

Practical Activity Report

Computer Science (CP) CS181001

Submitted by

Husain Lokhandwala

241030011015

Submitted to

Dr. Ashish Soni



Department of Electrical and Computer Science

Institute of Infrastructure, Technology research and Management

(Established by Govt. of Gujrat)

Ahmedabad, Gujarat

Table of Content

S no.	Name of Practical	Date	Remark
1	Practical 1		
2	Practical 2		
3	Practical 3		
4	Practical 4		
5	Practical 5		
6	Practical 6		
7	Practical 7		
8	Practical 8		

Practical 1:

Question:

Swapping numbers.

1. Write a Python program to swap the values of two variables.
2. Create a Python function that takes no arguments, initializes two variables with specific values, swaps their values using a temporary variable, and prints the before and after values.
3. Develop a Python function that swaps the values of two predefined variables without using a temporary third variable. The function should display the values before and after the swap.
4. Implement a Python function to swap two variables using arithmetic operations only (addition and subtraction). The function should print the initial and final values of the variables.
5. Write a Python program that takes two numerical inputs from the user and then defines and calls functions to calculate and display their sum, difference, and product.
6. Extend the program to include a function that takes a numerator and a denominator as input (from the user within the function) and prints the result of their division.
7. Add a function to the program that takes a base and an exponent as input (from the user within the function) and calculates and prints the result of the exponentiation.
8. Create two Python functions: one to calculate and print the square of two predefined numbers (obtained as initial user input), and another to calculate and print their square roots.
9. Develop two Python functions: one to calculate and print the cube of the initial two numbers, and another to calculate and print their cube roots.
10. Write a Python function that takes the two initial numbers as input and prints the remainder of their division.
11. Implement two Python functions: each should prompt the user for two new numbers, perform an increment (add 1) or decrement (subtract 1) operation on each, and then display the resulting values.
12. Write a Python function that allows the user to calculate the simple interest by taking the principal amount, rate of interest, and time period as input.
13. Create a Python function that enables the user to calculate the compound interest by taking the principal amount, rate of interest, time period, and the number of times the interest is compounded per year as input.

Code:

```
# 1

a = 10

b = 20

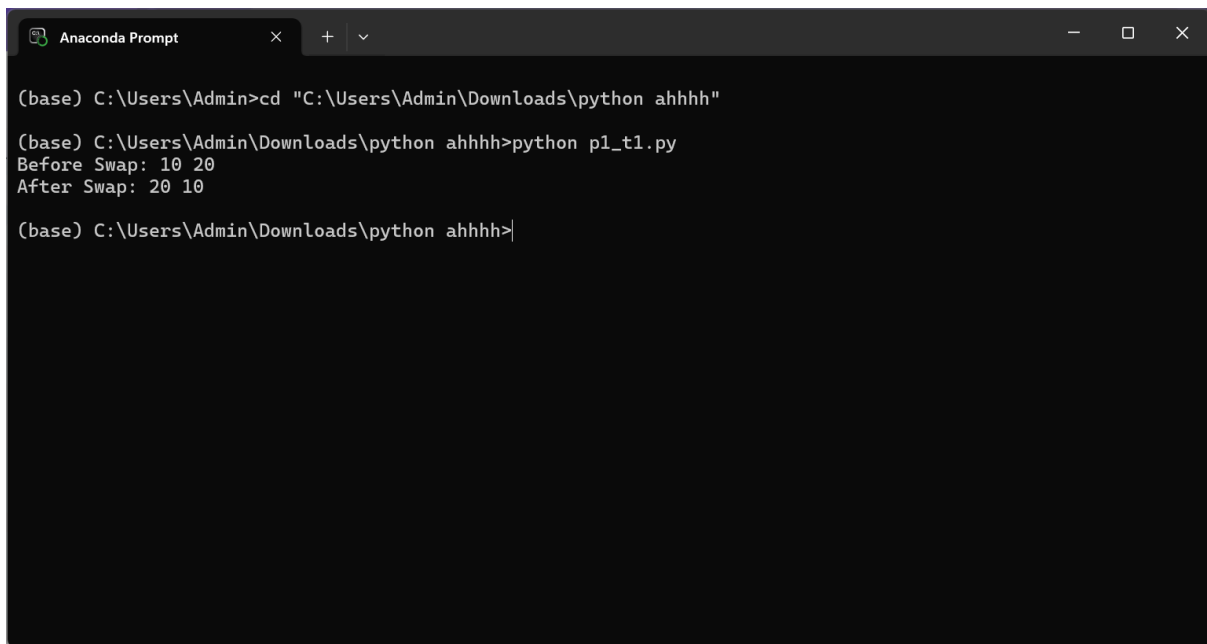
print("Before Swap:", a, b)

c = a

a = b

b = c
```

```
print("After Swap:", a, b)
```

A screenshot of an Anaconda Prompt terminal window. The window has a dark background and a title bar that says "Anaconda Prompt". The terminal shows the following commands and output:

```
(base) C:\Users\Admin>cd "C:\Users\Admin\Downloads\python ahhhh"  
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t1.py  
Before Swap: 10 20  
After Swap: 20 10  
(base) C:\Users\Admin\Downloads\python ahhhh>|
```

2 and 3

```
def swapOneVariable():  
    a = 15  
    b = 30  
    print("Before Swap:", a, b)  
    c = a  
    a = b  
    b = c  
    print("After Swap:", a, b)
```

```
swapOneVariable()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t2.py
python: can't open file 'C:\\Users\\Admin\\Downloads\\python ahhhh\\p1_t2.py': [Errno 2] N
o such file or directory

(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t2_3.py
Before Swap: 15 30
After Swap: 30 15

(base) C:\Users\Admin\Downloads\python ahhhh>
```

4

```
def swapWithArithmetic():
```

```
    a = 25
```

```
    b = 50
```

```
    print("Before Swap:", a, b)
```

```
    a = a + b
```

```
    b = a - b
```

```
    a = a - b
```

```
    print("After Swap:", a, b)
```

```
swapWithArithmetic()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t4.py
Before Swap: 25 50
After Swap: 50 25
(base) C:\Users\Admin\Downloads\python ahhhh>
```

5

def arithmetic():

 a = int(input("Enter the first number: "))

 b = int(input("Enter the second number: "))

 print("Sum:", a + b)

 print("Difference:", a - b)

 print("Product:", a * b)

arithmetic()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t5.py
Enter the first number: 10
Enter the second number: 3
Sum: 13
Difference: 7
Product: 30

(base) C:\Users\Admin\Downloads\python ahhhh>
```

6

def divide():

