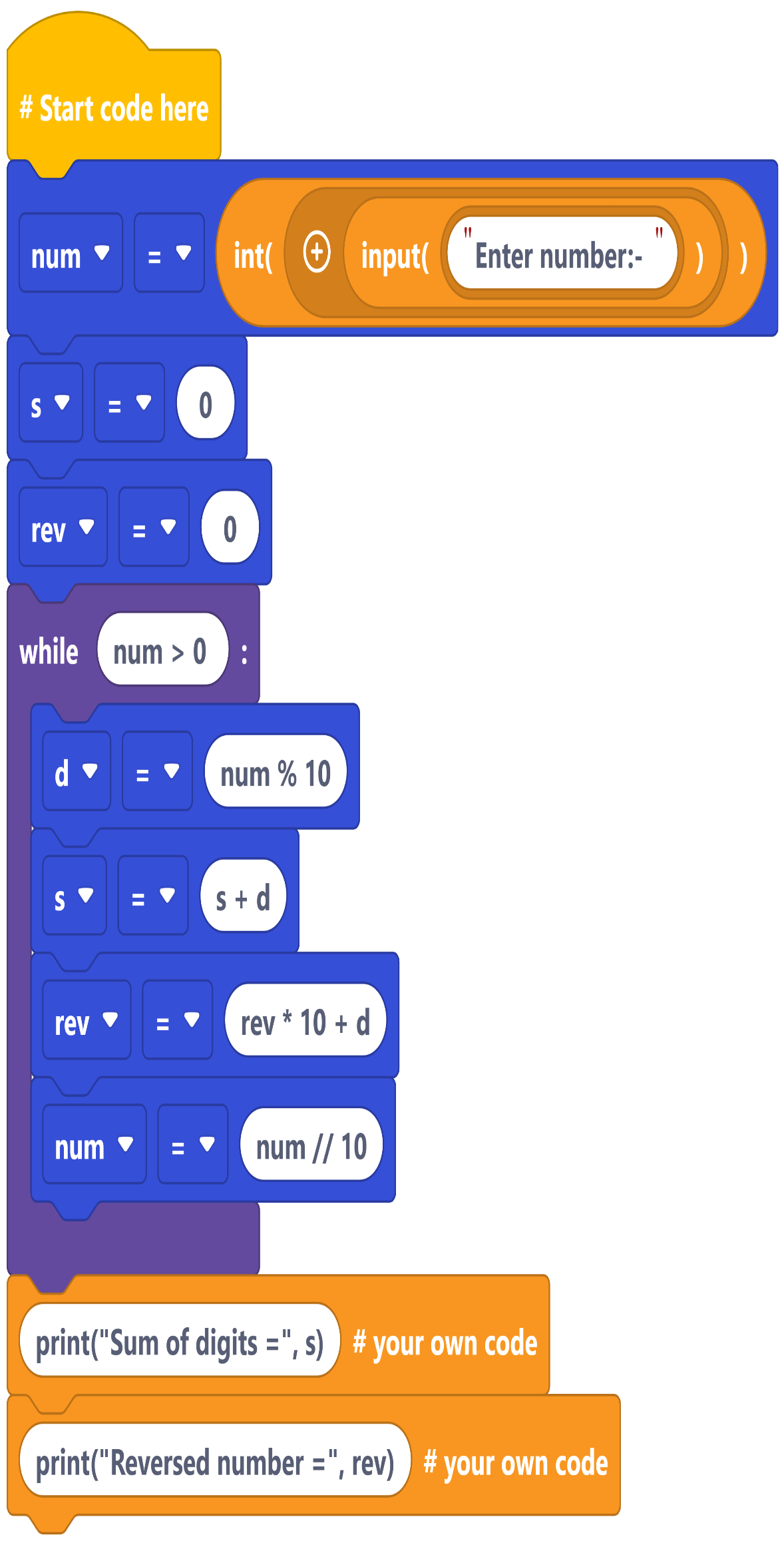
*👉* ***In short (lab manual summary):*** *This program prints the* ***multiplication table*** *of a given number up to a user-defined limit.*



.

.

