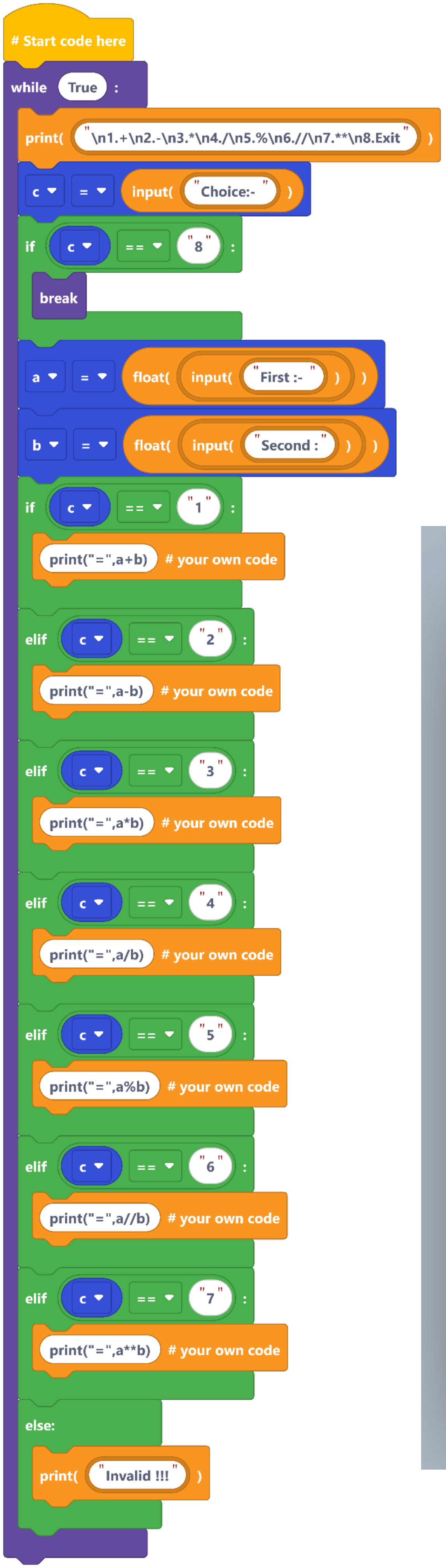
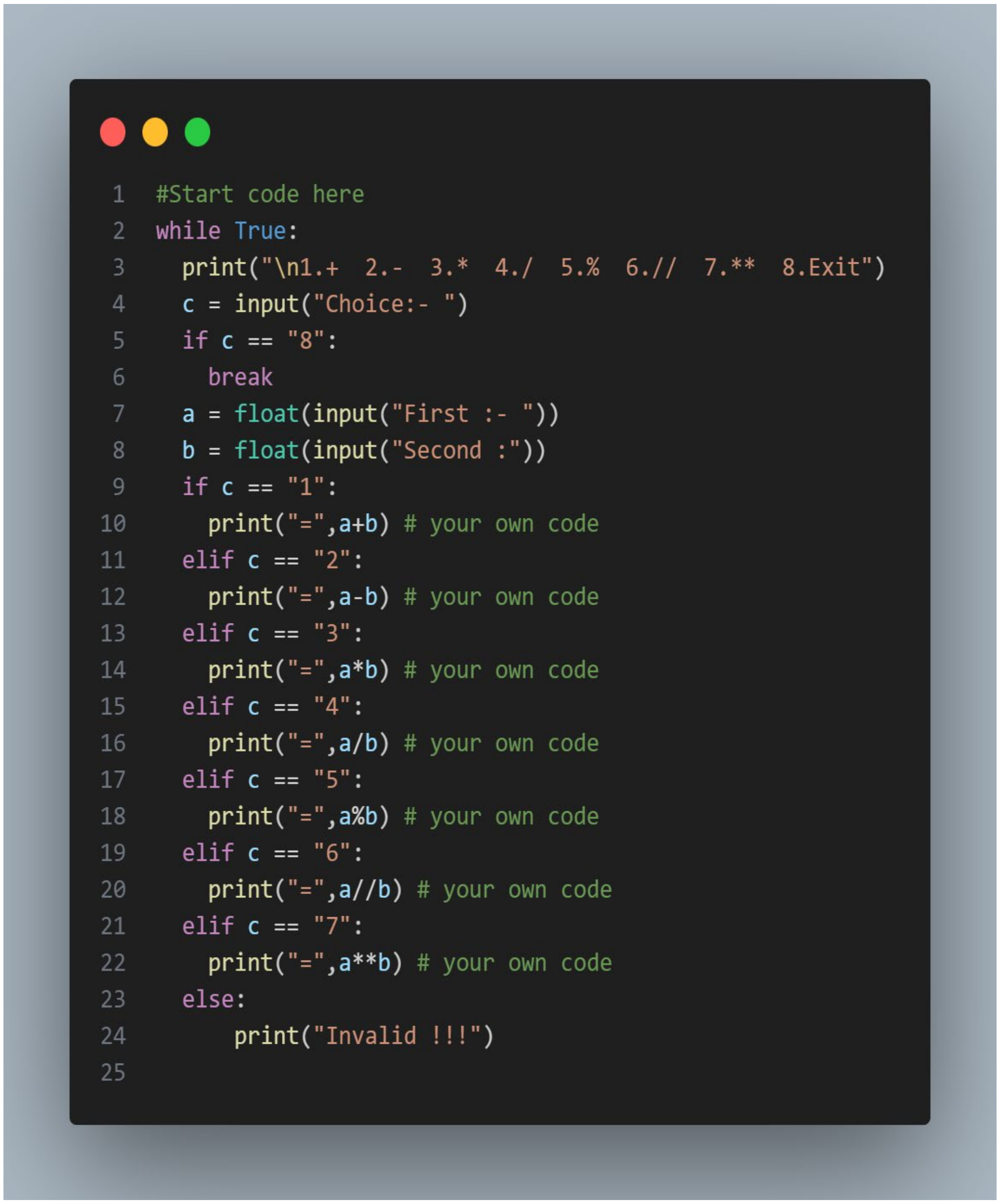
1. Menu Calculator

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.

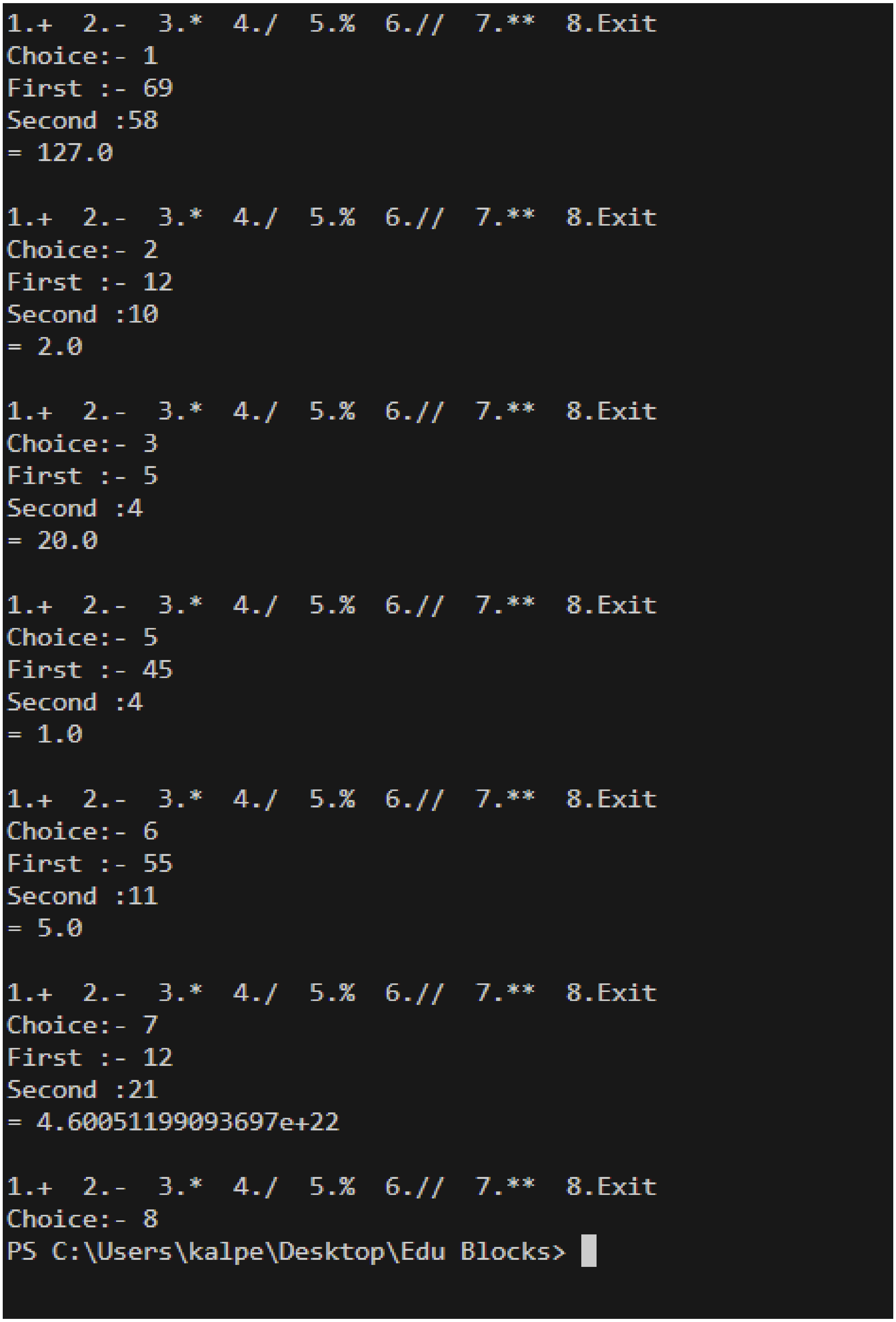
*This Python program is a **menu-driven calculator**.*
It allows the user to choose an operation (addition, subtraction, multiplication, division, modulus, floor division, exponentiation) from a menu, enter two numbers, and get the result.
*The program runs in a loop until the user selects **Exit**.*



```
# Start code here
while True :
  print( "\n1.+\n2.-\n3.*\n4./\n5.%\n6.//\n7.**\n8.Exit" )
  c = input( "Choice:- " )
  if c == "8" :
    break
  a = float( input( "First :- " ) )
  b = float( input( "Second : " ) )
  if c == "1" :
    print("=",a+b) # your own code
  elif c == "2" :
    print("=",a-b) # your own code
  elif c == "3" :
    print("=",a*b) # your own code
  elif c == "4" :
    print("=",a/b) # your own code
  elif c == "5" :
    print("=",a%b) # your own code
  elif c == "6" :
    print("=",a//b) # your own code
  elif c == "7" :
    print("=",a**b) # your own code
  else:
    print( "Invalid !!!" )
```



```
#Start code here
while True:
  print("\n1.+  2.-  3.*  4./  5.%  6.//  7.**  8.Exit")
  c = input("Choice:- ")
  if c == "8":
    break
  a = float(input("First :- "))
  b = float(input("Second :"))
  if c == "1":
    print("=",a+b) # your own code
  elif c == "2":
    print("=",a-b) # your own code
  elif c == "3":
    print("=",a*b) # your own code
  elif c == "4":
    print("=",a/b) # your own code
  elif c == "5":
    print("=",a%b) # your own code
  elif c == "6":
    print("=",a//b) # your own code
  elif c == "7":
    print("=",a**b) # your own code
  else:
      print("Invalid !!!")
```

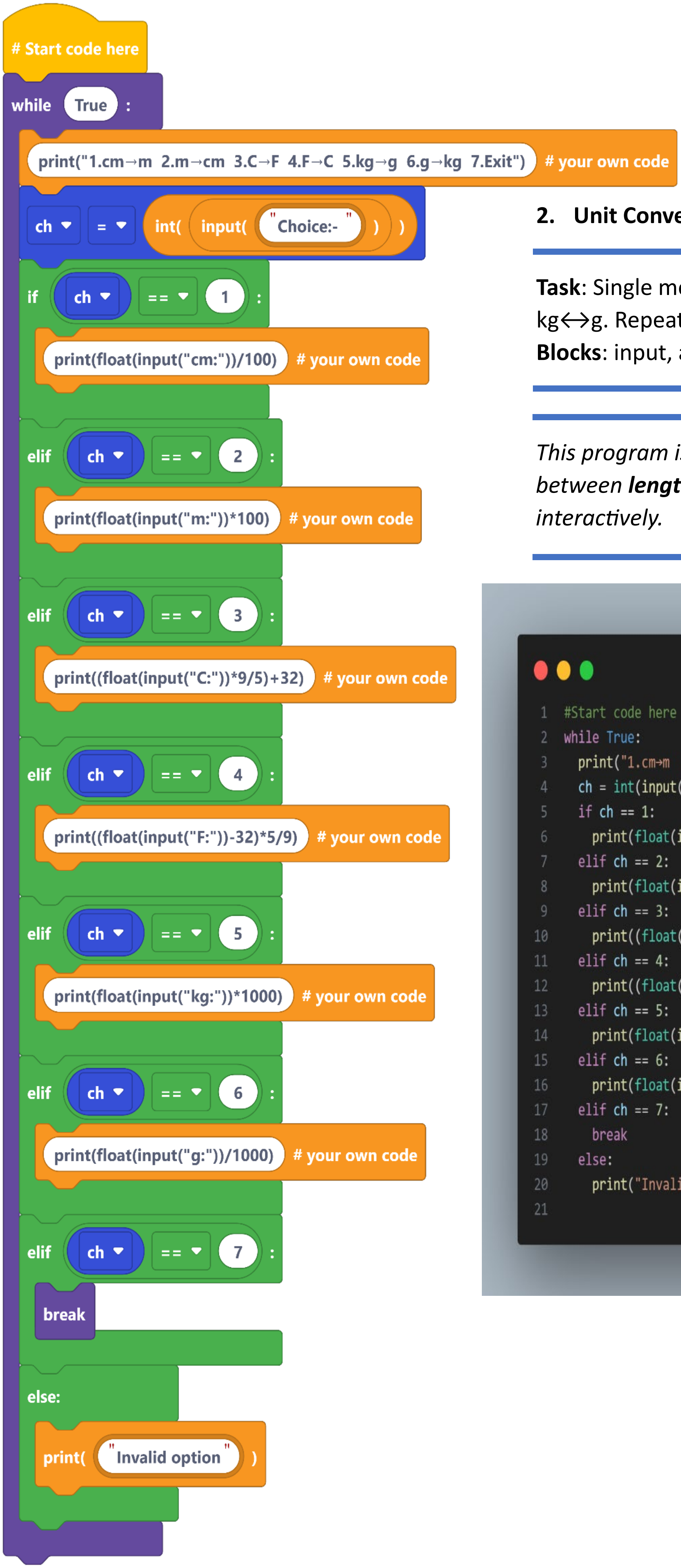


```
1.+  2.-  3.*  4./  5.%  6.//  7.**  8.Exit
Choice:- 1
First :- 69
Second :58
= 127.0

1.+  2.-  3.*  4./  5.%  6.//  7.**  8.Exit
Choice:- 2
First :- 12
Second :10
= 2.0

1.+  2.-  3.*  4./  5.%  6.//  7.**  8.Exit
Choice:- 3
First :- 5
Second :4
= 20.0

1.+  2.-  3.*  4./  5.%  6.//  7.**  8.Exit
Choice:- 5
First :- 45
Second :4
= 1.0

1.+  2.-  3.*  4./  5.%  6.//  7.**  8.Exit
Choice:- 6
First :- 55
Second :11
= 5.0

1.+  2.-  3.*  4./  5.%  6.//  7.**  8.Exit
Choice:- 7
First :- 12
Second :21
= 4.60051199093697e+22

1.+  2.-  3.*  4./  5.%  6.//  7.**  8.Exit
Choice:- 8
PS C:\Users\kalpe\Desktop\Edu Blocks>
```