```
    numerator = int(input("Enter the numerator: "))
    denominator = int(input("Enter the denominator: "))
    if denominator != 0:
        print("Division Result:", numerator / denominator)
    else:
        print("Error: Division by zero is not allowed.")
```

divide()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t6.py
Enter the numerator: 10
Enter the denominator: 2
Division Result: 5.0

(base) C:\Users\Admin\Downloads\python ahhhh>
```

7

def power():

base = int(input("Enter the base: "))

exponent = int(input("Enter the exponent: "))

result = base ** exponent

print("Result:", result)

power()


```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t7.py
Enter the base: 10
Enter the exponent: 3
Result: 1000
(base) C:\Users\Admin\Downloads\python ahhhh>
```

8

```
import math
```

```
def square():
```

```
    a = int(input("Enter the first number: "))
```

```
    b = int(input("Enter the second number: "))
```

```
    print("Square of", a, "is", a * a)
```

```
    print("Square of", b, "is", b * b)
```

```
def squareRoot():
```

```
    a = int(input("Enter the first number: "))
```

```
    b = int(input("Enter the second number: "))
```

```
    print("Square root of", a, "is", math.sqrt(a))
```

```
    print("Square root of", b, "is", math.sqrt(b))
```

```
square()
```

```
squareRoot()
```

```
Anaconda Prompt - python p × + ∨  
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t8.py  
Enter the first number: 10  
Enter the second number: 4  
Square of 10 is 100  
Square of 4 is 16  
Enter the first number:
```

9

def cube():

```
    a = int(input("Enter the first number: "))  
    b = int(input("Enter the second number: "))  
    print("Cube of", a, "is", a * a * a)  
    print("Cube of", b, "is", b * b * b)
```

def cubeRoot():

```
    a = int(input("Enter the first number: "))  
    b = int(input("Enter the second number: "))  
    print("Cube root of", a, "is", math.pow(a, 1/3))  
    print("Cube root of", b, "is", math.pow(b, 1/3))
```

cube()

cubeRoot()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t9.py
Enter the first number: 10
Enter the second number: 3
Cube of 10 is 1000
Cube of 3 is 27
Enter the first number: 5
Enter the second number: 6
Cube root of 5 is 1.7099759466766968
Cube root of 6 is 1.8171205928321397

(base) C:\Users\Admin\Downloads\python ahhhh>
```

10

def remainder():

 a = int(input("Enter the first number: "))

 b = int(input("Enter the second number: "))

 if b != 0:

 print("Remainder of", a, "divided by", b, "is", a % b)

 else:

 print("Error: Division by zero is not allowed.")

remainder()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t10.py
Enter the first number: 4
Enter the second number: 5
Remainder of 4 divided by 5 is 4
(base) C:\Users\Admin\Downloads\python ahhhh>
```

11

def increment():

 a = int(input("Enter the first number: "))

 b = int(input("Enter the second number: "))

 a = a + 1

 b = b + 1

 print("After incrementing, first number is", a, "and second number is", b)

def decrement():

 a = int(input("Enter the first number: "))

 b = int(input("Enter the second number: "))

 a = a - 1

 b = b - 1

 print("After decrementing, first number is", a, "and second number is", b)

increment()

decrement()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t11.py
Enter the first number: 87
Enter the second number: 56
After incrementing, first number is 88 and second number is 57
Enter the first number: 3
Enter the second number: 5
After decrementing, first number is 2 and second number is 4

(base) C:\Users\Admin\Downloads\python ahhhh>
```

12

def Sinterest():

principal = float(input("Enter the principal amount: "))

rate = float(input("Enter the rate of interest: "))

time = float(input("Enter the time period in years: "))

simple_interest = (principal * rate * time) / 100

print("Simple Interest is", simple_interest)

Sinterest()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t12.py
Enter the principal amount: 10000
Enter the rate of interest: 1
Enter the time period in years: 4
Simple Interest is 400.0

(base) C:\Users\Admin\Downloads\python ahhhh>
```

13

def Cinterest():

 principal = float(input("Enter the principal amount: "))

 rate = float(input("Enter the rate of interest: "))

 time = float(input("Enter the time period in years: "))

 n = int(input("Enter the number of times interest is compounded per year: "))

 compound_interest = principal * (1 + rate / (100 * n))**(n * time) - principal

 print("Compound Interest is", compound_interest)

Cinterest()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p1_t13.py
Enter the principal amount: 16543
Enter the rate of interest: 87
Enter the time period in years: 7
Enter the number of times interest is compounded per year: 3
Compound Interest is 3458804.0984960785

(base) C:\Users\Admin\Downloads\python ahhhh>
```

Practical 2:

Question:

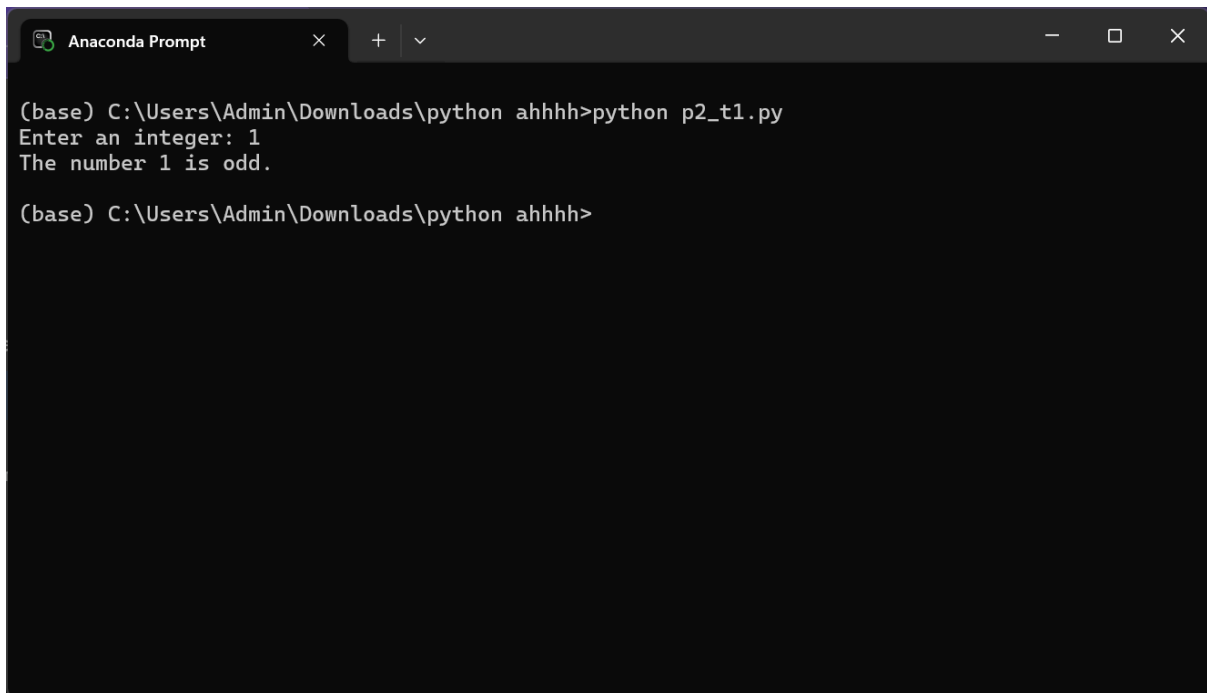
1. Write a Python program that takes an integer as input from the user and determines whether the number is even or odd. The program should then print an appropriate message indicating the result.
2. Define a Python function that takes two integer arguments and prints a message indicating the relationship between them (greater than, less than, or equal). Call this function with two numbers entered by the user.
3. Write a Python program that takes three integers as input and determines the greatest among them using conditional statements. The program should then print the greatest number.
4. Write a Python program that takes three integers as input and determines the greatest number among them using nested if statements. The program should print the greatest number.
5. Create a Python function that takes six subject marks as input, calculates the average, and prints the grade using nested if statements to determine the grade range. The function should include error handling for invalid average values.

Code:

1