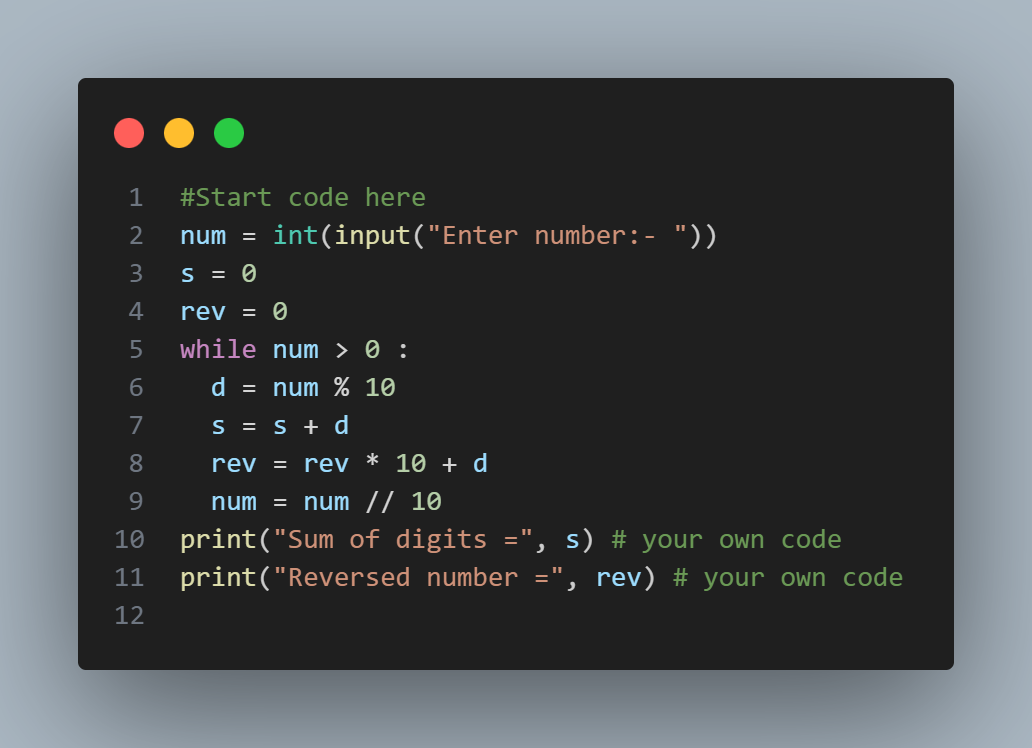


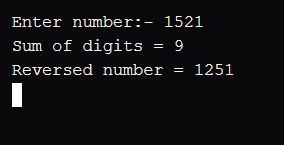


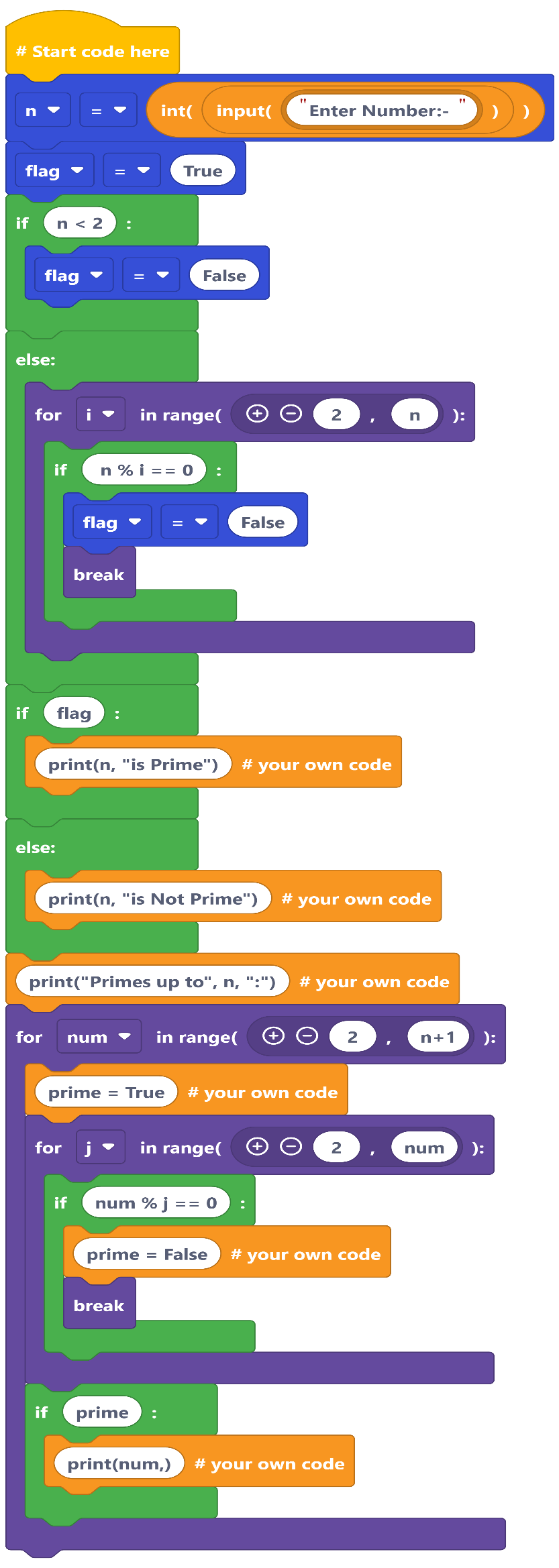
1. *Takes a number as input.*
2. *Uses a loop to extract each digit.*
3. *Adds digits together to find the* ***sum of digits****.*
4. *Builds the digits in reverse order to get the* ***reversed number****.*
5. *Prints both results.*
6. *This program calculates the* ***sum of digits*** *of a number and also prints its* ***reverse****.*