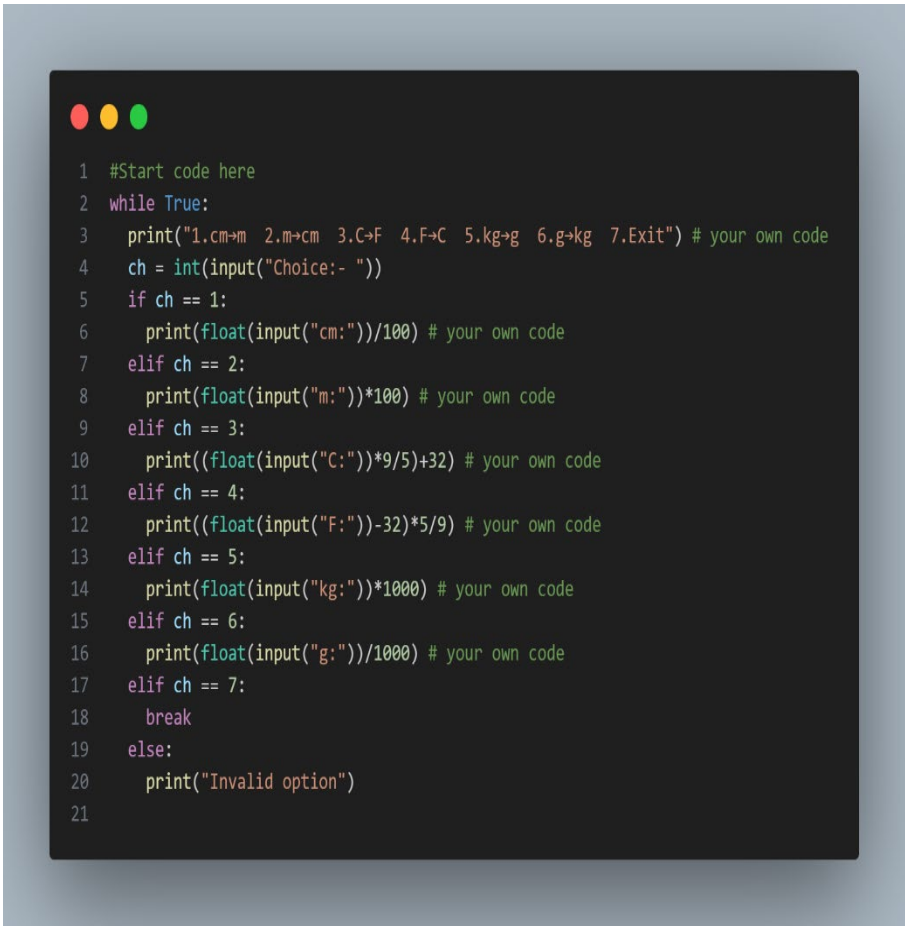
2. Unit Converter Hub

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

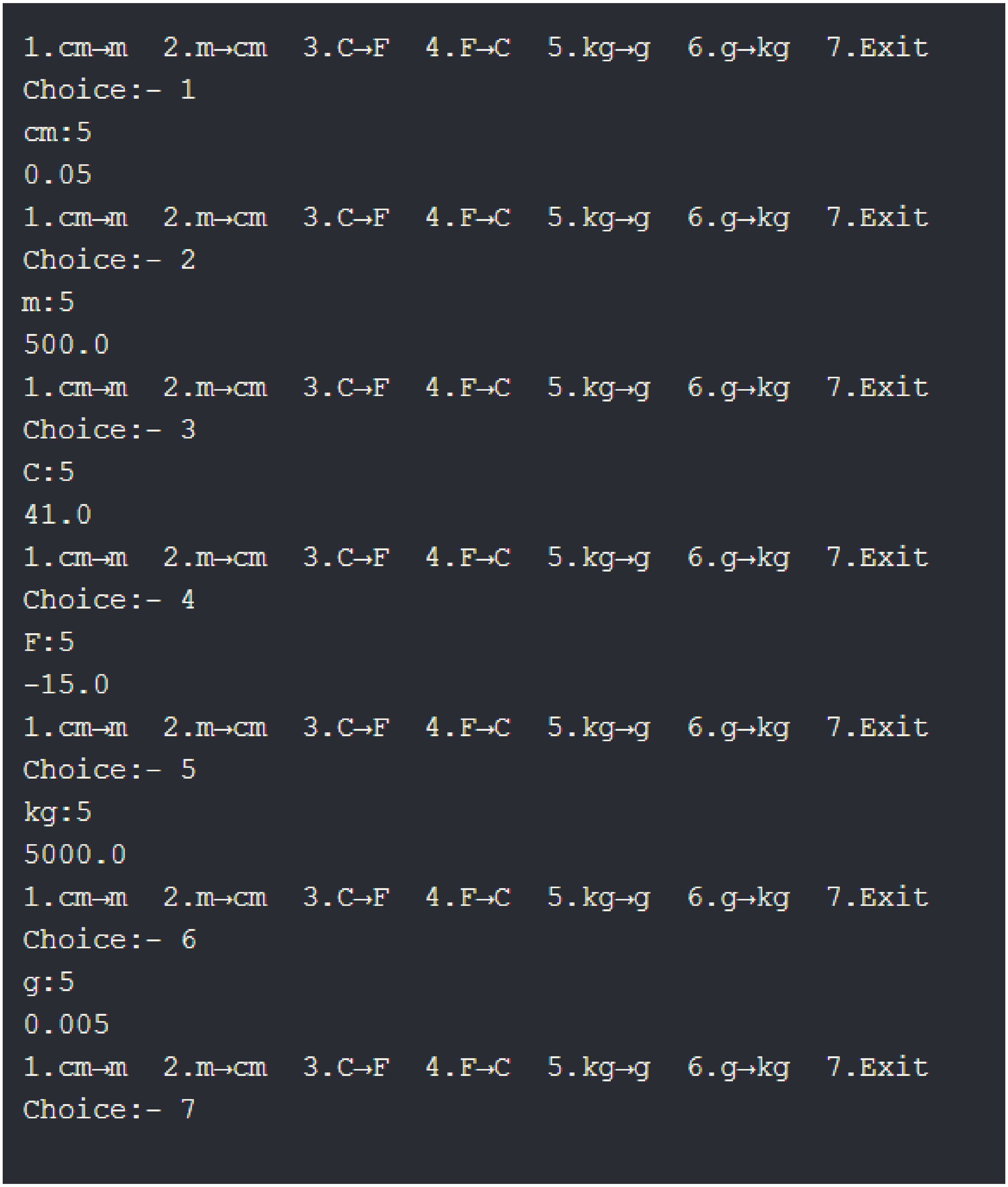
This program is a ***unit converter tool*** *that lets the user convert between* ***length (cm, m), temperature (C, F), and mass (kg, g)*** *interactively.*



```
# Start code here
while True :
    print("1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit") # your own code
    ch = int( input( "Choice:- " ) )
    if ch == 1 :
        print(float(input("cm:"))/100) # your own code
    elif ch == 2 :
        print(float(input("m:"))*100) # your own code
    elif ch == 3 :
        print((float(input("C:"))*9/5)+32) # your own code
    elif ch == 4 :
        print((float(input("F:"))-32)*5/9) # your own code
    elif ch == 5 :
        print(float(input("kg:"))*1000) # your own code
    elif ch == 6 :
        print(float(input("g:"))/1000) # your own code
    elif ch == 7 :
        break
    else:
        print( "Invalid option" )
```



```
#Start code here
while True:
  print("1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit") # your own code
  ch = int(input("Choice:- "))
  if ch == 1:
    print(float(input("cm:"))/100) # your own code
  elif ch == 2:
    print(float(input("m:"))*100) # your own code
  elif ch == 3:
    print((float(input("C:"))*9/5)+32) # your own code
  elif ch == 4:
    print((float(input("F:"))-32)*5/9) # your own code
  elif ch == 5:
    print(float(input("kg:"))*1000) # your own code
  elif ch == 6:
    print(float(input("g:"))/1000) # your own code
  elif ch == 7:
    break
  else:
    print("Invalid option")
```



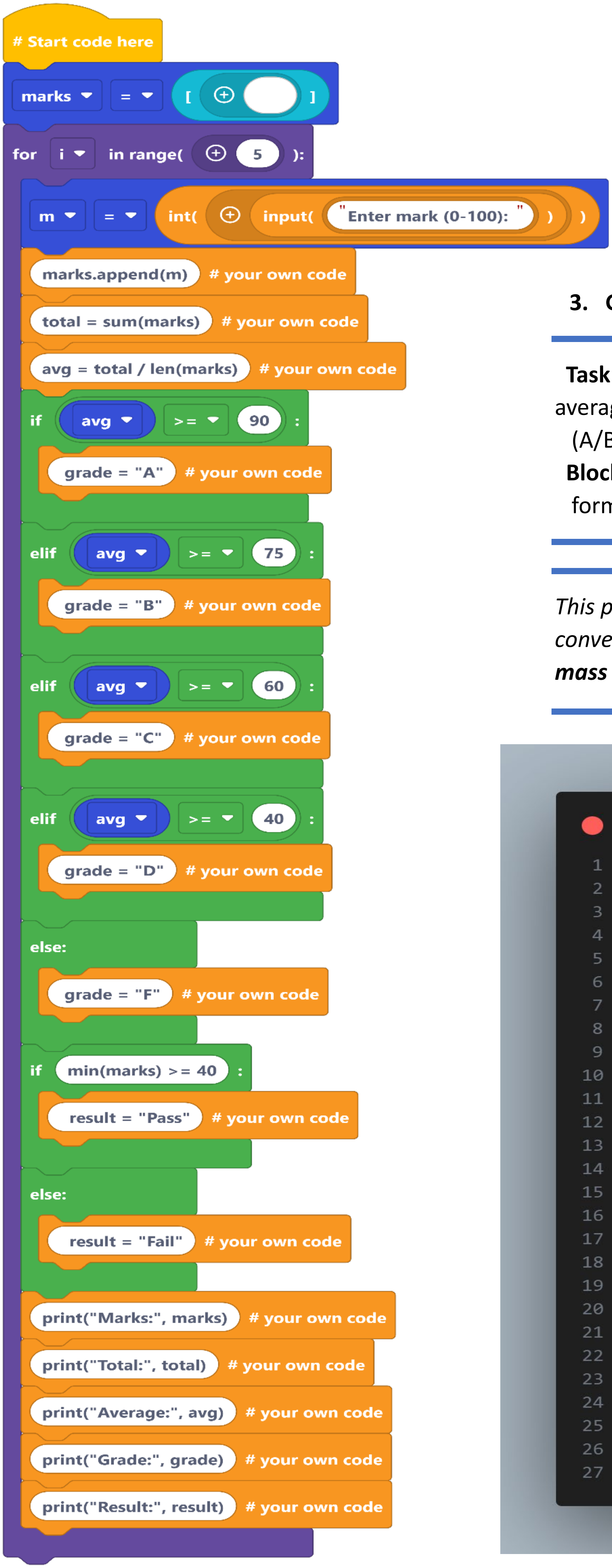
```
1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit
Choice:- 1
cm:5
0.05
1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit
Choice:- 2
m:5
500.0
1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit
Choice:- 3
C:5
41.0
1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit
Choice:- 4
F:5
-15.0
1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit
Choice:- 5
kg:5
5000.0
1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit
Choice:- 6
g:5
0.005
1.cm→m  2.m→cm  3.C→F  4.F→C  5.kg→g  6.g→kg  7.Exit
Choice:- 7
```

3. Grading & Result Sheet

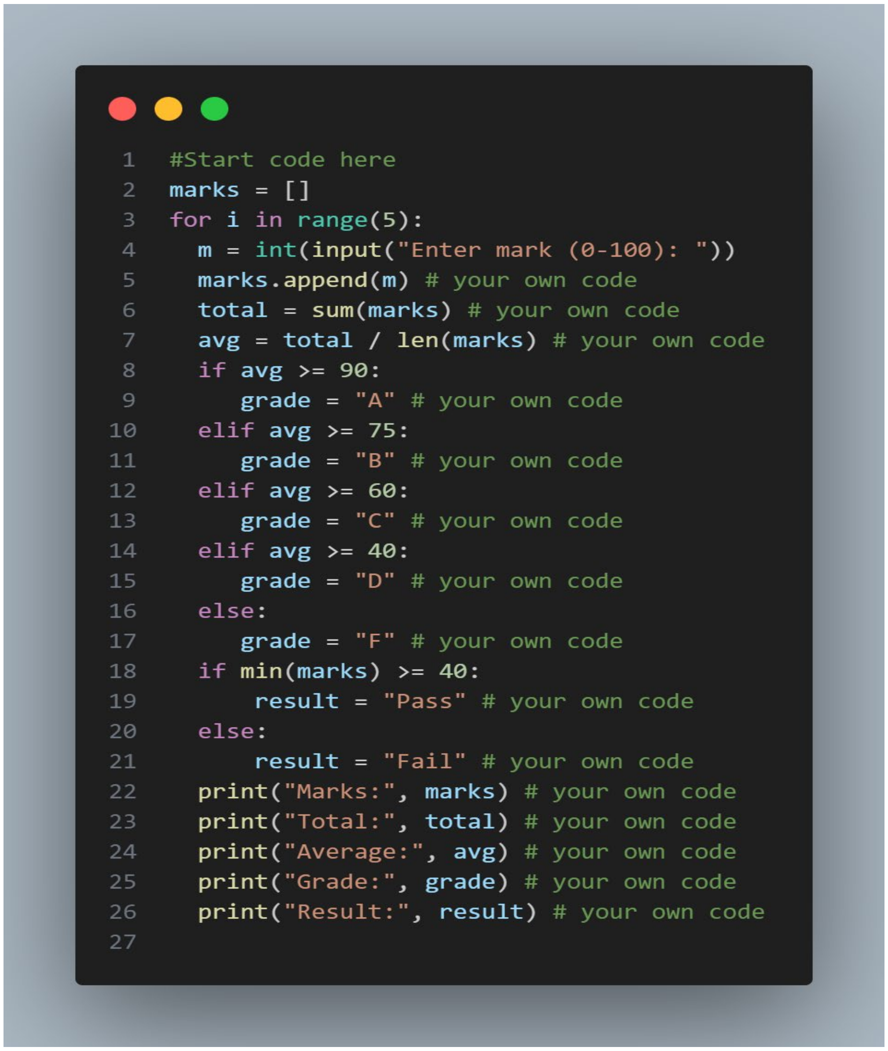
Task: Take 5 subject marks (0–100). Compute total, average, grade (A/B/C/D/F), and pass/fail.

Blocks: lists (append, len, sum), comparisons, if/elif, print formatting.