```
def evenOdd():  
    number = int(input("Enter an integer: "))  
    if number % 2 == 0:  
        print("The number", number, "is even.")  
    else:  
        print("The number", number, "is odd.")
```

evenOdd()

A screenshot of an Anaconda Prompt window. The title bar shows 'Anaconda Prompt' with standard window controls. The terminal content shows a command prompt at '(base) C:\Users\Admin\Downloads\python ahhhh>' where the user has run 'python p2_t1.py'. The program prompts 'Enter an integer: 1' and outputs 'The number 1 is odd.'. The prompt returns to '(base) C:\Users\Admin\Downloads\python ahhhh>'.

```
(base) C:\Users\Admin\Downloads\python ahhhh>python p2_t1.py  
Enter an integer: 1  
The number 1 is odd.  
  
(base) C:\Users\Admin\Downloads\python ahhhh>
```

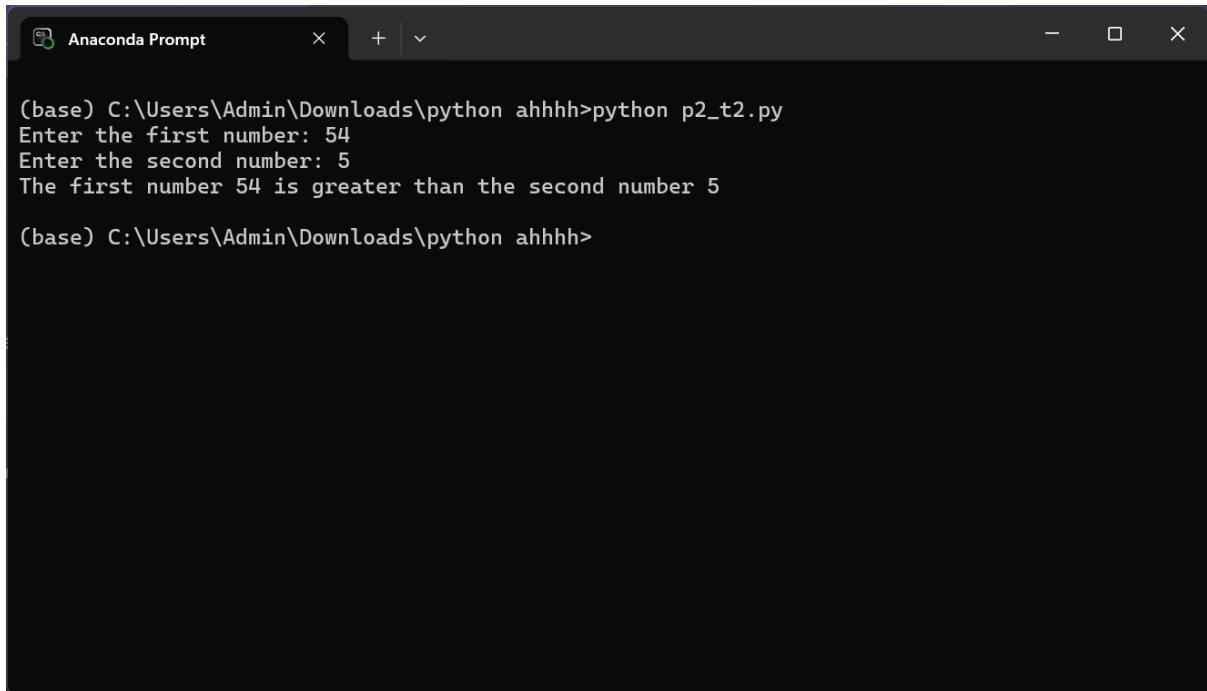
2

```
def compare_num():  
    a = int(input("Enter the first number: "))  
    b = int(input("Enter the second number: "))  
    if a > b:  
        print("The first number", a, "is greater than the second number", b)  
    elif a < b:
```



```
    print("The first number", a, "is less than the second number", b)
else:
    print("The first number", a, "is equal to the second number", b)
```

compare_num()

A screenshot of an Anaconda Prompt window. The window title is "Anaconda Prompt". The command prompt shows the following text:

```
(base) C:\Users\Admin\Downloads\python ahhhh>python p2_t2.py
Enter the first number: 54
Enter the second number: 5
The first number 54 is greater than the second number 5
(base) C:\Users\Admin\Downloads\python ahhhh>
```

3

def greatest():

```
    a = int(input("Enter the first number: "))
    b = int(input("Enter the second number: "))
    c = int(input("Enter the third number: "))
    if a > b and a > c:
        print("The greatest number is", a)
    elif b > c:
        print("The greatest number is", b)
    else:
        print("The greatest number is", c)
```

greatest()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p2_t3.py
Enter the first number: 345678
Enter the second number: 231
Enter the third number: 6543
The greatest number is 345678
(base) C:\Users\Admin\Downloads\python ahhhh>
```