**7. Sum of Digits & Reverse Number**

**Task**: Given an integer, compute sum of digits and reversed number.  
**Blocks**: while loop, %, //, arithmetic, variables.



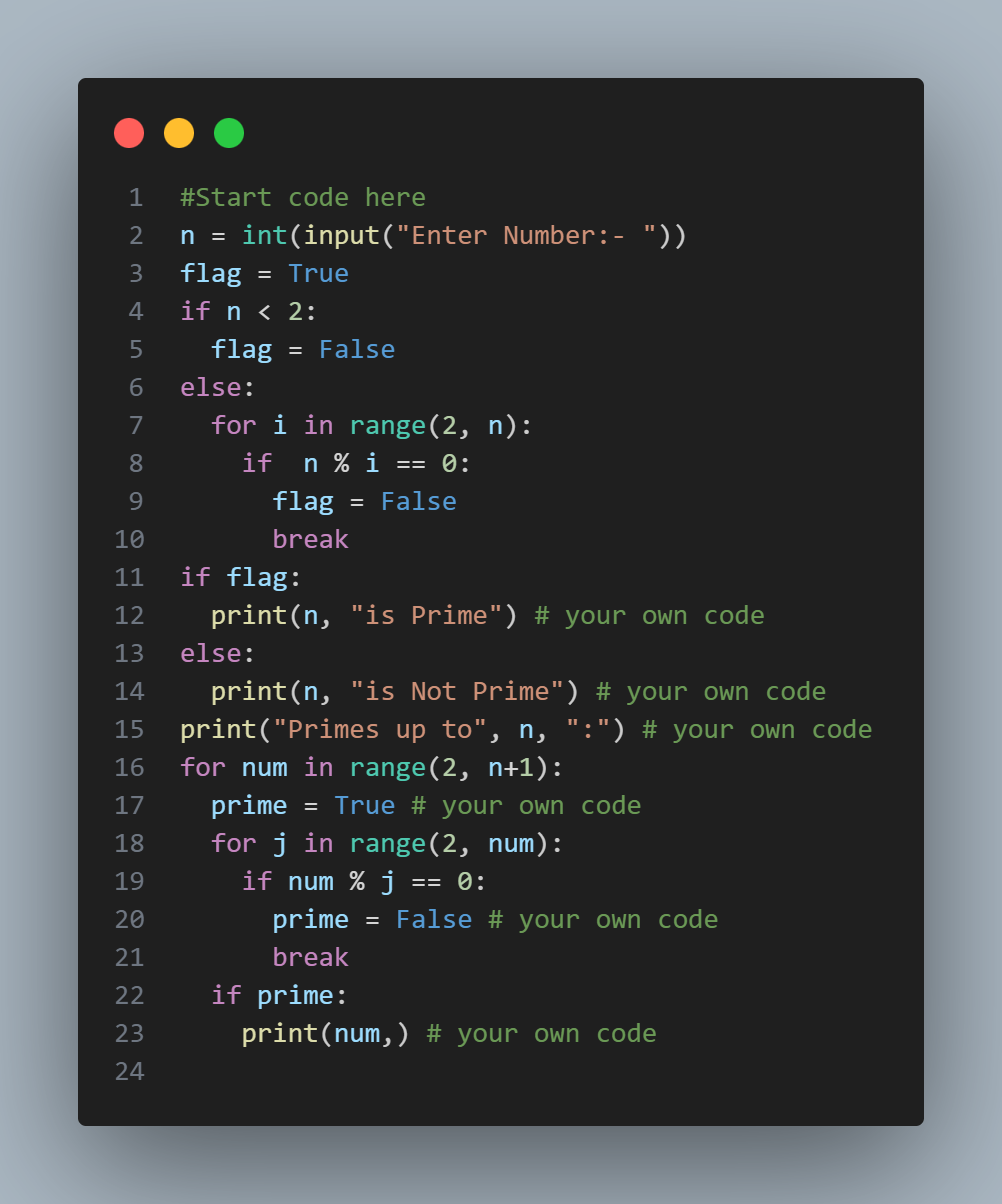


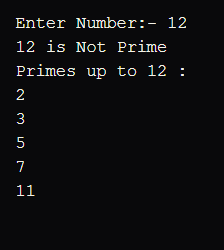


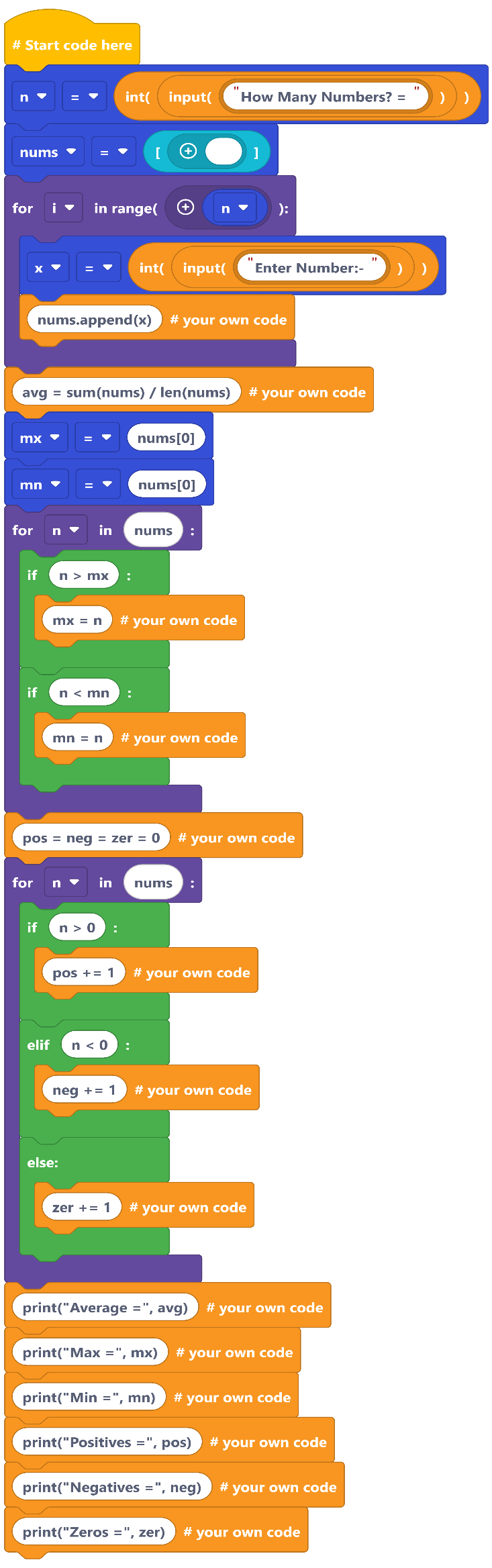
1. *Takes a number n as input.*
2. *Checks whether n is* ***Prime or Not Prime****.*
3. *Prints all* ***prime numbers up to n****.*
4. *This program checks if a given number is* ***prime*** *and also displays all* ***prime numbers up to that number****.*

**8. Prime Checker + List Primes**

**Task**: Check if a number is prime; then list all primes up to N using simple trial division.  
**Blocks**: nested loops, counters/flags, if/else.