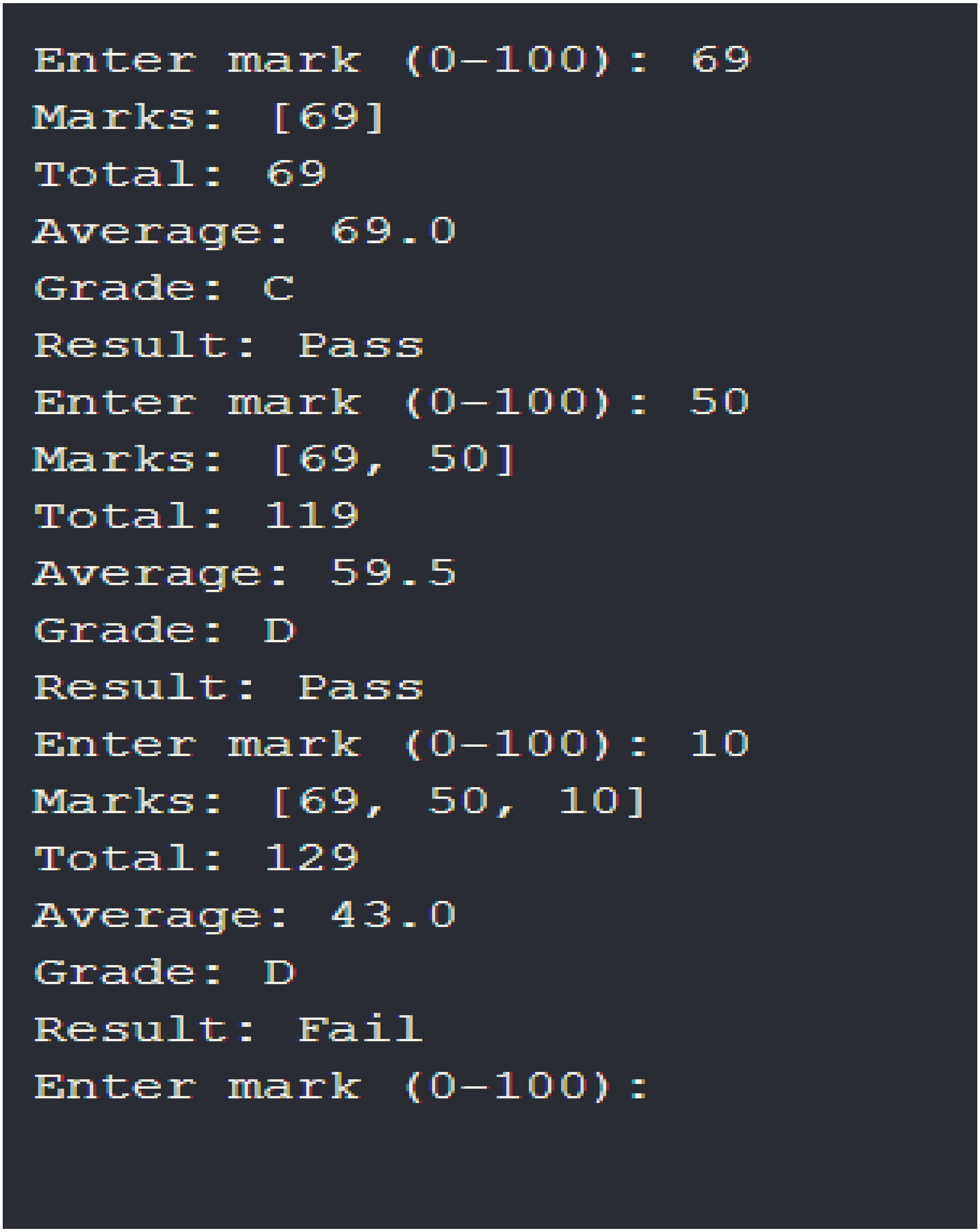
This program is a ***unit converter tool*** *that lets the user convert between* ***length (cm, m), temperature (C, F), and mass (kg, g)*** *interactively.*



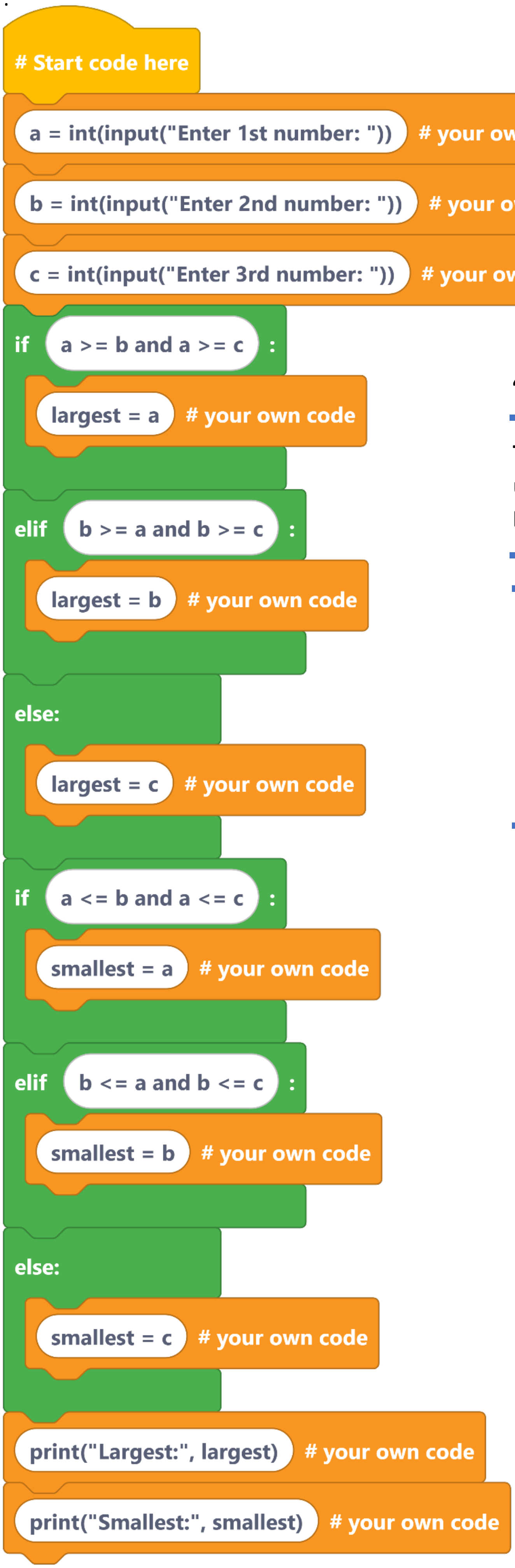
```
# Start code here
marks = [ ]
for i in range( 5 ):
    m = int( input( "Enter mark (0-100): " ) )
    marks.append(m) # your own code
    total = sum(marks) # your own code
    avg = total / len(marks) # your own code
    if avg >= 90 :
        grade = "A" # your own code
    elif avg >= 75 :
        grade = "B" # your own code
    elif avg >= 60 :
        grade = "C" # your own code
    elif avg >= 40 :
        grade = "D" # your own code
    else:
        grade = "F" # your own code
    if min(marks) >= 40 :
        result = "Pass" # your own code
    else:
        result = "Fail" # your own code
    print("Marks:", marks) # your own code
    print("Total:", total) # your own code
    print("Average:", avg) # your own code
    print("Grade:", grade) # your own code
    print("Result:", result) # your own code
```



```
#Start code here
marks = []
for i in range(5):
  m = int(input("Enter mark (0-100): "))
  marks.append(m) # your own code
  total = sum(marks) # your own code
  avg = total / len(marks) # your own code
  if avg >= 90:
     grade = "A" # your own code
  elif avg >= 75:
     grade = "B" # your own code
  elif avg >= 60:
     grade = "C" # your own code
  elif avg >= 40:
     grade = "D" # your own code
  else:
     grade = "F" # your own code
  if min(marks) >= 40:
      result = "Pass" # your own code
  else:
      result = "Fail" # your own code
  print("Marks:", marks) # your own code
  print("Total:", total) # your own code
  print("Average:", avg) # your own code
  print("Grade:", grade) # your own code
  print("Result:", result) # your own code
```



```
Enter mark (0-100): 69
Marks: [69]
Total: 69
Average: 69.0
Grade: C
Result: Pass
Enter mark (0-100): 50
Marks: [69, 50]
Total: 119
Average: 59.5
Grade: D
Result: Pass
Enter mark (0-100): 10
Marks: [69, 50, 10]
Total: 129
Average: 43.0
Grade: D
Result: Fail
Enter mark (0-100):
```

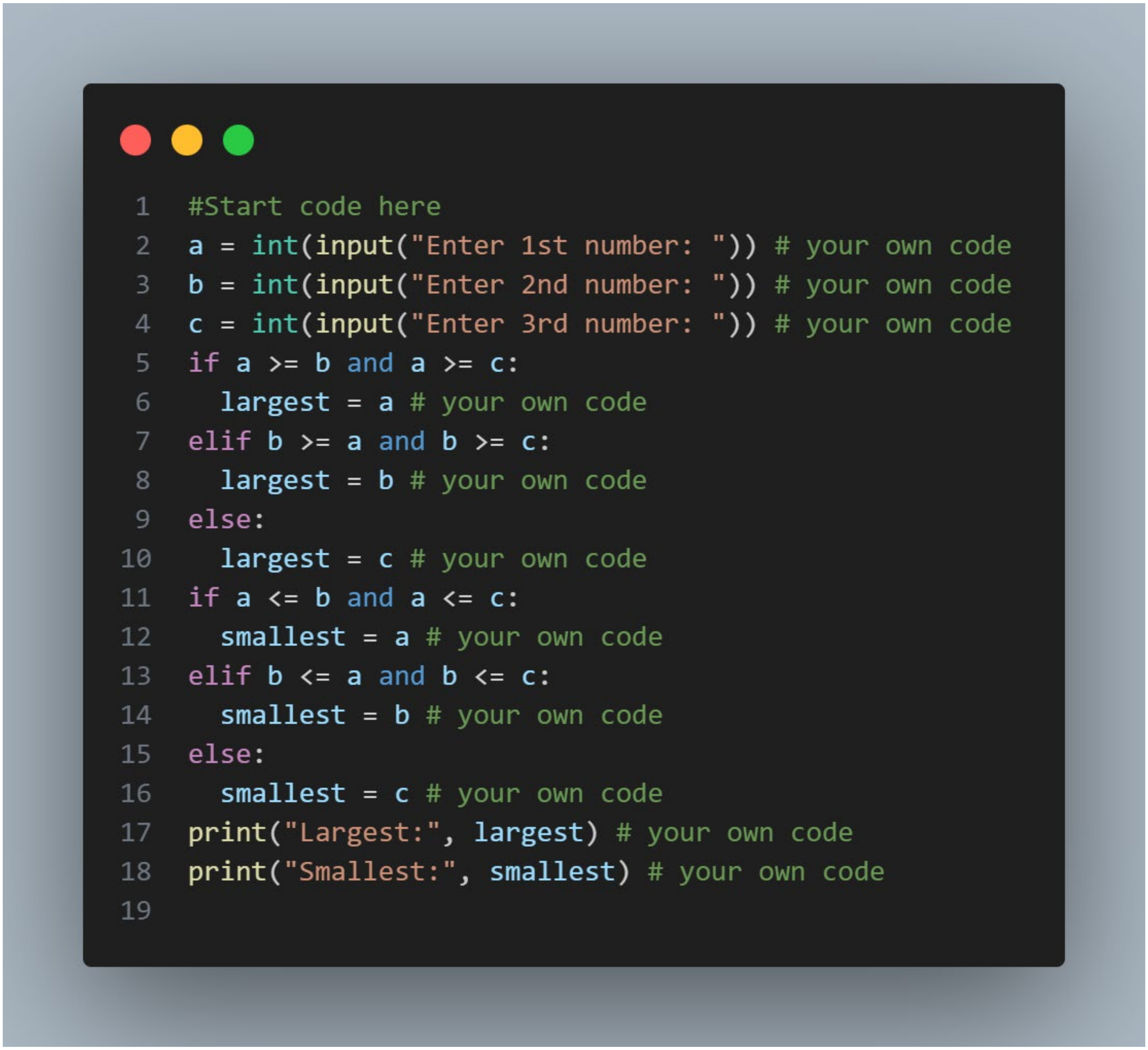


```
# Start code here
a = int(input("Enter 1st number: ")) # your own code
b = int(input("Enter 2nd number: ")) # your own code
c = int(input("Enter 3rd number: ")) # your own code
if a >= b and a >= c :
    largest = a # your own code
elif b >= a and b >= c :
    largest = b # your own code
else:
    largest = c # your own code
if a <= b and a <= c :
    smallest = a # your own code
elif b <= a and b <= c :
    smallest = b # your own code
else:
    smallest = c # your own code
print("Largest:", largest) # your own code
print("Smallest:", smallest) # your own code
```

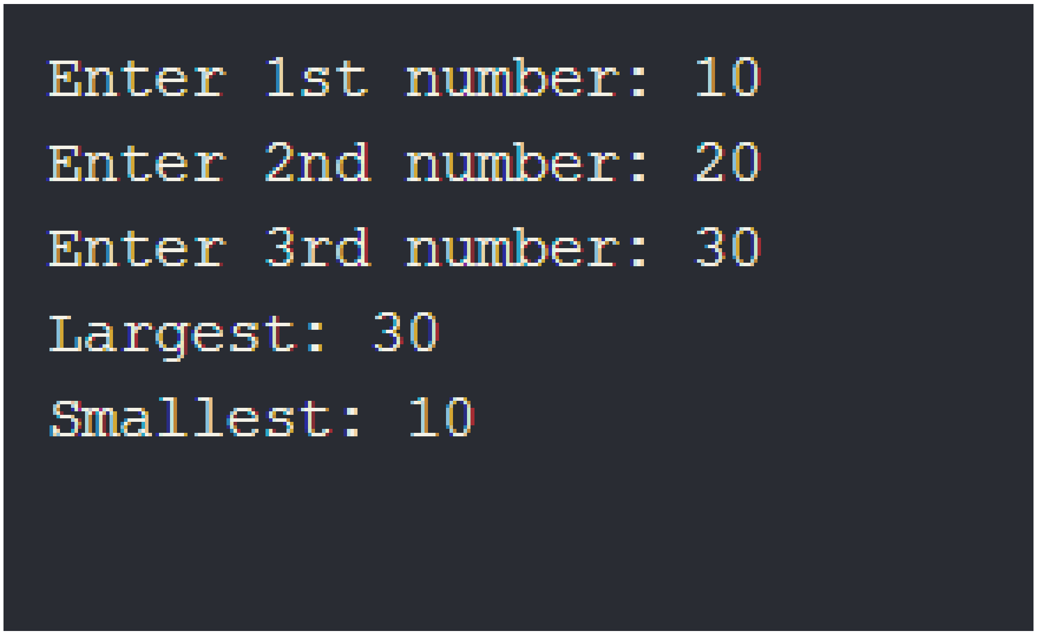
4. Max–Min of Three

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

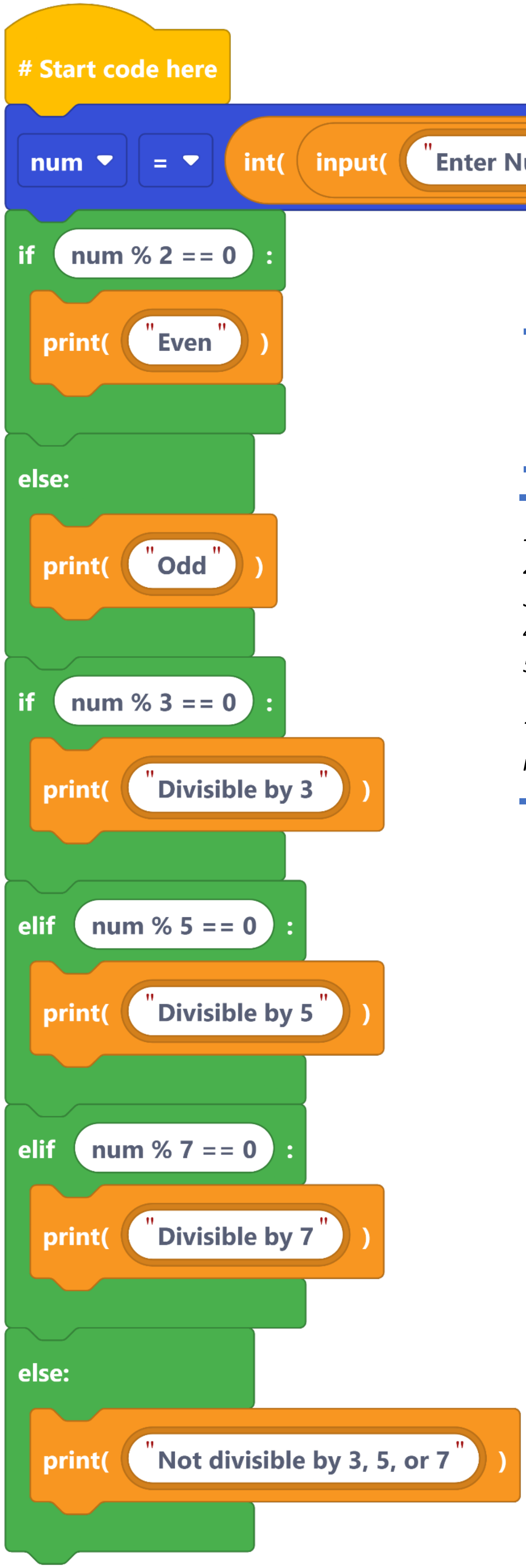
1. *Takes three numbers as input (a, b, c).*
2. *Uses if-elif-else conditions to check:*
3. *Which number is the **largest**.Which number is the **smallest**.*
4. *Prints both the largest and smallest numbers*
5. ***The program finds and displays the largest and smallest of three numbers entered by the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



```python
#Start code here
a = int(input("Enter 1st number: ")) # your own code
b = int(input("Enter 2nd number: ")) # your own code
c = int(input("Enter 3rd number: ")) # your own code
if a >= b and a >= c:
  largest = a # your own code
elif b >= a and b >= c:
  largest = b # your own code
else:
  largest = c # your own code
if a <= b and a <= c:
  smallest = a # your own code
elif b <= a and b <= c:
  smallest = b # your own code
else:
  smallest = c # your own code
print("Largest:", largest) # your own code
print("Smallest:", smallest) # your own code

```



```
Enter 1st number: 10
Enter 2nd number: 20
Enter 3rd number: 30
Largest: 30
Smallest: 10
```