4

```
def greatest_nested():
```

```
    a = int(input("Enter the first number: "))
```

```
    b = int(input("Enter the second number: "))
```

```
    c = int(input("Enter the third number: "))
```

```
    if a > b:
```

```
        if a > c:
```

```
            print("The greatest number is", a)
```

```
        else:
```

```
            print("The greatest number is", c)
```

```
    else:
```

```
        if b > c:
```

```
            print("The greatest number is", b)
```

```
        else:
```

```
            print("The greatest number is", c)
```

```
greatest_nested()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p2_t4.py
Enter the first number: 1243
Enter the second number: 65
Enter the third number: 765
The greatest number is 1243

(base) C:\Users\Admin\Downloads\python ahhhh>
```

5

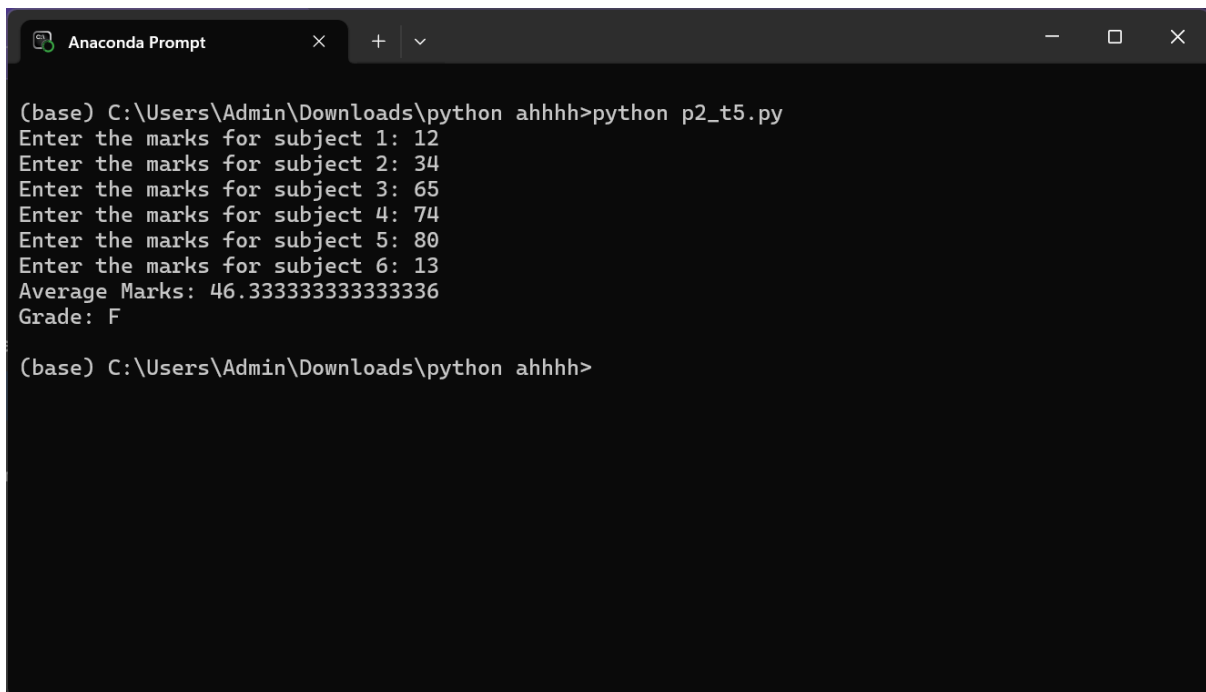
```
def grade():
    marks = []
    for i in range(1, 7):
        mark = int(input("Enter the marks for subject " + str(i) + ": "))
        if mark > 100:
            print("Error: Marks cannot exceed 100.")
            return
        marks.append(mark)

    average = sum(marks) / 6
    print("Average Marks:", average)

    if average >= 90:
        print("Grade: A")
    elif average >= 80:
        print("Grade: B")
    elif average >= 70:
```

```
    print("Grade: C")
elif average >= 60:
    print("Grade: D")
else:
    print("Grade: F")
```

grade()



```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p2_t5.py
Enter the marks for subject 1: 12
Enter the marks for subject 2: 34
Enter the marks for subject 3: 65
Enter the marks for subject 4: 74
Enter the marks for subject 5: 80
Enter the marks for subject 6: 13
Average Marks: 46.333333333333336
Grade: F
(base) C:\Users\Admin\Downloads\python ahhhh>
```

Practical 3:

Question:

1. Write a Python function that prints "Hello World" five times using a loop.
2. Create a Python function that prints all the numbers from 1 to 100 (inclusive) using a loop.
3. Write a Python function that prints all the odd natural numbers from 1 to 100 (inclusive).
4. Implement a Python function that prints each number from 1 to 100, indicating whether it's even or odd.
5. Write a Python function that calculates and prints the sum of all natural numbers from 1 to 100 (inclusive).

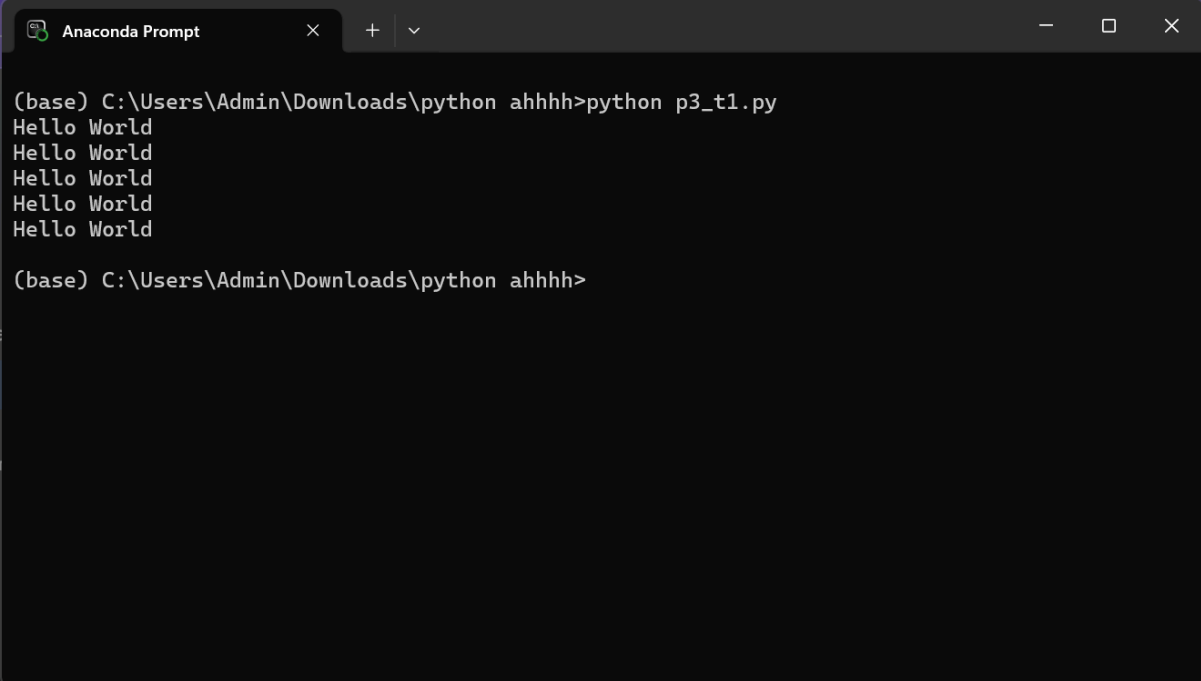
6. Create a Python function that calculates and prints the sum of all even numbers and the sum of all odd numbers from 1 to 100 (inclusive).
7. Write a Python function that prints all numbers from 1 to 100 (inclusive) that are divisible by 5 or 7.
8. Implement a Python function that takes an integer as input and calculates and prints its factorial.
9. Write a Python function that takes an integer n as input and generates and prints the Fibonacci series up to the nth term.
10. Write a Python function that takes an integer as input and determines whether it is a prime number. The function should then print an appropriate message indicating the result.