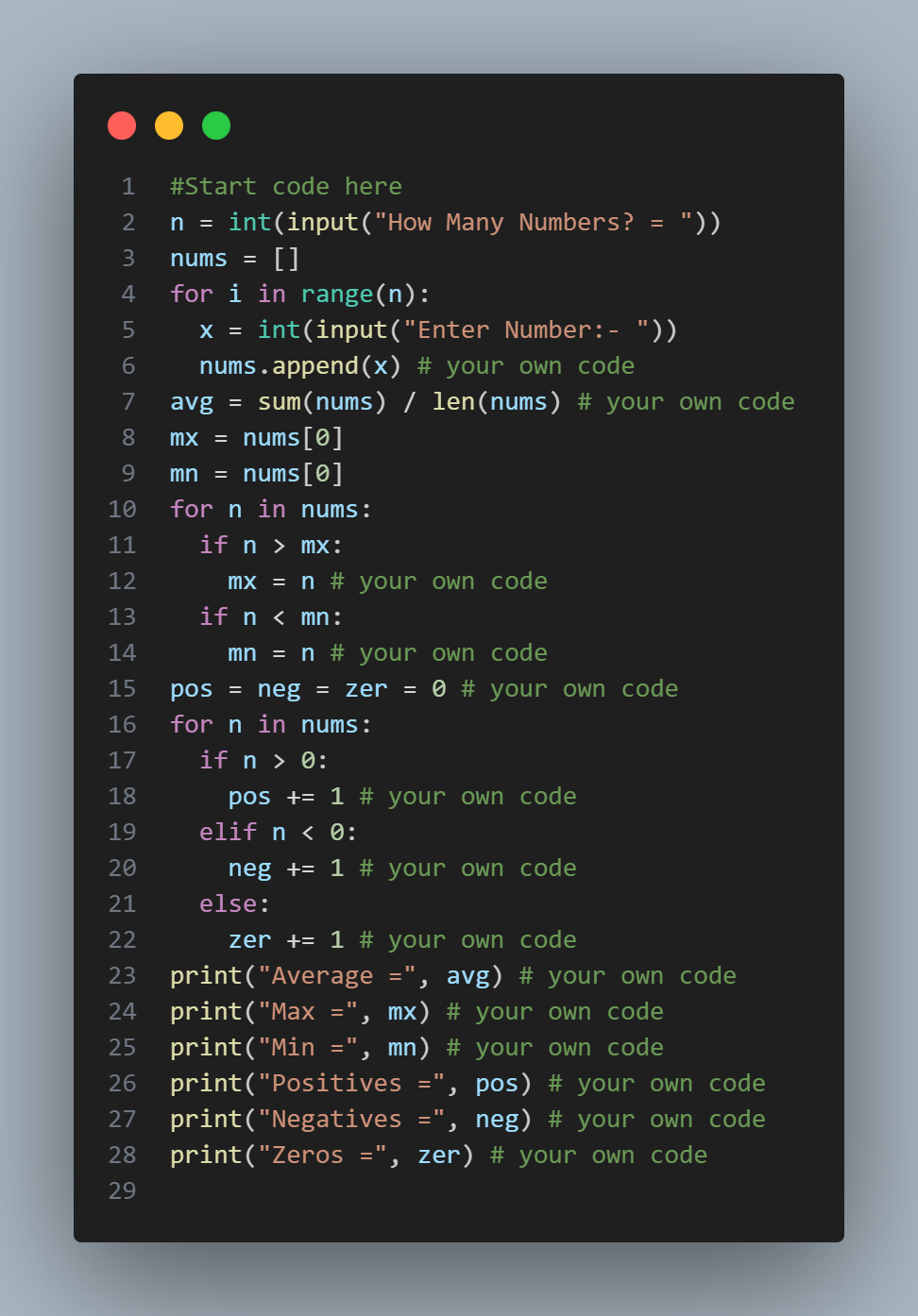


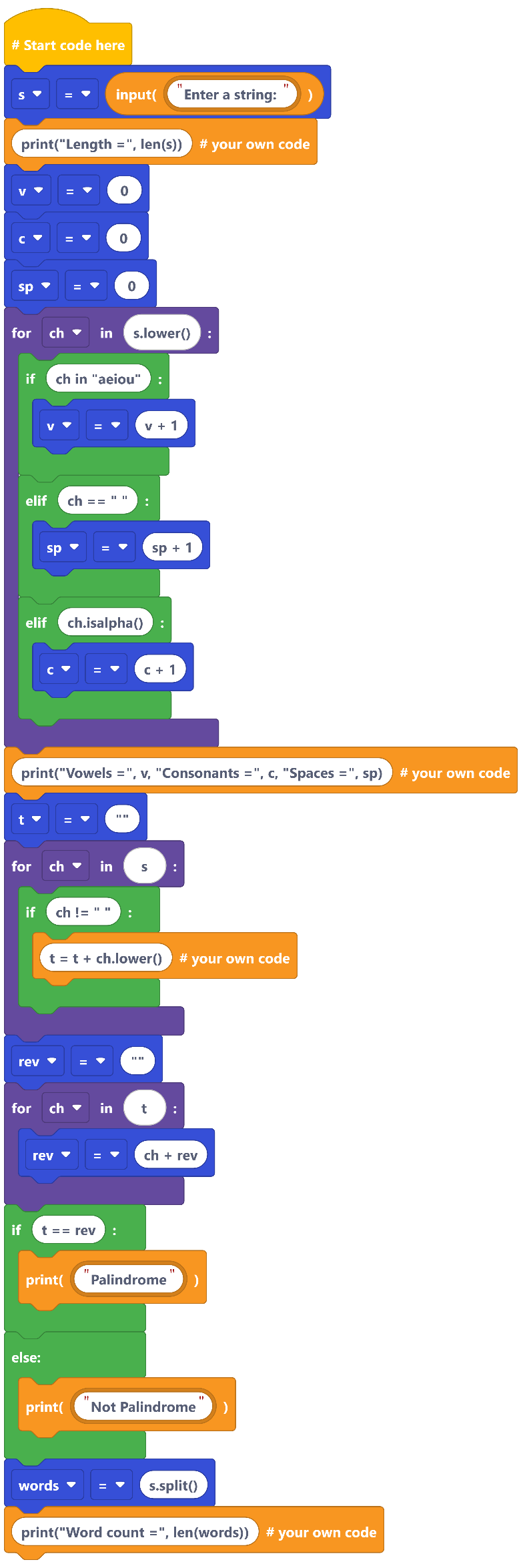
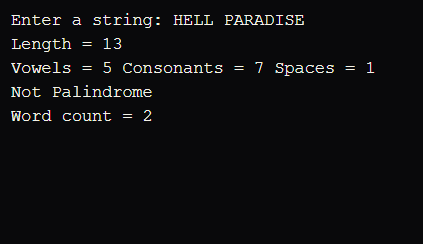