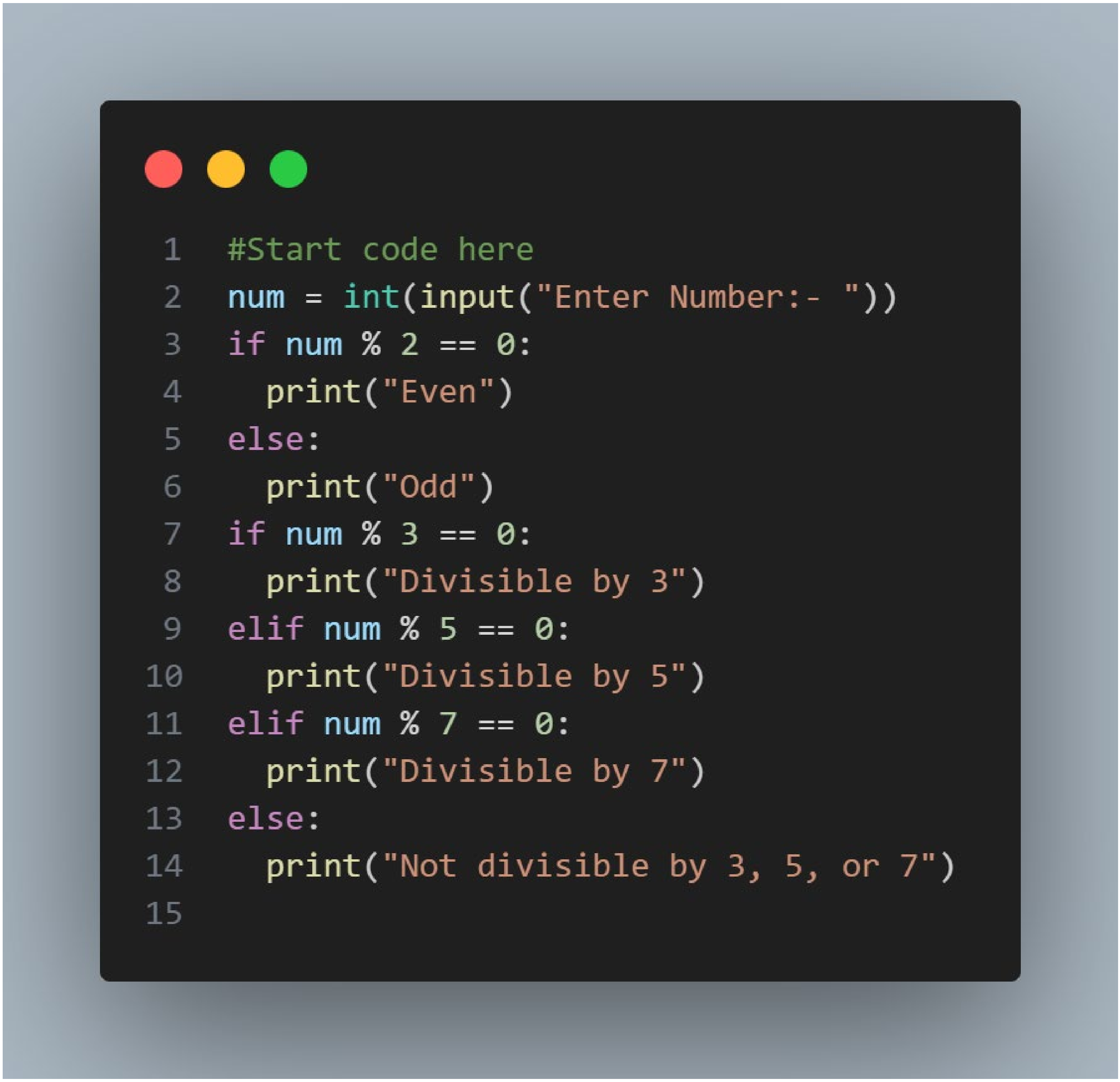
```
# Start code here
num = int( input( "Enter Number:-" ) )
if num % 2 == 0 :
    print( "Even" )
else:
    print( "Odd" )
if num % 3 == 0 :
    print( "Divisible by 3" )
elif num % 5 == 0 :
    print( "Divisible by 5" )
elif num % 7 == 0 :
    print( "Divisible by 7" )
else:
    print( "Not divisible by 3, 5, or 7" )
```

5. Even/Odd & Divisibility Checker

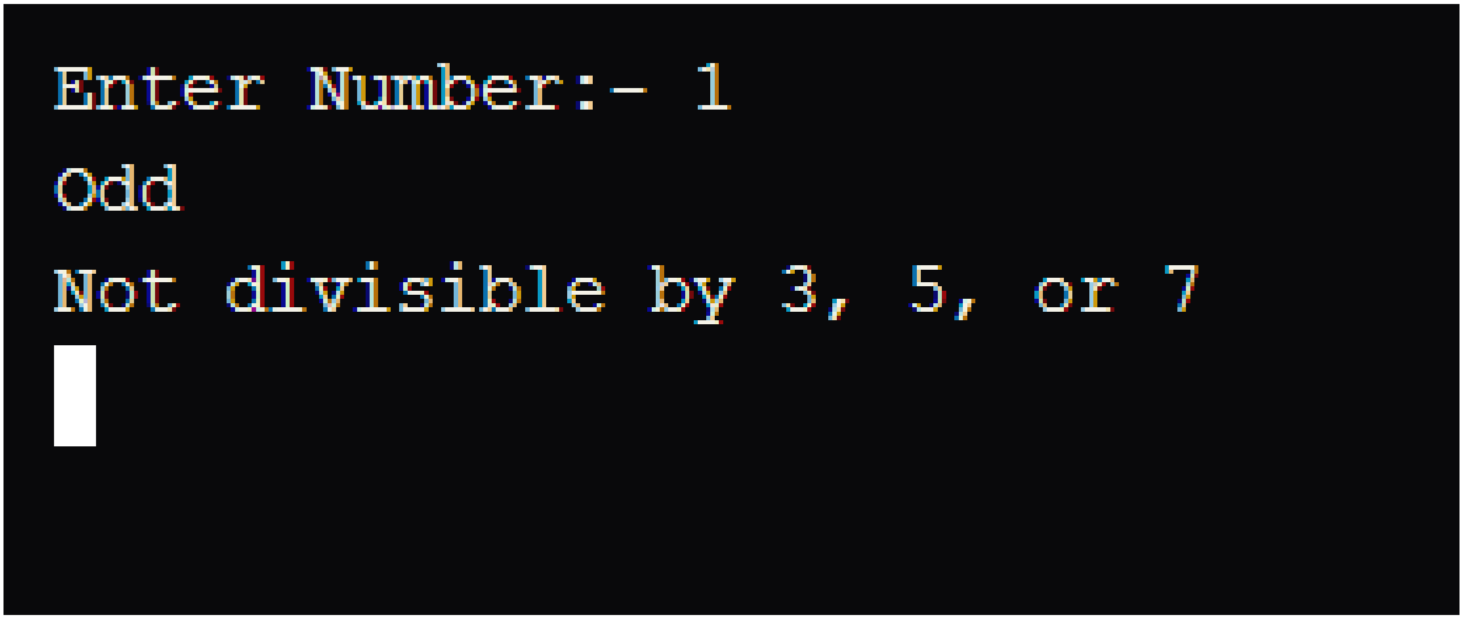
Task: Read a number; report even/odd, and divisibility by 3, 5, and 7.
Blocks: modulus %, chained if/elif, logical operators.

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****.*



```python
#Start code here
num = int(input("Enter Number:- "))
if num % 2 == 0:
  print("Even")
else:
  print("Odd")
if num % 3 == 0:
  print("Divisible by 3")
elif num % 5 == 0:
  print("Divisible by 5")
elif num % 7 == 0:
  print("Divisible by 7")
else:
  print("Not divisible by 3, 5, or 7")

```



```
Enter Number:- 1
Odd
Not divisible by 3, 5, or 7
```

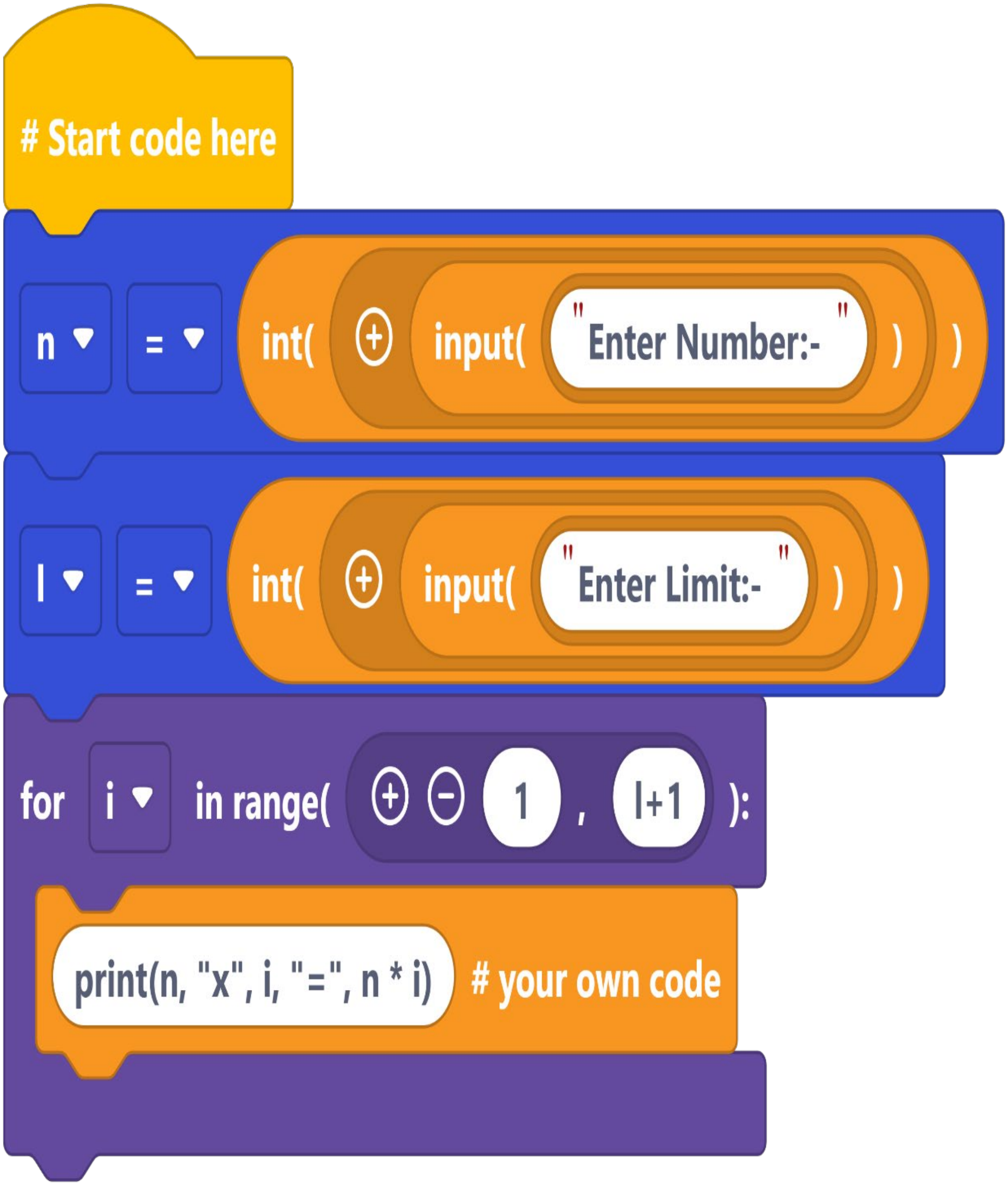
6. Times Table Generator

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

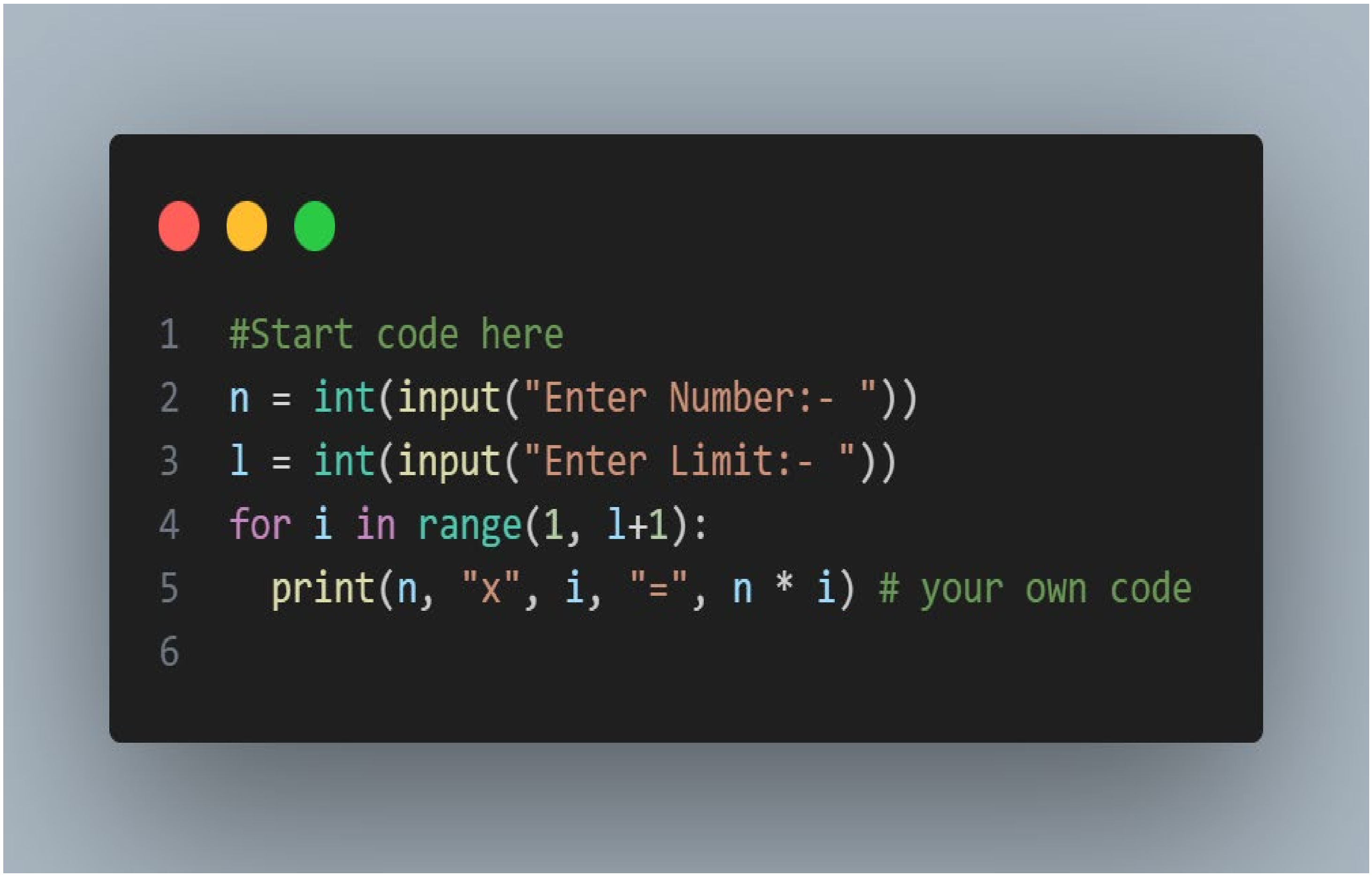
1. *Asks the user for a number n.*
2. *Asks the user for a limit l.*
3. *Uses a loop to print the multiplication table of n from 1 to l.*

☞ ***In short (lab manual summary):***

This program prints the ***multiplication table*** *of a given number up to a user-defined limit.*