Code:

1

```
def printhelloworld():  
    for i in range(5):  
        print("Hello World")
```

printhelloworld()



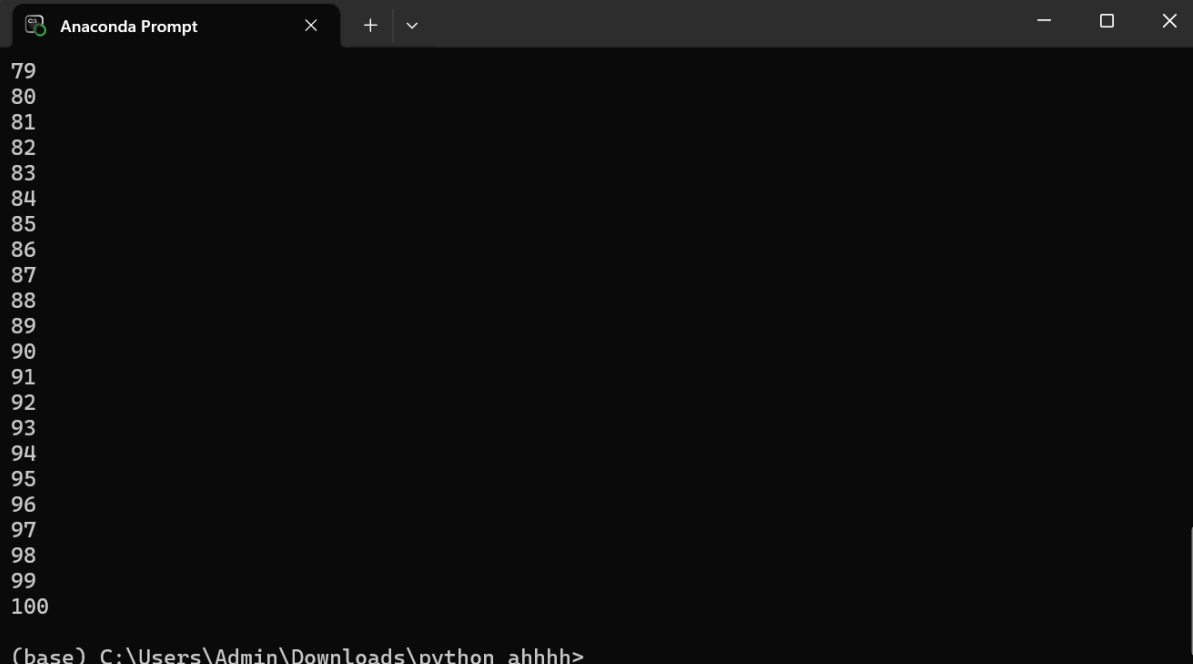
The screenshot shows an Anaconda Prompt window with a dark background. The title bar reads 'Anaconda Prompt'. The command prompt shows the user running 'python p3_t1.py' in the directory 'C:\Users\Admin\Downloads\python ahhhh'. The output of the script is five lines of 'Hello World' printed one after another. The prompt then returns to 'C:\Users\Admin\Downloads\python ahhhh>'.

```
(base) C:\Users\Admin\Downloads\python ahhhh>python p3_t1.py  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
  
(base) C:\Users\Admin\Downloads\python ahhhh>
```

2

```
def printnumbers():  
    for i in range(1, 101):  
        print(i)
```

printnumbers()



The image shows a screenshot of an Anaconda Prompt window. The window has a dark theme and a title bar that says "Anaconda Prompt". Inside the window, a list of numbers is displayed, starting from 79 and ending at 100, with each number on a new line. At the bottom of the window, the command prompt shows the current directory as "C:\Users\Admin\Downloads\python" and the prompt character is "ahhhh".

```
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

(base) C:\Users\Admin\Downloads\python ahhhh>
```

3

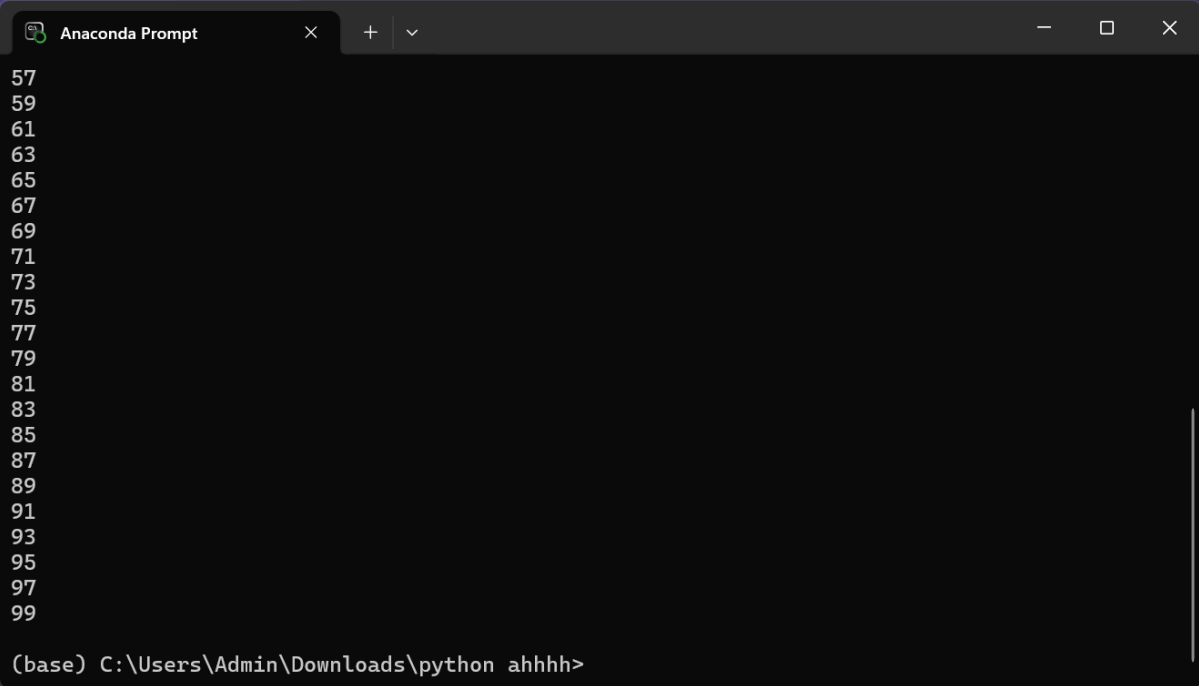
def printodddnumbers():