1. *Asks the user how many numbers they want to enter.*
2. *Stores all numbers in a list.*
3. *Calculates:*
4. ***Average*** *of numbers*
5. ***Maximum*** *number*
6. ***Minimum*** *number*
7. *Counts how many numbers are* ***positive, negative, and zeros****.*
8. *Displays all results.*
9. *This program finds the* ***average, maximum, minimum, and counts positives, negatives, and zeros*** *from a list of numbers.*

**9. Number List Analyzer**

**Task**: Read N numbers into a list; print avg, max, min, count positives/negatives/zeros.  
**Blocks:** lists (append, indexing), loops, comparisons, sum/len.

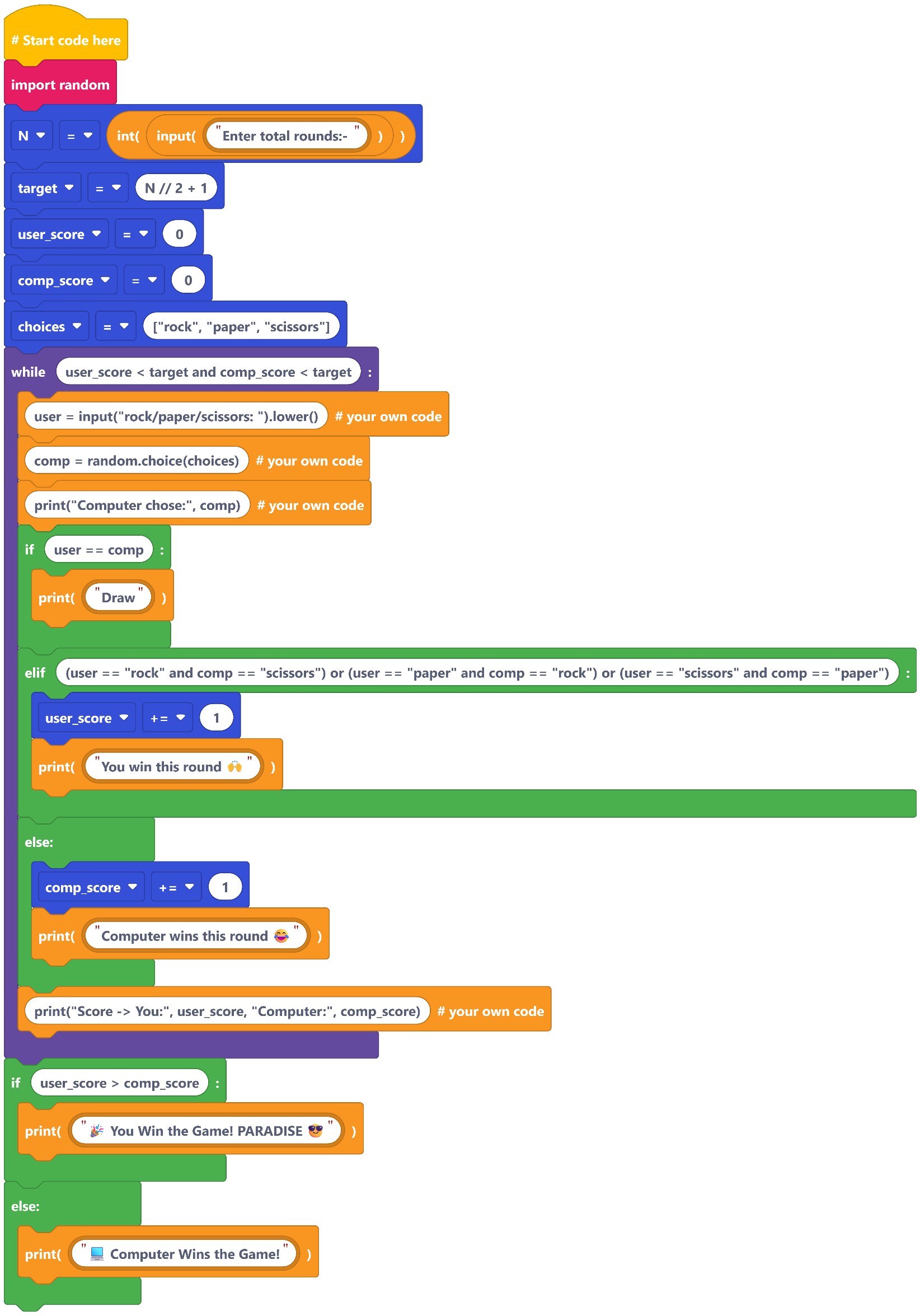