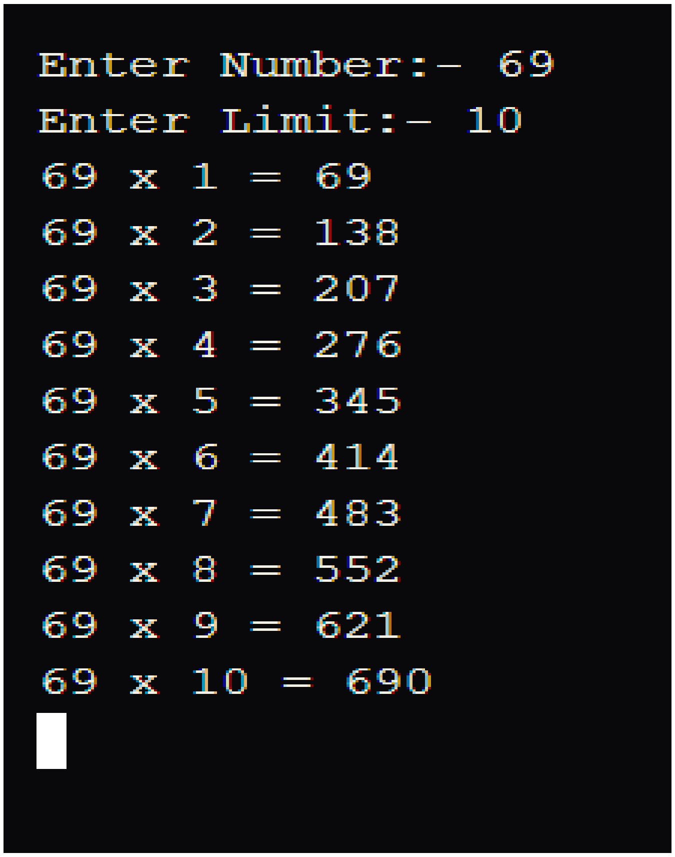


```
# Start code here
n = int( input( "Enter Number:-" ) )
l = int( input( "Enter Limit:-" ) )
for i in range( 1 , l+1 ):
    print(n, "x", i, "=", n * i) # your own code
```

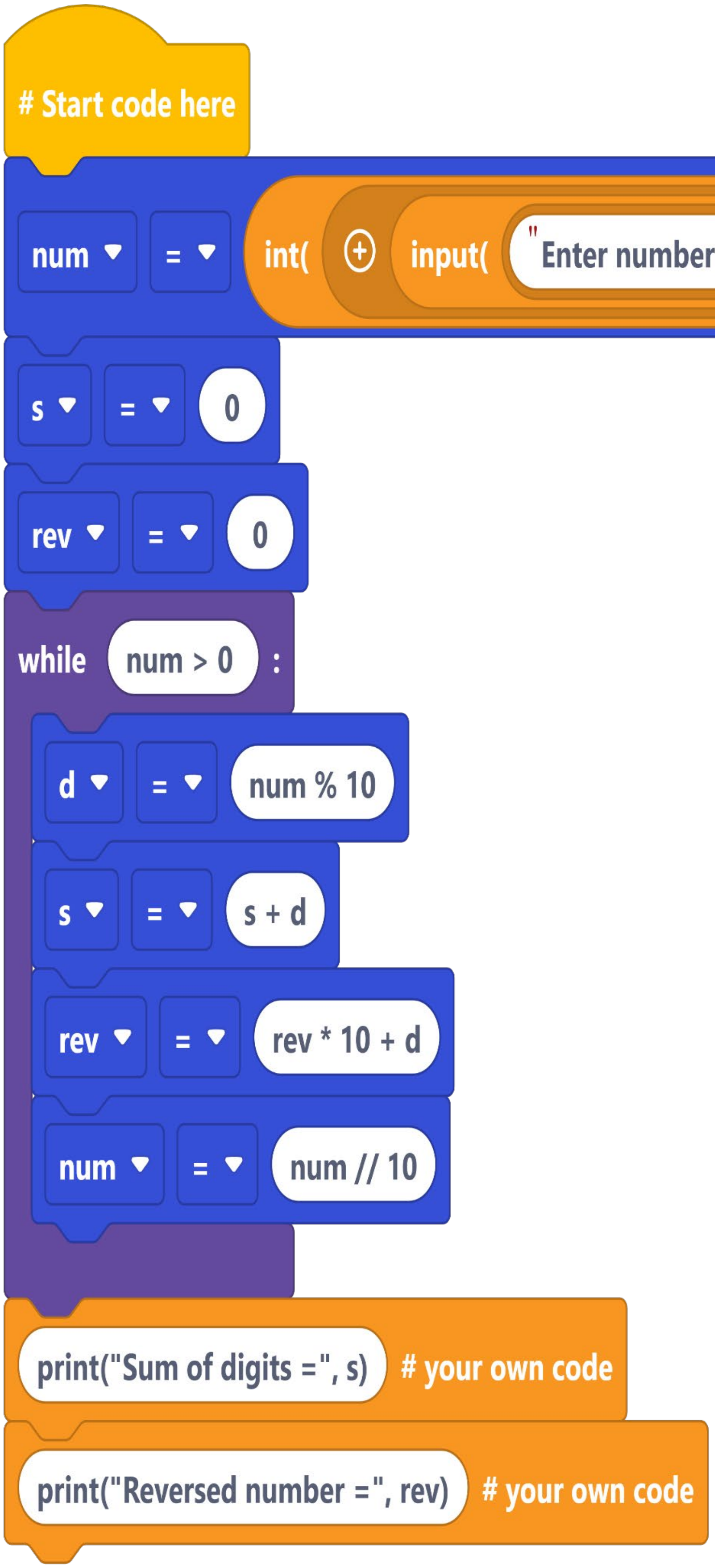


```python
#Start code here
n = int(input("Enter Number:- "))
l = int(input("Enter Limit:- "))
for i in range(1, l+1):
  print(n, "x", i, "=", n * i) # your own code

```

.



```
Enter Number:- 69
Enter Limit:- 10
69 x 1 = 69
69 x 2 = 138
69 x 3 = 207
69 x 4 = 276
69 x 5 = 345
69 x 6 = 414
69 x 7 = 483
69 x 8 = 552
69 x 9 = 621
69 x 10 = 690
```

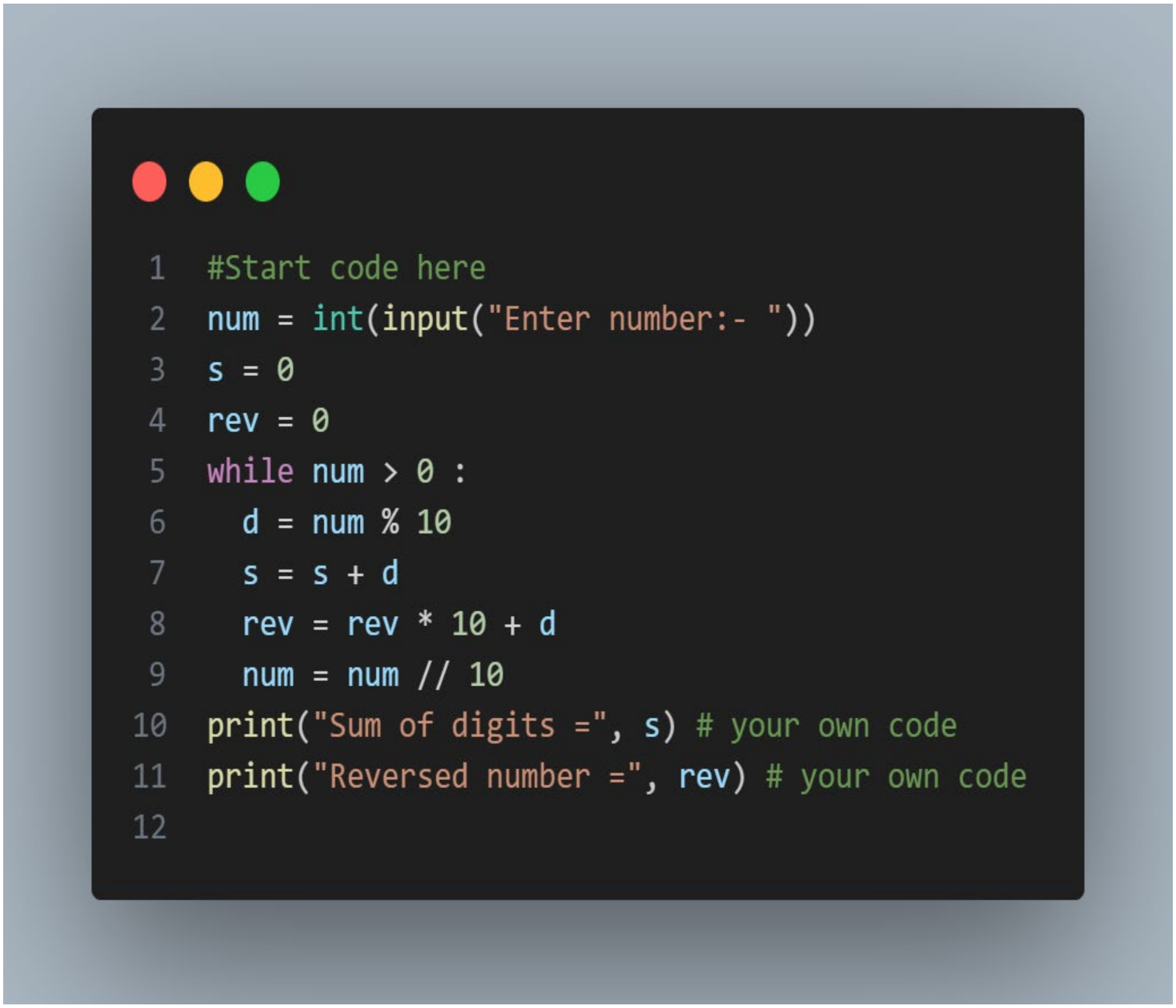


```
# Start code here
num = int( input( "Enter number:-" ) )
s = 0
rev = 0
while num > 0 :
    d = num % 10
    s = s + d
    rev = rev * 10 + d
    num = num // 10
print("Sum of digits =", s) # your own code
print("Reversed number =", rev) # your own code
```

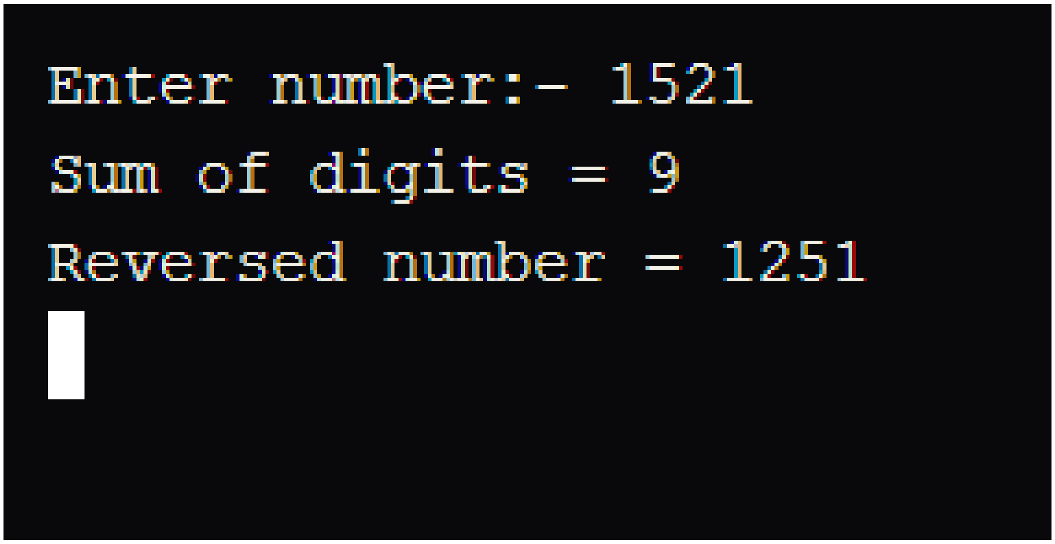
7. Sum of Digits & Reverse Number

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

1. *Takes a number 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**.*



```python
#Start code here
num = int(input("Enter number:- "))
s = 0
rev = 0
while num > 0 :
  d = num % 10
  s = s + d
  rev = rev * 10 + d
  num = num // 10
print("Sum of digits =", s) # your own code
print("Reversed number =", rev) # your own code
```

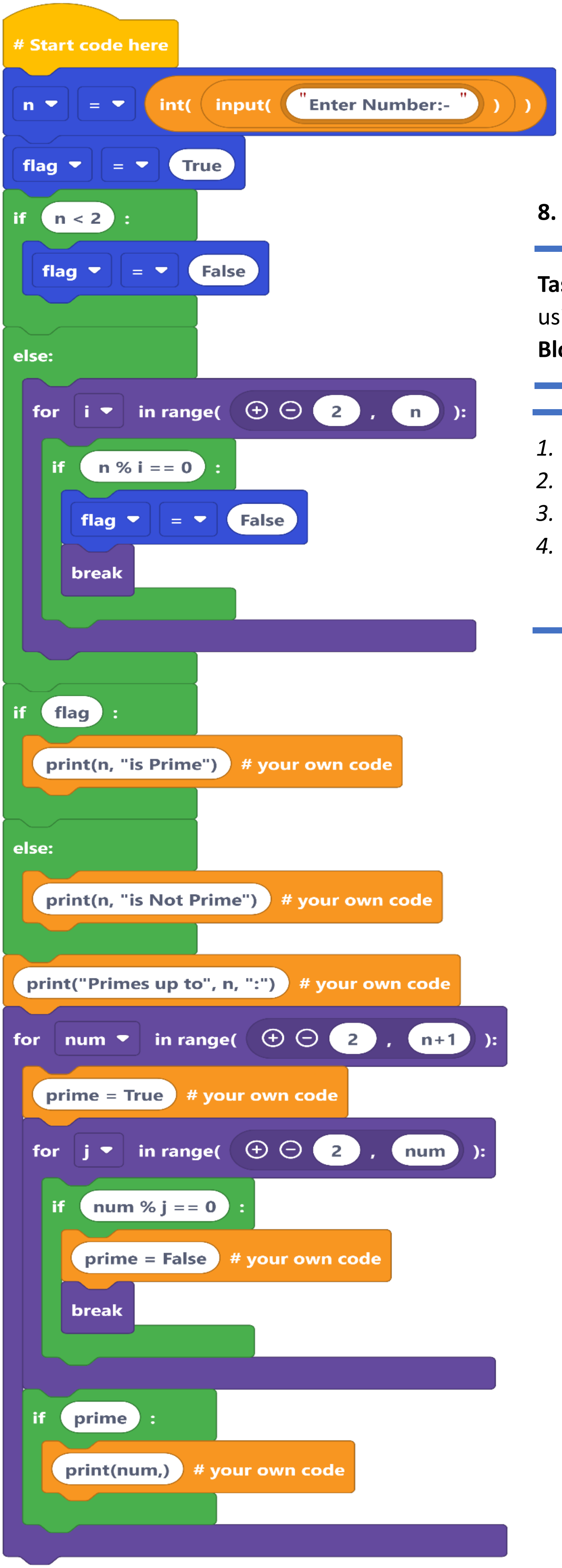


```
Enter number:- 1521
Sum of digits = 9
Reversed number = 1251
```

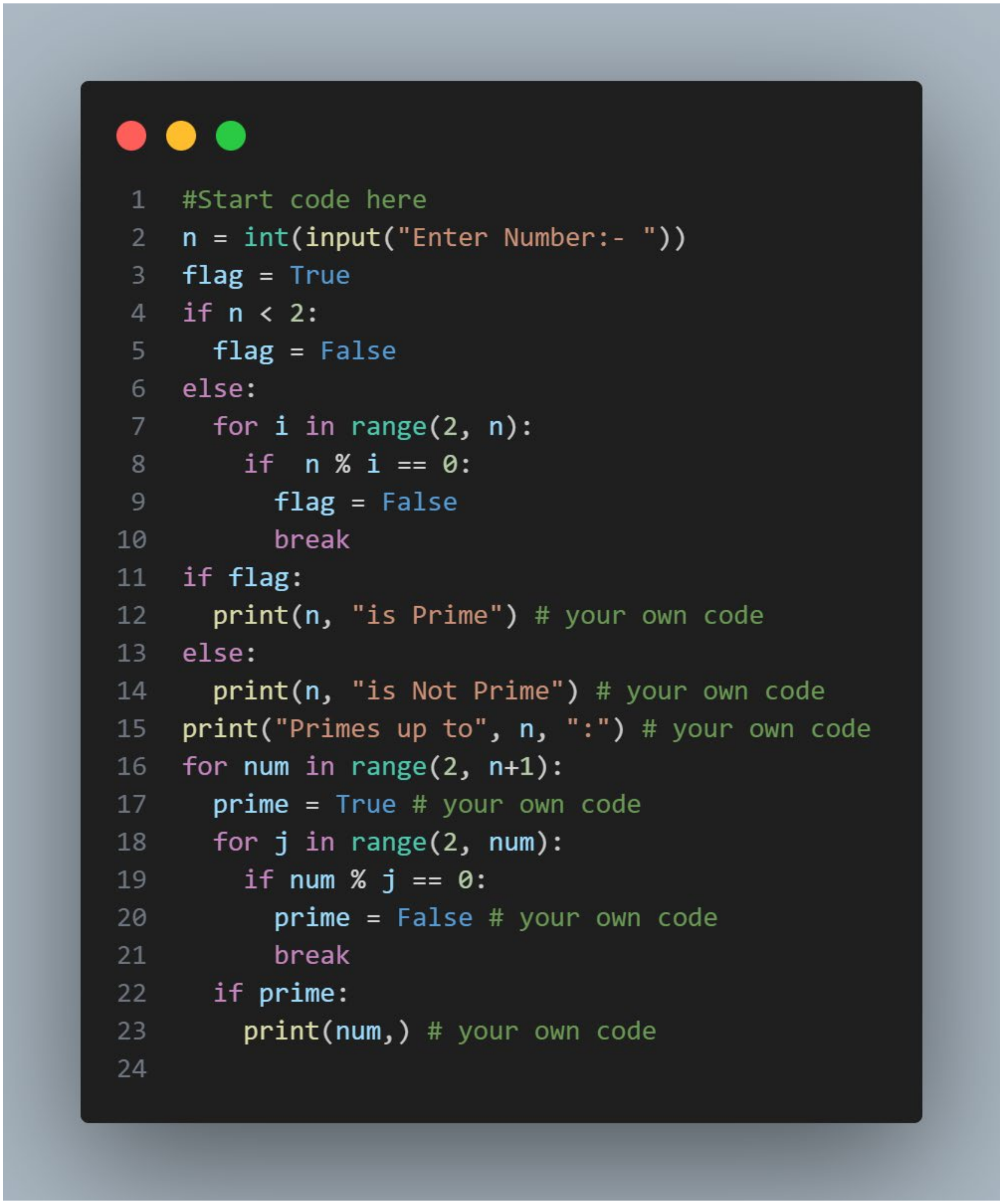
8. Prime Checker + List Primes

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.