for i in range(1, 101):

if i % 2 != 0:

print(i)

printodddnumbers()



The image shows a screenshot of an Anaconda Prompt terminal window. The window has a dark background and a title bar that says "Anaconda Prompt". Inside the terminal, a list of odd numbers is displayed, starting from 57 and ending at 99, with an increment of 2. The numbers are listed vertically on the left side of the terminal. At the bottom of the terminal, the command prompt shows the current directory as "C:\Users\Admin\Downloads\python" and the prompt character is ">".

```
57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99

(base) C:\Users\Admin\Downloads\python ahhhh>
```

4

```
def print_even_or_odd():
    for i in range(1, 101):
        if i % 2 == 0:
            print(i, "is even")
        else:
            print(i, "is odd")

print_even_or_odd()
```

```
Anaconda Prompt
79 is odd
80 is even
81 is odd
82 is even
83 is odd
84 is even
85 is odd
86 is even
87 is odd
88 is even
89 is odd
90 is even
91 is odd
92 is even
93 is odd
94 is even
95 is odd
96 is even
97 is odd
98 is even
99 is odd
100 is even

(base) C:\Users\Admin\Downloads\python_ahhhh>
```

5

```
def sum_of_numbers():
    total = 0
    for i in range(1, 101):
        total += i
    print("The sum of all natural numbers from 1 to 100 is", total)

sum_of_numbers()
```



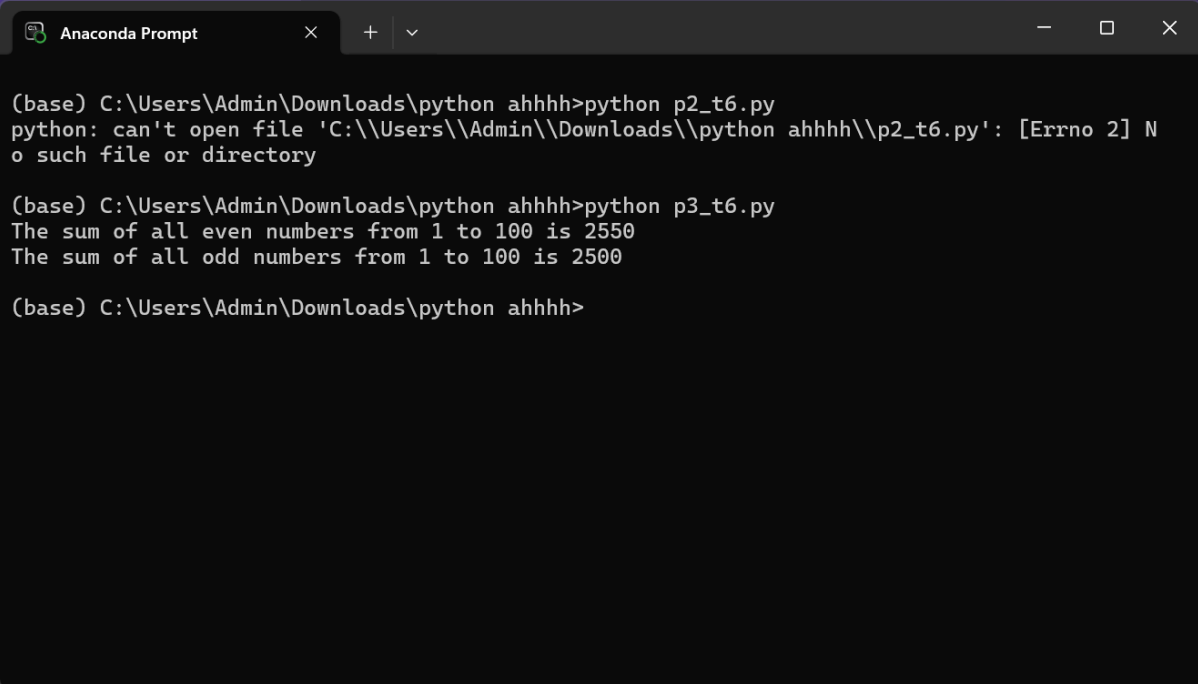
```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p2_t5.py
Enter the marks for subject 1: 12
Enter the marks for subject 2: 34
Enter the marks for subject 3: 56
Enter the marks for subject 4: 78
Enter the marks for subject 5: 89
Enter the marks for subject 6: 90
Average Marks: 59.83333333333336
Grade: F

(base) C:\Users\Admin\Downloads\python ahhhh>
```

6

```
def sum_even_and_odd():
    sum_even = 0
    sum_odd = 0
    for i in range(1, 101):
        if i % 2 == 0:
            sum_even += i
        else:
            sum_odd += i
    print("The sum of all even numbers from 1 to 100 is", sum_even)
    print("The sum of all odd numbers from 1 to 100 is", sum_odd)
```