.  ..

**10. String Utility Box**

**Task**: Menu with: length of string, count vowels/consonants/spaces, palindrome test (ignore spaces, case), word count.  
**Blocks**: string lower/split, for loops, membership checks, if/elif.

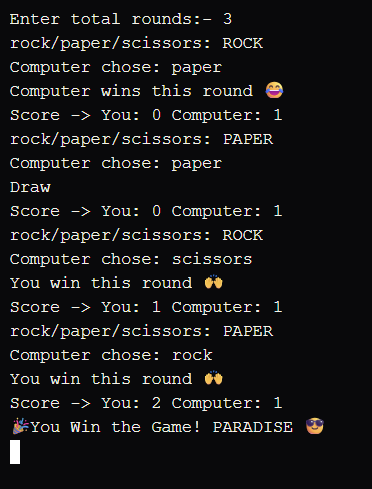
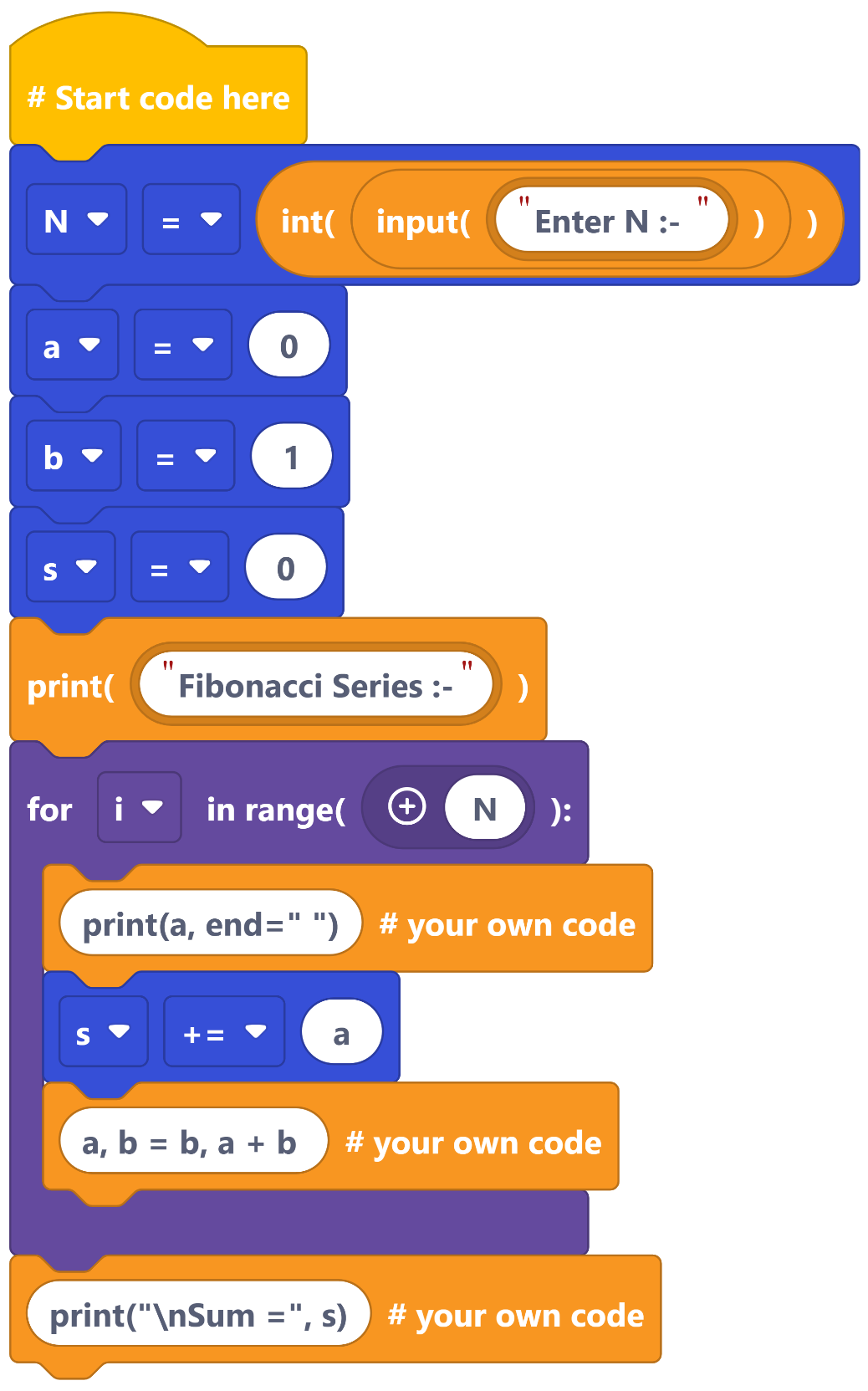
*This Python program takes a user-input string and performs several operations: it calculates the string's length, counts vowels, consonants, and spaces, checks if the string is a palindrome (ignoring spaces and case), and counts the number of words. It helps demonstrate basic string handling, loops, and conditions in Python.*