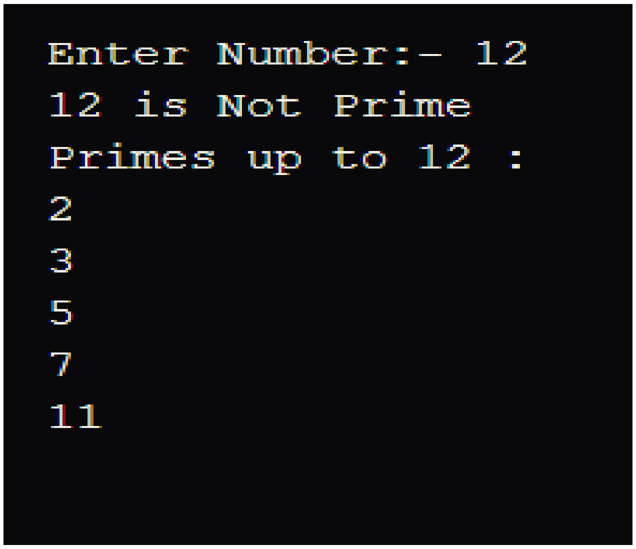
1. *Takes a number n as input.*
2. *Checks whether n is **Prime or Not Prime**.*
3. *Prints all **prime numbers up to n**.*
4. *This program checks if a given number is **prime** and also displays all **prime numbers up to that number**.*



```
# Start code here
n = int( input( "Enter Number:- " ) )
flag = True
if n < 2 :
    flag = False
else:
    for i in range( 2 , n ):
        if n % i == 0 :
            flag = False
            break
if flag :
    print(n, "is Prime") # your own code
else:
    print(n, "is Not Prime") # your own code
print("Primes up to", n, ":") # your own code
for num in range( 2 , n+1 ):
    prime = True # your own code
    for j in range( 2 , num ):
        if num % j == 0 :
            prime = False # your own code
            break
    if prime :
        print(num,) # your own code
```



```python
#Start code here
n = int(input("Enter Number:- "))
flag = True
if n < 2:
  flag = False
else:
  for i in range(2, n):
    if  n % i == 0:
      flag = False
      break
if flag:
  print(n, "is Prime") # your own code
else:
  print(n, "is Not Prime") # your own code
print("Primes up to", n, ":") # your own code
for num in range(2, n+1):
  prime = True # your own code
  for j in range(2, num):
    if num % j == 0:
      prime = False # your own code
      break
  if prime:
    print(num,) # your own code
```

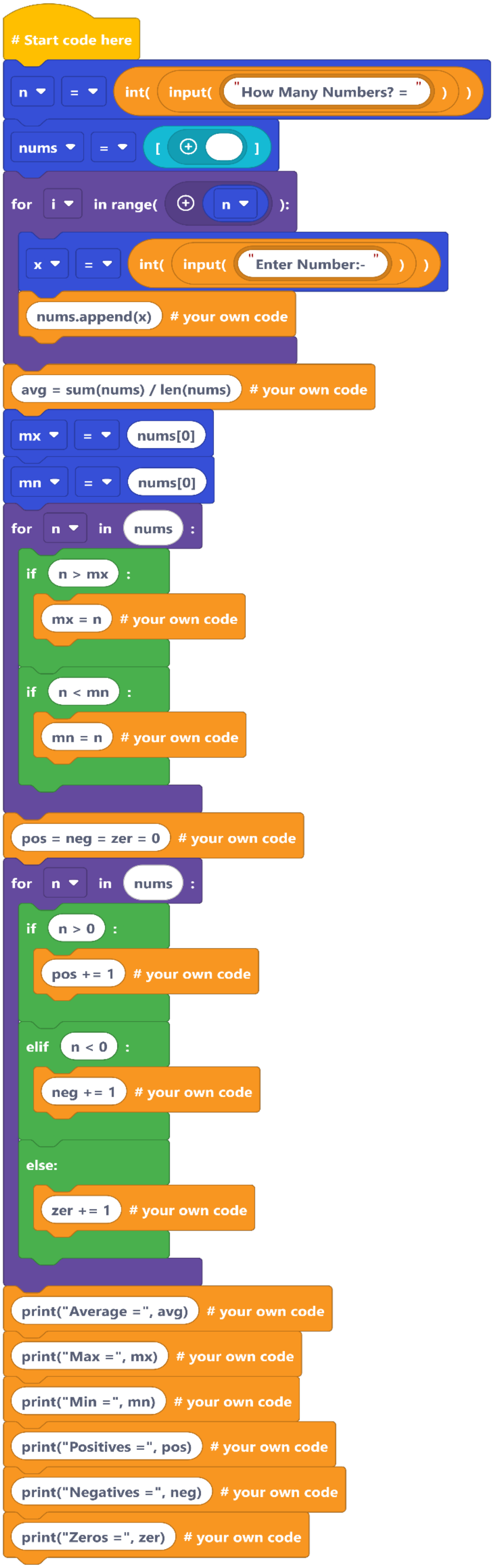


```
Enter Number:- 12
12 is Not Prime
Primes up to 12 :
2
3
5
7
11
```

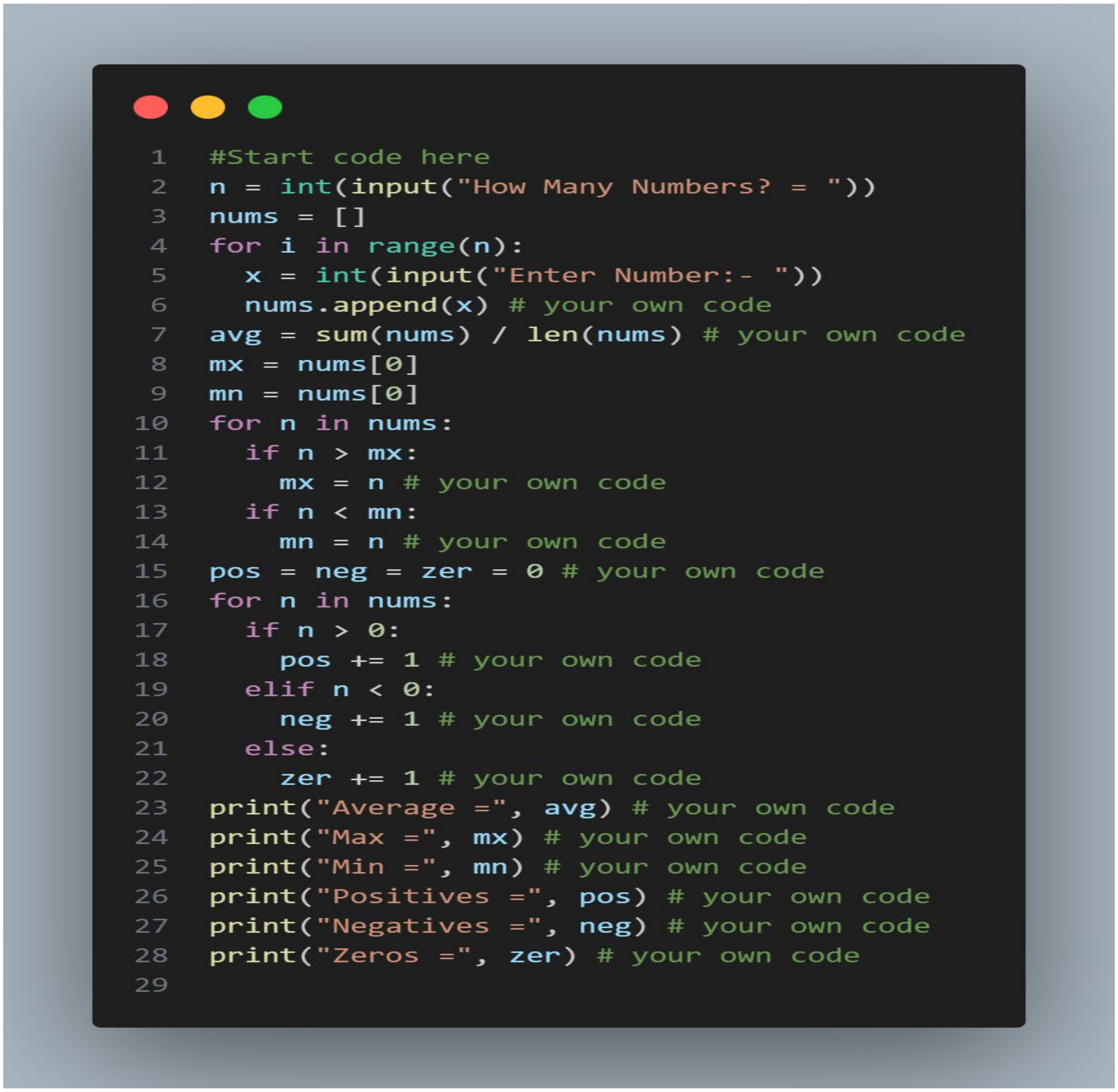
9. Number List Analyzer

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

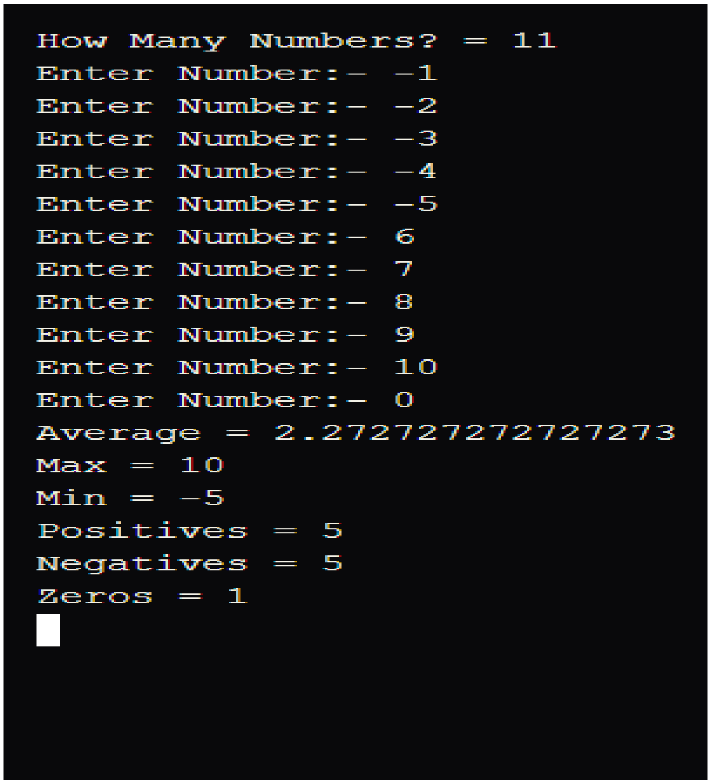
1. *Asks the user how many numbers they want to enter.*
2. *Stores all numbers in a list.*
3. *Calculates:*
4. ***Average** of numbers*
5. ***Maximum** number*
6. ***Minimum** number*
7. *Counts how many numbers are **positive, negative, and zeros**.*
8. *Displays all results.*
9. *This program finds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



```
# Start code here
n = int( input( "How Many Numbers? = " ) )
nums = [ ]
for i in range( n ):
    x = int( input( "Enter Number:-" ) )
    nums.append(x) # your own code
avg = sum(nums) / len(nums) # your own code
mx = nums[0]
mn = nums[0]
for n in nums :
    if n > mx :
        mx = n # your own code
    if n < mn :
        mn = n # your own code
pos = neg = zer = 0 # your own code
for n in nums :
    if n > 0 :
        pos += 1 # your own code
    elif n < 0 :
        neg += 1 # your own code
    else:
        zer += 1 # your own code
print("Average =", avg) # your own code
print("Max =", mx) # your own code
print("Min =", mn) # your own code
print("Positives =", pos) # your own code
print("Negatives =", neg) # your own code
print("Zeros =", zer) # your own code
```



```python
#Start code here
n = int(input("How Many Numbers? = "))
nums = []
for i in range(n):
  x = int(input("Enter Number:- "))
  nums.append(x) # your own code
avg = sum(nums) / len(nums) # your own code
mx = nums[0]
mn = nums[0]
for n in nums:
  if n > mx:
    mx = n # your own code
  if n < mn:
    mn = n # your own code
pos = neg = zer = 0 # your own code
for n in nums:
  if n > 0:
    pos += 1 # your own code
  elif n < 0:
    neg += 1 # your own code
  else:
    zer += 1 # your own code
print("Average =", avg) # your own code
print("Max =", mx) # your own code
print("Min =", mn) # your own code
print("Positives =", pos) # your own code
print("Negatives =", neg) # your own code
print("Zeros =", zer) # your own code

```

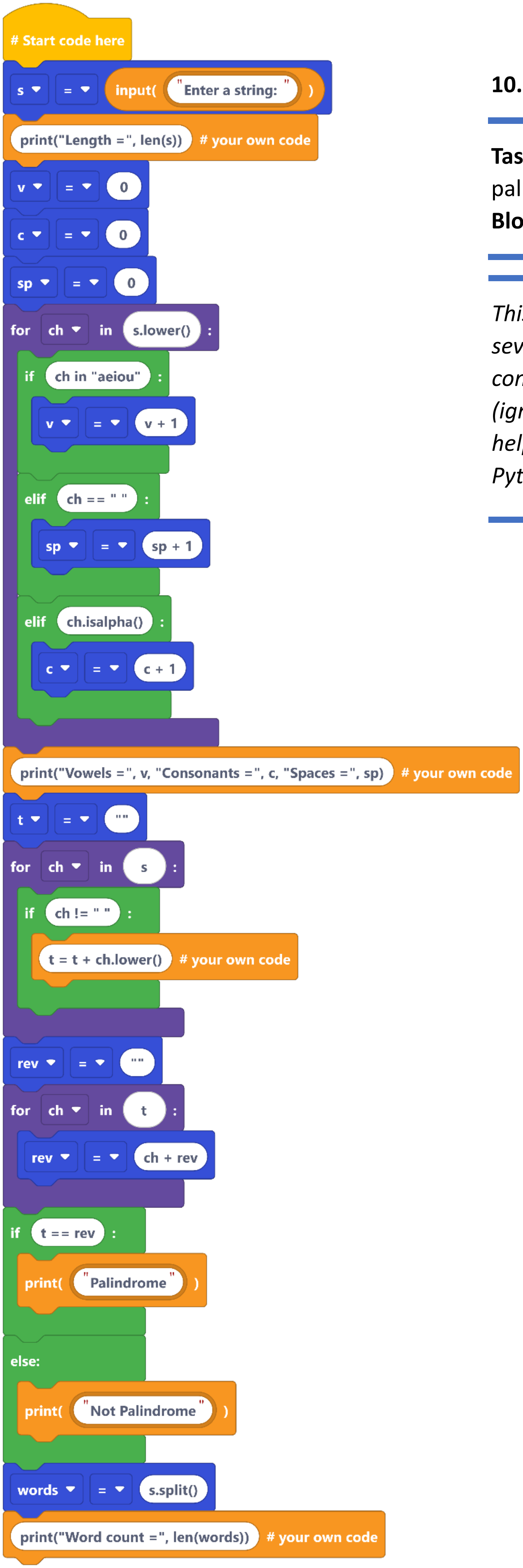


```
How Many Numbers? = 11
Enter Number:- -1
Enter Number:- -2
Enter Number:- -3
Enter Number:- -4
Enter Number:- -5
Enter Number:- 6
Enter Number:- 7
Enter Number:- 8
Enter Number:- 9
Enter Number:- 10
Enter Number:- 0
Average = 2.272727272727273
Max = 10
Min = -5
Positives = 5
Negatives = 5
Zeros = 1
```