sum_even_and_odd()



```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p2_t6.py
python: can't open file 'C:\\Users\\Admin\\Downloads\\python ahhhh\\p2_t6.py': [Errno 2] N
o such file or directory

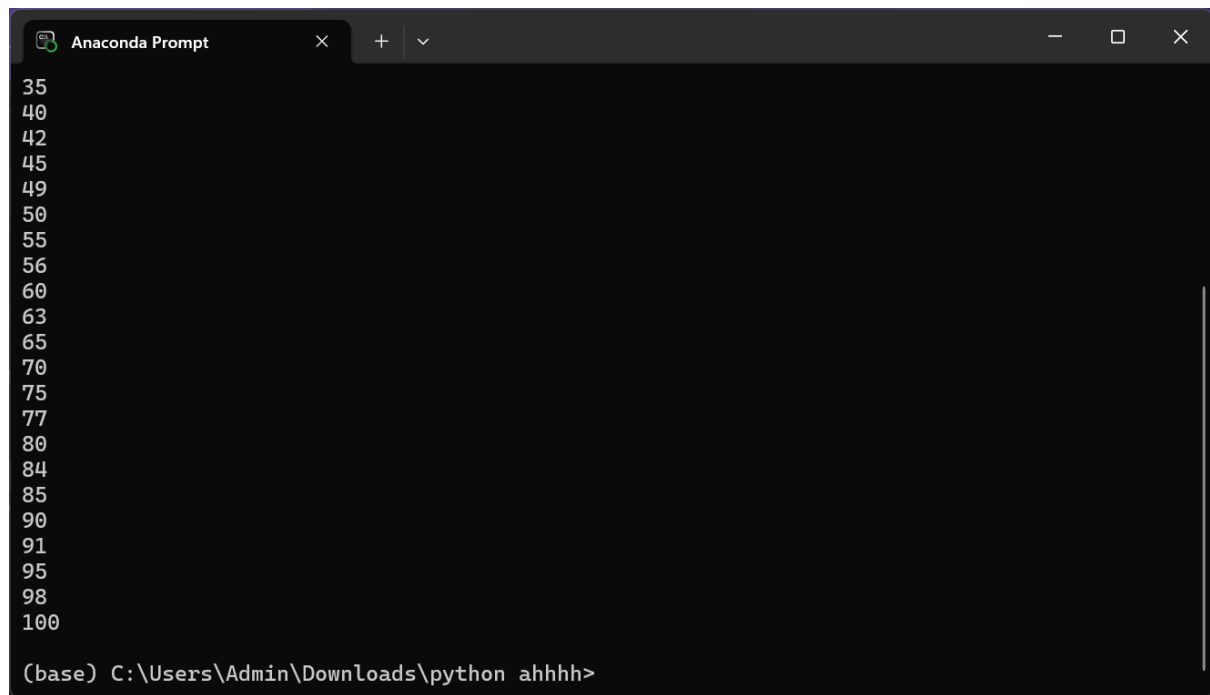
(base) C:\Users\Admin\Downloads\python ahhhh>python p3_t6.py
The sum of all even numbers from 1 to 100 is 2550
The sum of all odd numbers from 1 to 100 is 2500

(base) C:\Users\Admin\Downloads\python ahhhh>
```

7

```
def divisible_by_5_or_7():
    for i in range(1, 101):
        if i % 5 == 0 or i % 7 == 0:
            print(i)
```

divisible_by_5_or_7()

A screenshot of an Anaconda Prompt window. The window has a dark background and a title bar that says "Anaconda Prompt". Inside the window, a list of numbers is displayed on the left side: 35, 40, 42, 45, 49, 50, 55, 56, 60, 63, 65, 70, 75, 77, 80, 84, 85, 90, 91, 95, 98, and 100. Below the list, the command prompt shows the text "(base) C:\Users\Admin\Downloads\python ahhhh>".

```
35
40
42
45
49
50
55
56
60
63
65
70
75
77
80
84
85
90
91
95
98
100

(base) C:\Users\Admin\Downloads\python ahhhh>
```

8

```
def calculate_factorial():
    number = int(input("Enter an integer: "))
    if number < 0:
        print("Factorial is not defined for negative numbers.")
        return
    factorial = 1
    for i in range(1, number + 1):
        factorial *= i
    print("The factorial of", number, "is", factorial)

calculate_factorial()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p3_t8.py
Enter an integer: 12
The factorial of 12 is 479001600
(base) C:\Users\Admin\Downloads\python ahhhh>
```

9

```
def generate_fibonacci():
    n = int(input("Enter the number of terms: "))
    if n <= 0:
        print("Please enter a positive integer.")
        return
    a, b = 0, 1
    print("Fibonacci series up to", n, "terms:")
    for _ in range(n):
        print(a, end=" ")
        a, b = b, a + b
    print()

generate_fibonacci()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p3_t9.py
Enter the number of terms: 10
Fibonacci series up to 10 terms:
0 1 1 2 3 5 8 13 21 34

(base) C:\Users\Admin\Downloads\python ahhhh>
```

10

```
def is_prime():
    n = int(input("Enter a number: "))
    if n <= 1:
        print(n, "is not prime")
        return
    for i in range(2, n):
        if n % i == 0:
            print(n, "is not prime")
            return
    print(n, "is prime")

is_prime()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p3_t10.py
Enter a number: 12
12 is not prime

(base) C:\Users\Admin\Downloads\python ahhhh>
```

Practical 4:

Question:

1. Write a Python function that prints "Hello World" five times using a while loop.
2. Create a Python function that prints all the numbers from 1 to 100 (inclusive) using a while loop.
3. Write a Python function that prints all the odd natural numbers from 1 to 100 (exclusive) using a while loop.
4. Implement a Python function that prints each number from 0 to 100, indicating whether it's even or odd, using a while loop.
5. Write a Python function that calculates and prints the sum of all natural numbers from 0 to 100 (inclusive) using a while loop.
6. Create a Python function that calculates and prints the sum of all even numbers and the sum of all odd numbers from 0 to 100 (inclusive) using a while loop.
7. Write a Python function that prints all numbers from 0 to 100 (inclusive) that are divisible by 5 or 7 using a while loop.
8. Implement a Python function that takes an integer as input and calculates and prints its factorial using a while loop.
9. Write a Python function that takes an integer n as input and generates and prints the Fibonacci series up to the nth term using a while loop.
10. Write a Python function that takes an integer n as input and determines whether it is a prime number using a while loop. The function should then print an appropriate message indicating the result.