*This Python program is a Rock-Paper-Scissors game where the user plays against the computer. It takes the number of rounds as input, plays until one reaches the winning score, and displays the winner at the end. It uses loops, conditionals, and random choice for gameplay logic.*

**11. Rock–Paper–Scissors (Best of N)**

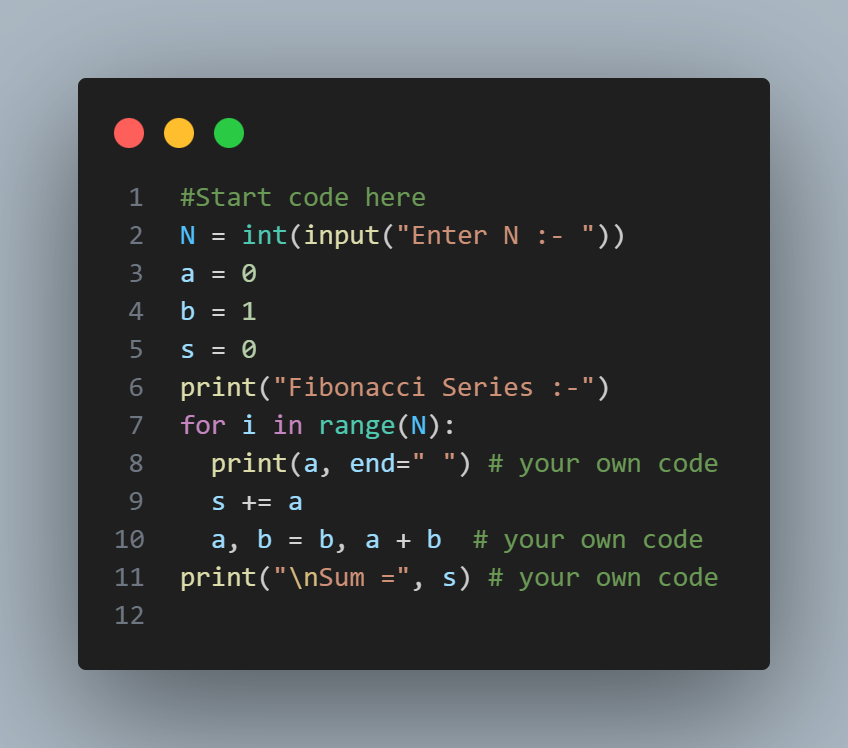
**Task**: User vs computer; keep score until someone reaches N/2+1 wins.  
**Blocks**: random.choice, while loop, if/elif, counters.

....

*This Python program prints the* ***Fibonacci series*** *up to* ***N terms****, where N is entered by the user. It also calculates and displays the* ***sum*** *of the series. The code uses a loop to generate the sequence and update the sum with each term.*

**12. Fibonacci Series (Iterative)**

**Task**: Print first N terms and their sum using a loop (no recursion).  
**Blocks:** for loop, multiple assignment (prev, curr), addition.

..