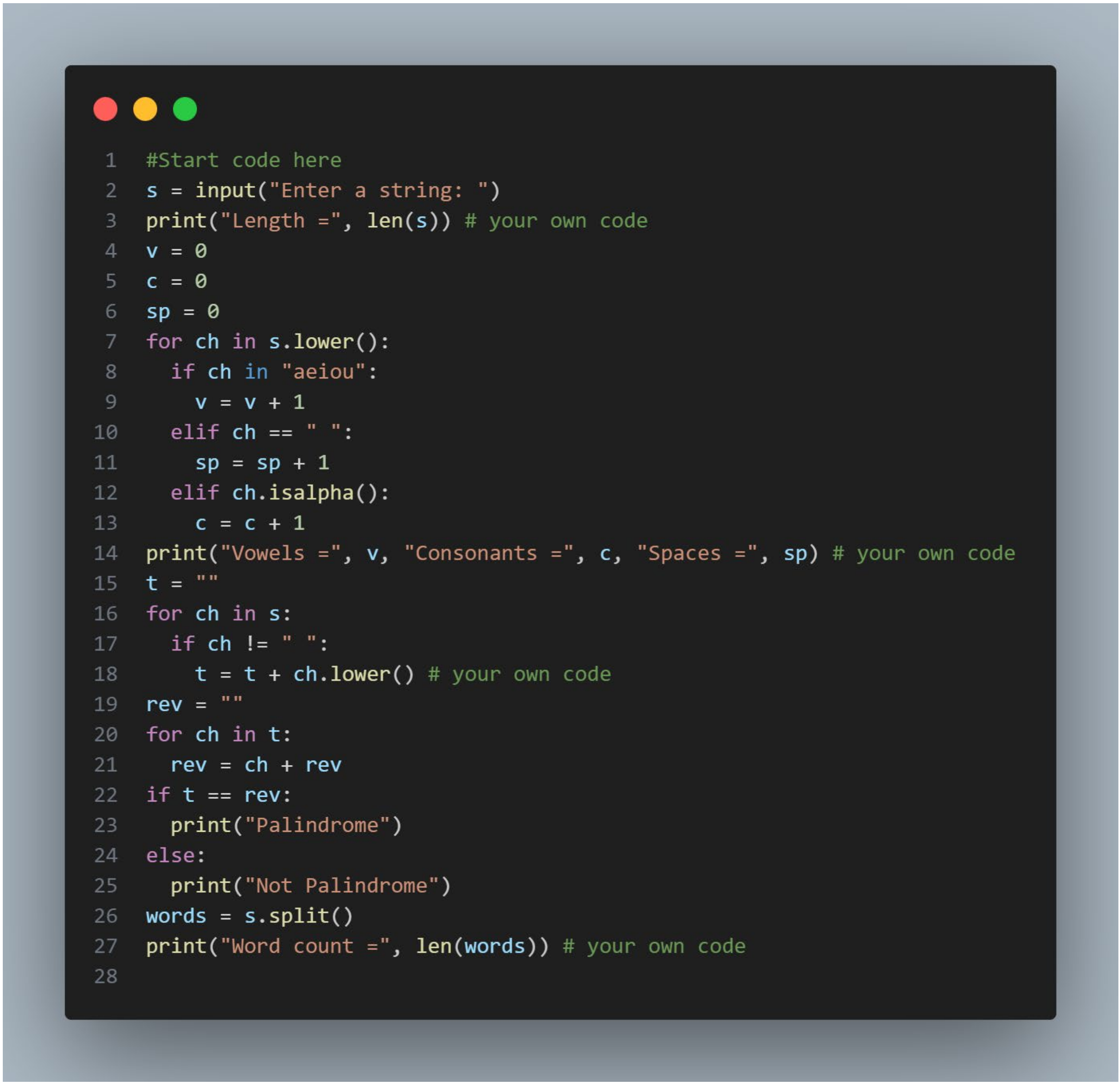
10. String Utility Box

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

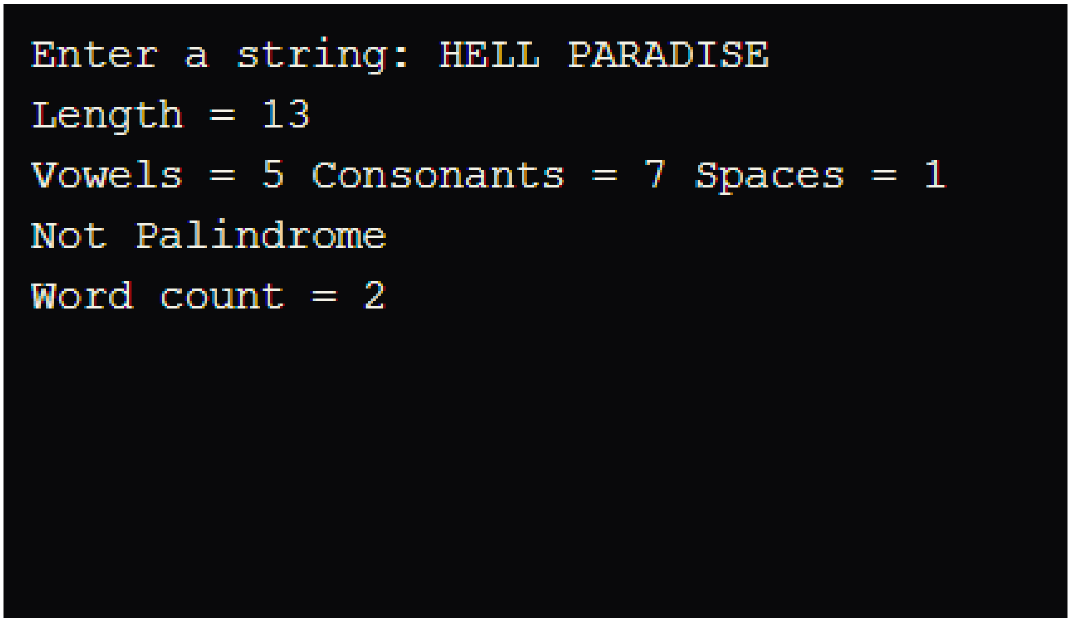
This Python program takes a user-input string and performs several operations: it calculates the string's length, counts vowels, consonants, and spaces, checks if the string is a palindrome (ignoring spaces and case), and counts the number of words. It helps demonstrate basic string handling, loops,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



```
# Start code here
s = input("Enter a string: ")
print("Length =", len(s)) # your own code
v = 0
c = 0
sp = 0
for ch in s.lower() :
    if ch in "aeiou" :
        v = v + 1
    elif ch == " " :
        sp = sp + 1
    elif ch.isalpha() :
        c = c + 1
print("Vowels =", v, "Consonants =", c, "Spaces =", sp) # your own code
t = ""
for ch in s :
    if ch != " " :
        t = t + ch.lower() # your own code
rev = ""
for ch in t :
    rev = ch + rev
if t == rev :
    print("Palindrome")
else:
    print("Not Palindrome")
words = s.split()
print("Word count =", len(words)) # your own code
```



```python
#Start code here
s = input("Enter a string: ")
print("Length =", len(s)) # your own code
v = 0
c = 0
sp = 0
for ch in s.lower():
  if ch in "aeiou":
    v = v + 1
  elif ch == " ":
    sp = sp + 1
  elif ch.isalpha():
    c = c + 1
print("Vowels =", v, "Consonants =", c, "Spaces =", sp) # your own code
t = ""
for ch in s:
  if ch != " ":
    t = t + ch.lower() # your own code
rev = ""
for ch in t:
  rev = ch + rev
if t == rev:
  print("Palindrome")
else:
  print("Not Palindrome")
words = s.split()
print("Word count =", len(words)) # your own code

```

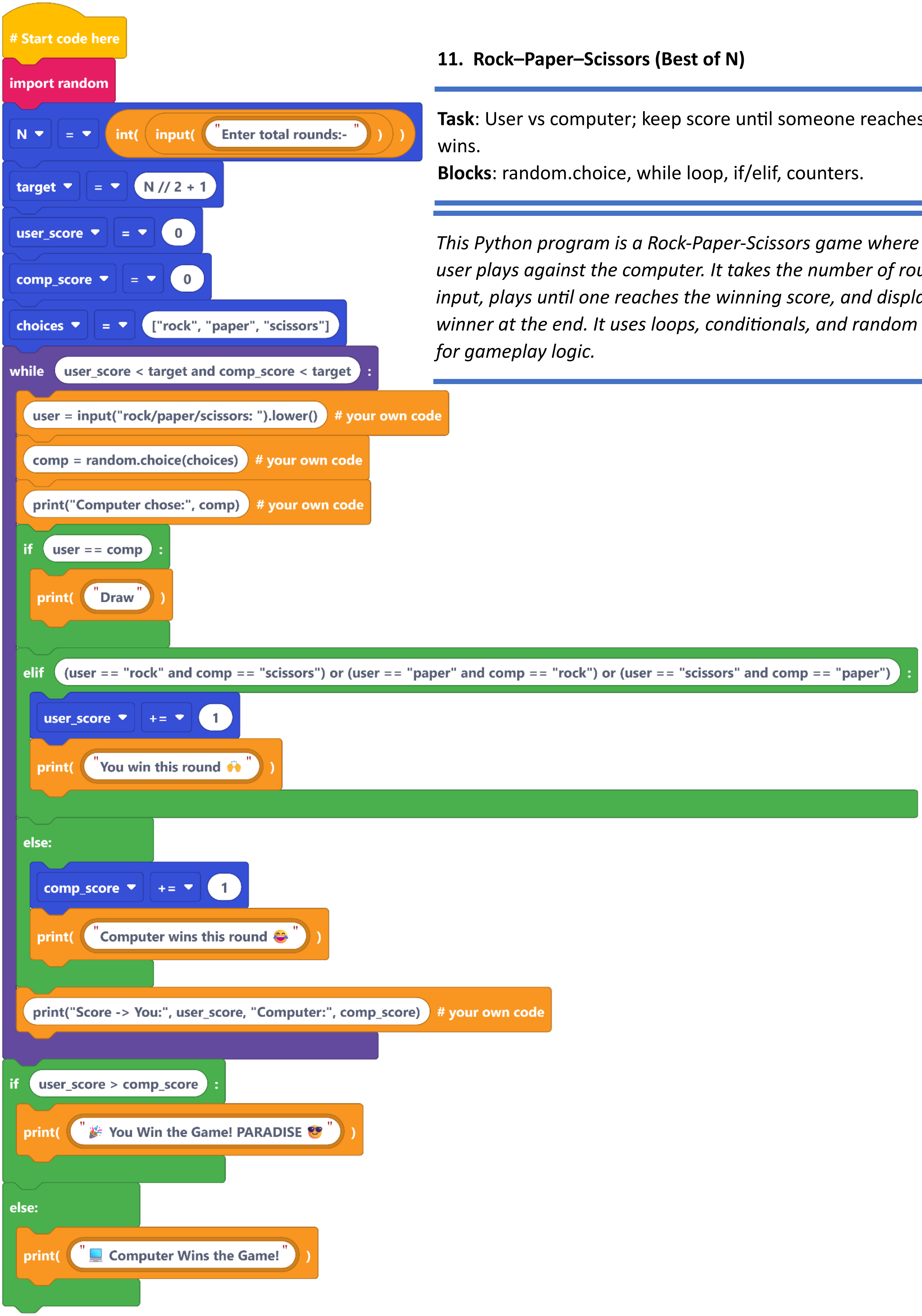


```
Enter a string: HELL PARADISE
Length = 13
Vowels = 5 Consonants = 7 Spaces = 1
Not Palindrome
Word count = 2
```

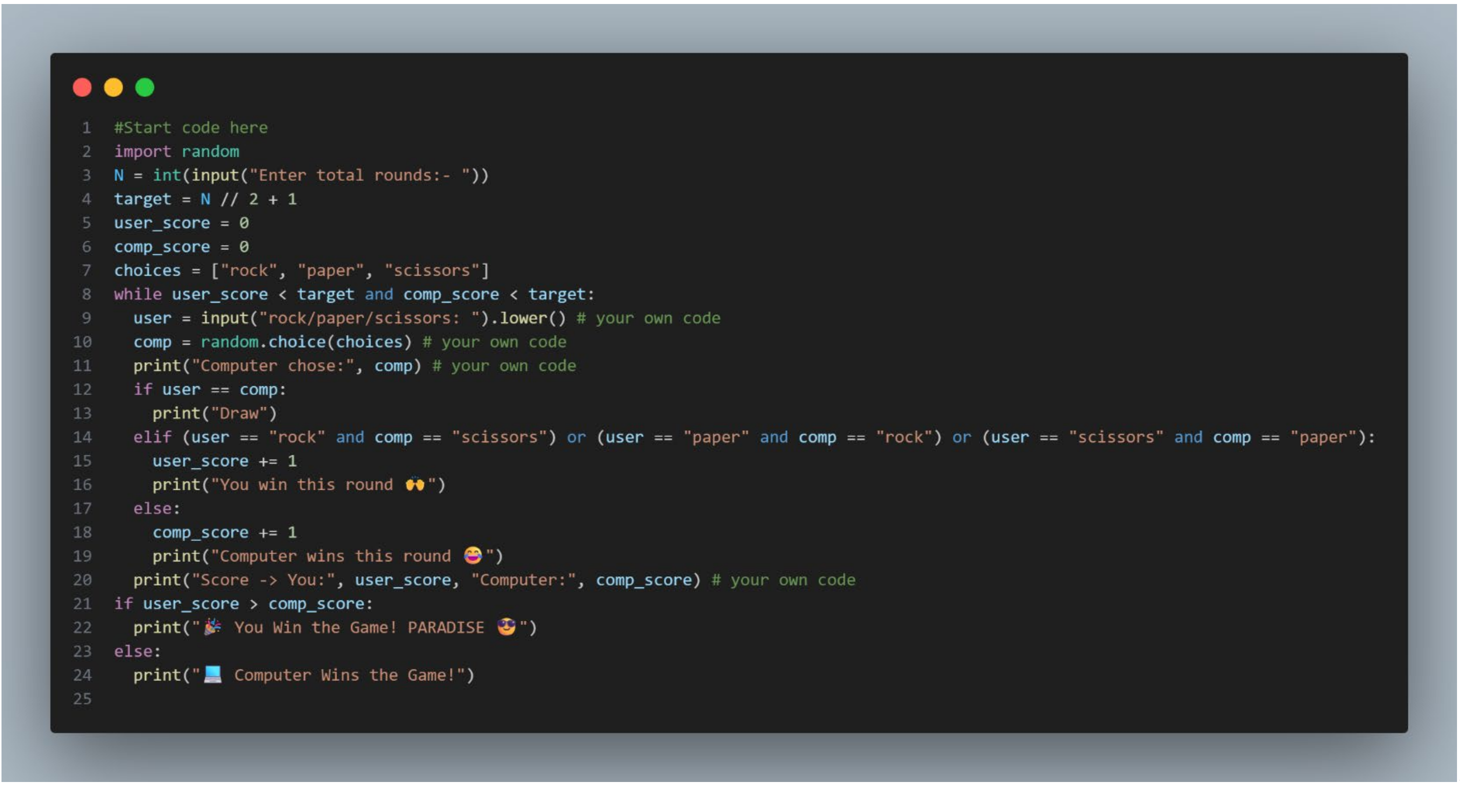
11. Rock–Paper–Scissors (Best of N)

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

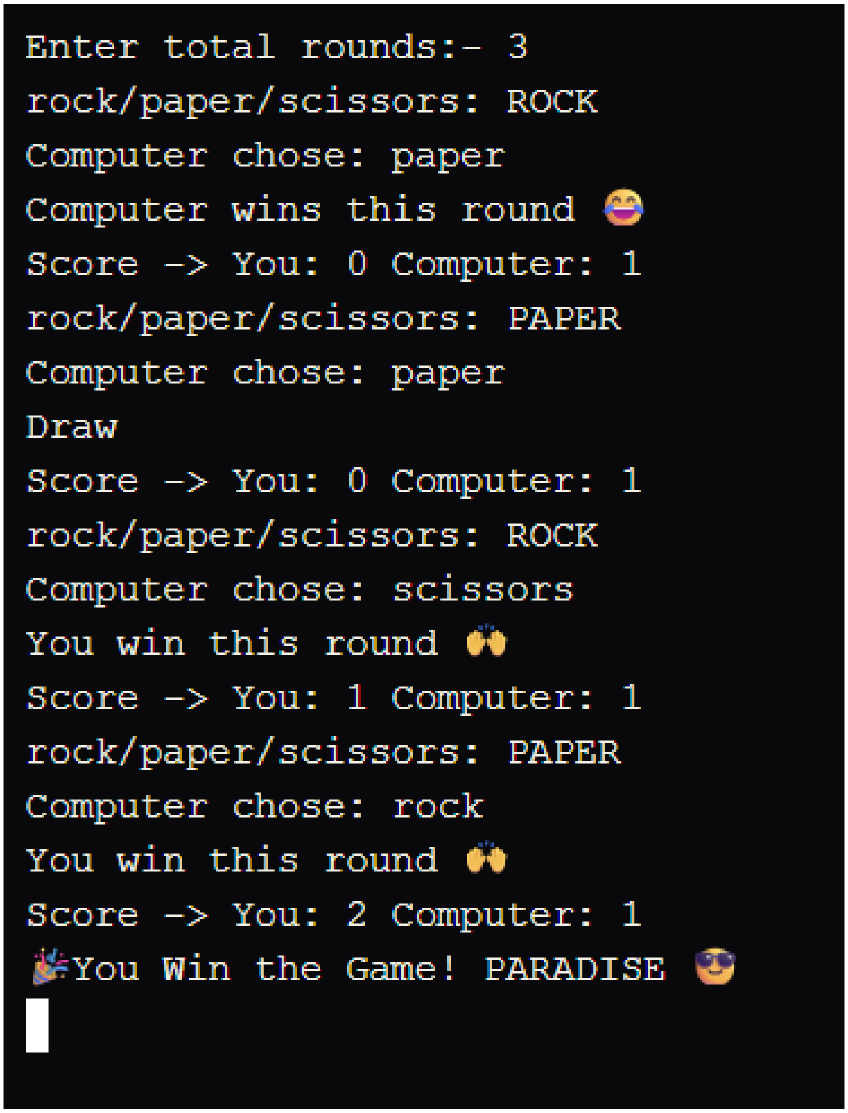
This Python program is a Rock-Paper-Scissors game where the user plays against the computer. It takes the number of rounds as input, plays until one reaches the winning score, and displays the winner at the end. It uses loops, conditionals, and random choice for gameplay logic.



```
# Start code here
import random
N = int( input( "Enter total rounds:-" ) )
target = N // 2 + 1
user_score = 0
comp_score = 0
choices = ["rock", "paper", "scissors"]
while user_score < target and comp_score < target :
    user = input("rock/paper/scissors: ").lower() # your own code
    comp = random.choice(choices) # your own code
    print("Computer chose:", comp) # your own code
    if user == comp :
        print( "Draw" )
    elif (user == "rock" and comp == "scissors") or (user == "paper" and comp == "rock") or (user == "scissors" and comp == "paper") :
        user_score += 1
        print( "You win this round 🙌" )
    else:
        comp_score += 1
        print( "Computer wins this round 😂" )
    print("Score -> You:", user_score, "Computer:", comp_score) # your own code
if user_score > comp_score :
    print( "🎉 You Win the Game! PARADISE 😎" )
else:
    print( "💻 Computer Wins the Game!" )
```



```python
#Start code here
import random
N = int(input("Enter total rounds:- "))
target = N // 2 + 1
user_score = 0
comp_score = 0
choices = ["rock", "paper", "scissors"]
while user_score < target and comp_score < target:
  user = input("rock/paper/scissors: ").lower() # your own code
  comp = random.choice(choices) # your own code
  print("Computer chose:", comp) # your own code
  if user == comp:
    print("Draw")
  elif (user == "rock" and comp == "scissors") or (user == "paper" and comp == "rock") or (user == "scissors" and comp == "paper"):
    user_score += 1
    print("You win this round 🙌")
  else:
    comp_score += 1
    print("Computer wins this round 😂")
  print("Score -> You:", user_score, "Computer:", comp_score) # your own code
if user_score > comp_score:
  print("🎉 You Win the Game! PARADISE 😎")
else:
  print("💻 Computer Wins the Game!")
```

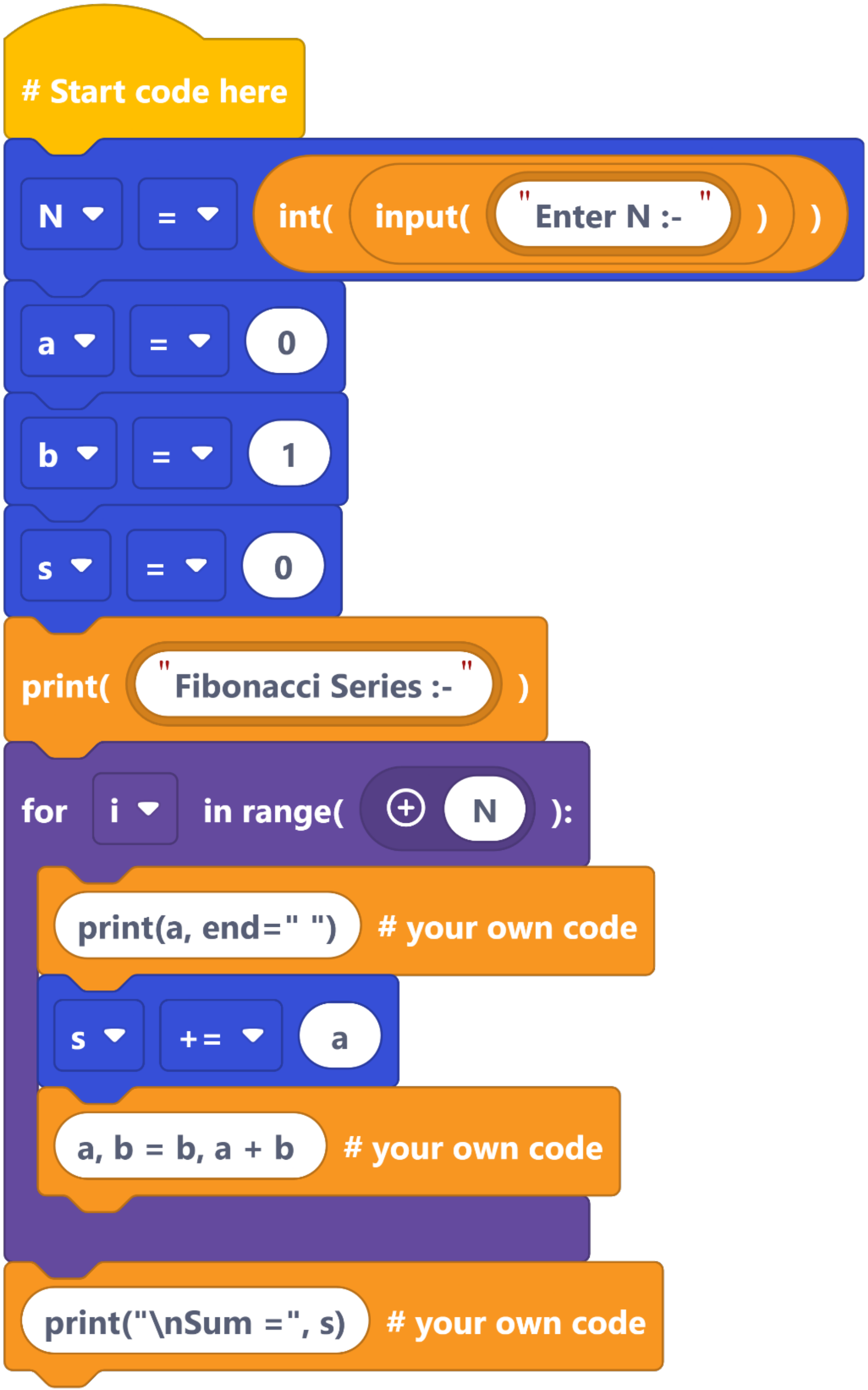


```
Enter total rounds:- 3
rock/paper/scissors: ROCK
Computer chose: paper
Computer wins this round 😂
Score -> You: 0 Computer: 1
rock/paper/scissors: PAPER
Computer chose: paper
Draw
Score -> You: 0 Computer: 1
rock/paper/scissors: ROCK
Computer chose: scissors
You win this round 🙌
Score -> You: 1 Computer: 1
rock/paper/scissors: PAPER
Computer chose: rock
You win this round 🙌
Score -> You: 2 Computer: 1
🎉You Win the Game! PARADISE 😎
```

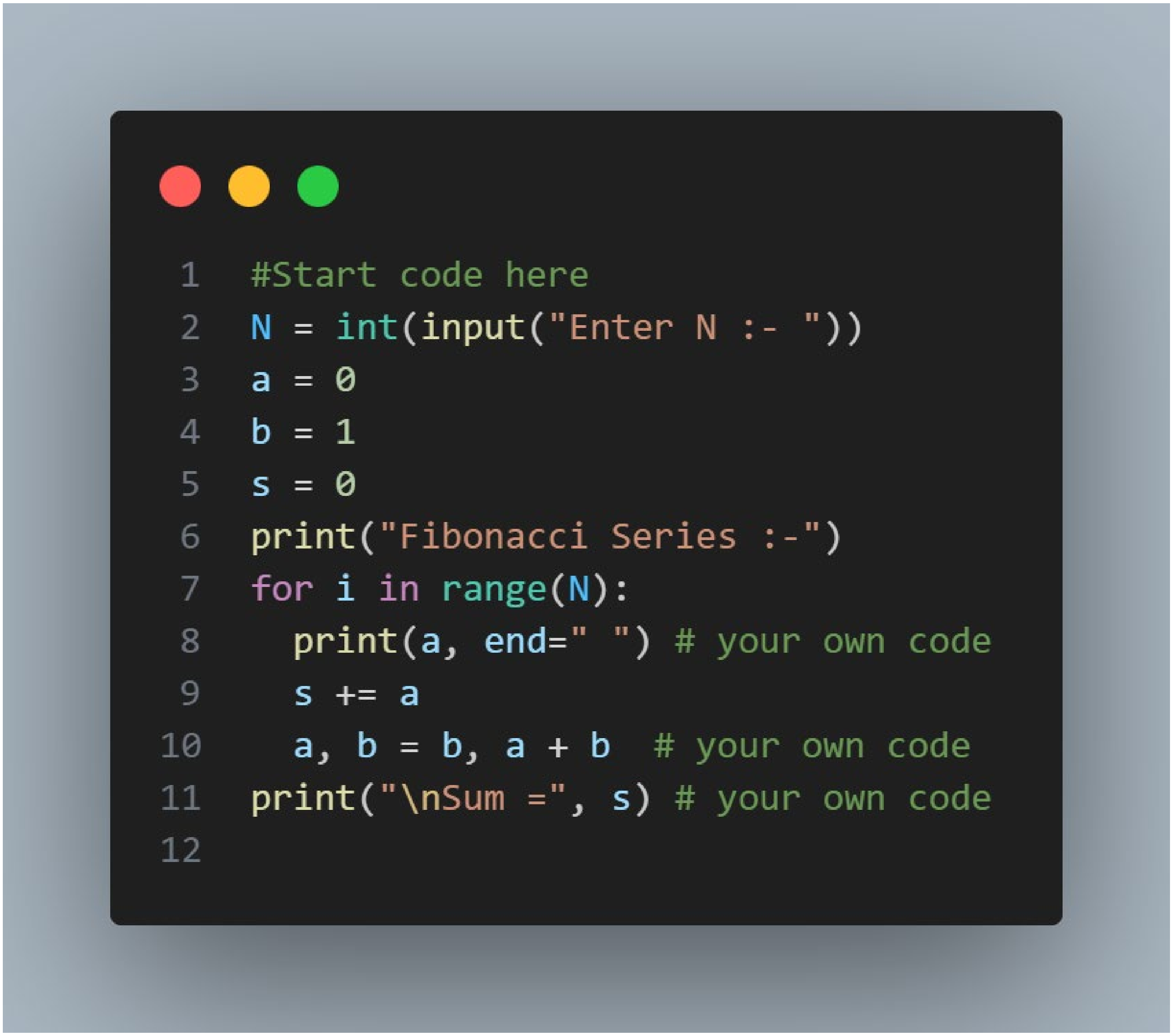
12. Fibonacci Series (Iterative)

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

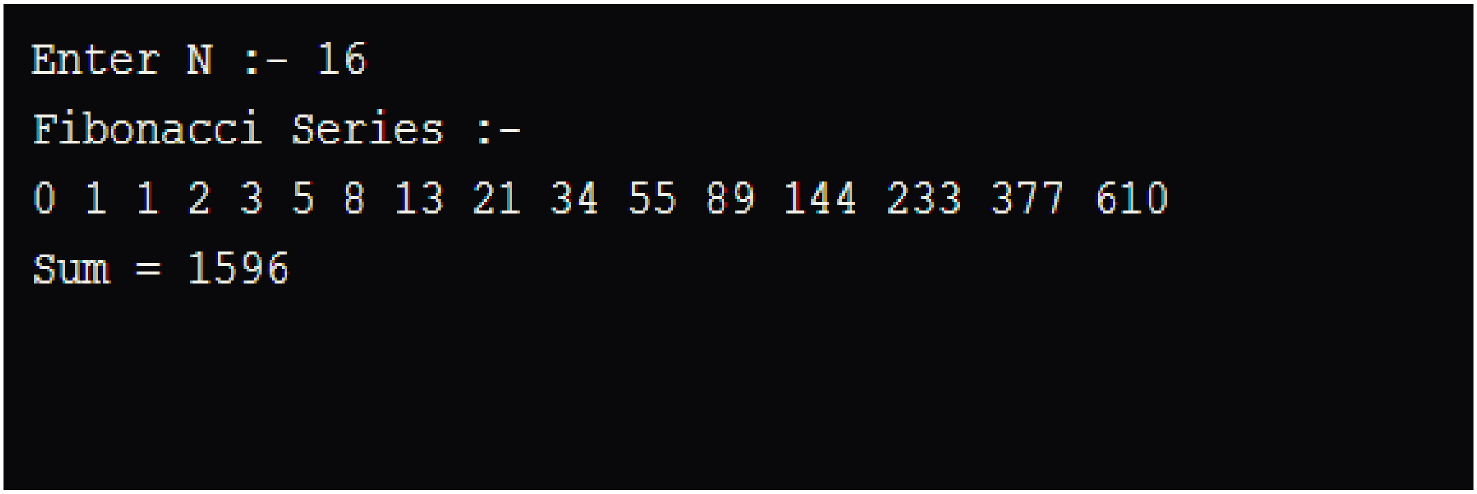
*This Python program prints the **Fibonacci series** up to **N terms**, where N is entered by the user. It also calculates and displays the **sum** of the series. The code uses a loop to generate the sequence and update the sum with each term.*



```
# Start code here
N = int( input( "Enter N :-" ) )
a = 0
b = 1
s = 0
print( "Fibonacci Series :-" )
for i in range( N ):
    print(a, end=" ") # your own code
    s += a
    a, b = b, a + b # your own code
print("\nSum =", s) # your own code
```



```python
#Start code here
N = int(input("Enter N :- "))
a = 0
b = 1
s = 0
print("Fibonacci Series :-")
for i in range(N):
  print(a, end=" ") # your own code
  s += a
  a, b = b, a + b  # your own code
print("\nSum =", s) # your own code
```



```
Enter N :- 16
Fibonacci Series :-
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
Sum = 1596
```