Code:

1

```
def print_hello_world():
```

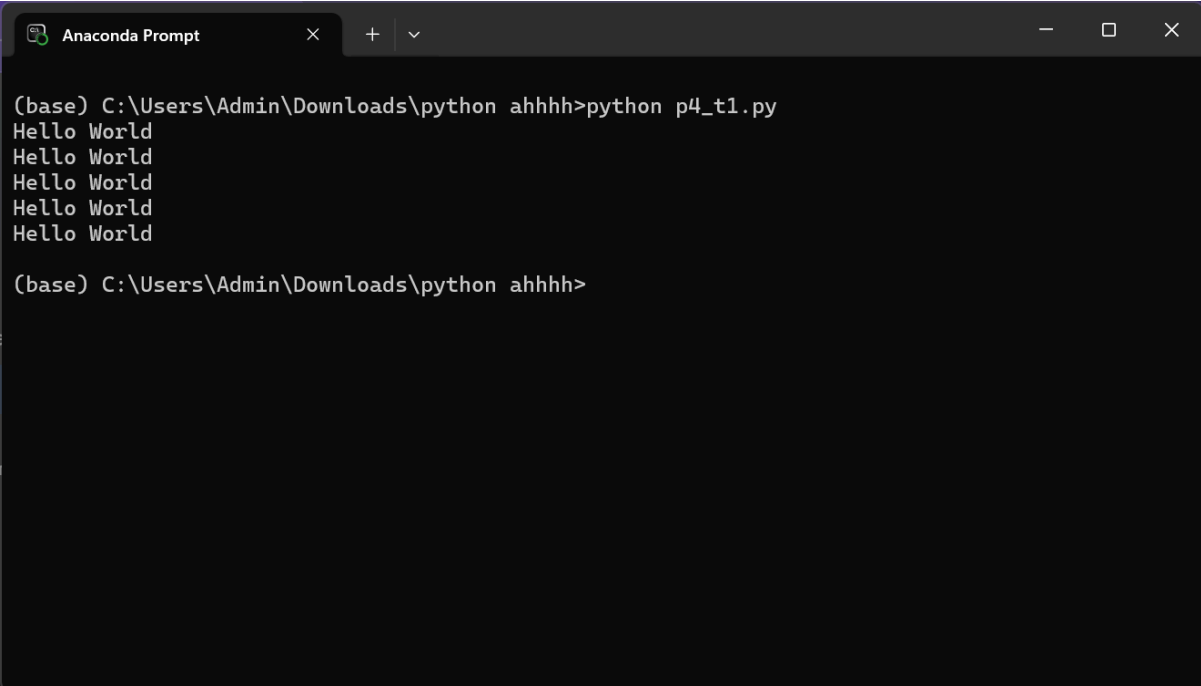
```
    i = 0
```

```
    while i < 5:
```

```
        print("Hello World")
```

```
        i = i + 1
```

```
print_hello_world()
```

A screenshot of an Anaconda Prompt window. The title bar shows 'Anaconda Prompt' with standard window controls. The command prompt shows the execution of a Python script:

```
(base) C:\Users\Admin\Downloads\python ahhhh>python p4_t1.py
```

 The output of the script is five lines of 'Hello World'. The prompt then returns to the command line:

```
(base) C:\Users\Admin\Downloads\python ahhhh>
```

```
(base) C:\Users\Admin\Downloads\python ahhhh>python p4_t1.py
Hello World
Hello World
Hello World
Hello World
Hello World

(base) C:\Users\Admin\Downloads\python ahhhh>
```

2

```
def print_numbers():
```

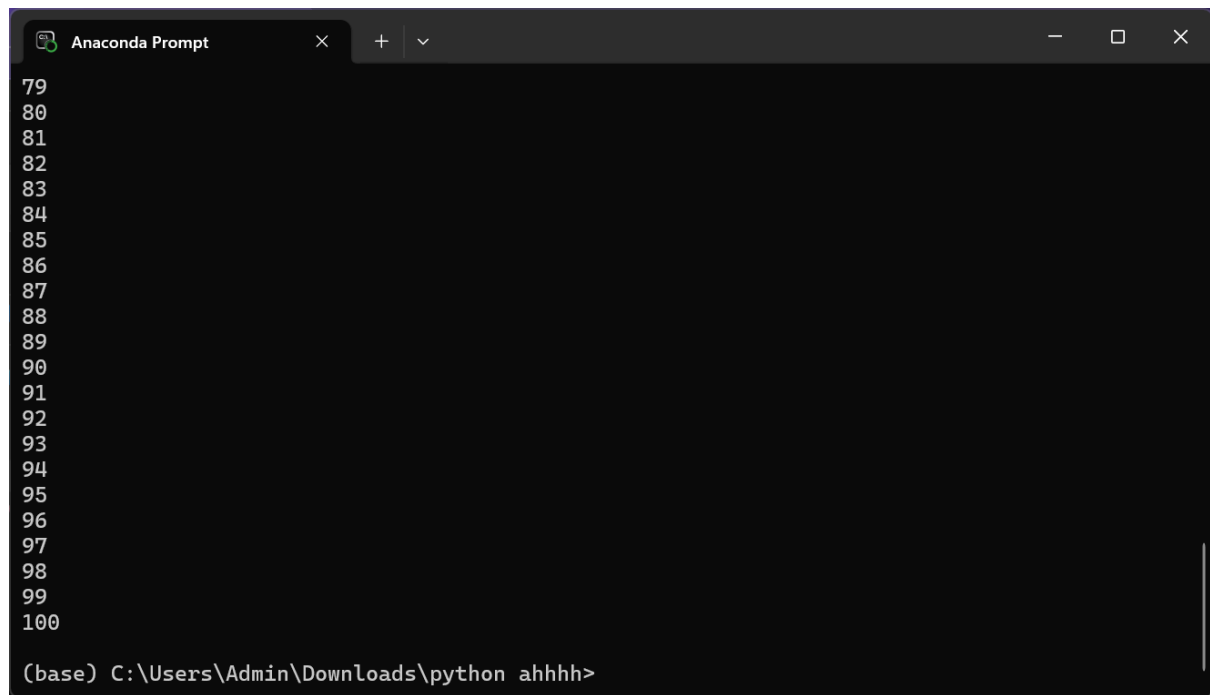
```
    i = 1
```

```
    while i <= 100:
```

```
        print(i)
```

```
        i = i + 1
```

```
print_numbers()
```



```
Anaconda Prompt
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

(base) C:\Users\Admin\Downloads\python_ahhhh>
```

3

```
def print_odd_numbers():
```

```
    i = 1
```

```
    while i < 100:
```

```
        if i % 2 != 0:
```

```
            print(i)
```

```
            i = i + 1
```

```
print_odd_numbers()
```



```
Anaconda Prompt
57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99

(base) C:\Users\Admin\Downloads\python_ahhhh>
```

4

```
def print_even_or_odd():
```

```
    i = 0
```

```
    while i <= 100:
```

```
        if i % 2 == 0:
```

```
            print(i, "is even")
```

```
        else:
```

```
            print(i, "is odd")
```

```
        i = i + 1
```

```
print_even_or_odd()
```

```
Anaconda Prompt
79 is odd
80 is even
81 is odd
82 is even
83 is odd
84 is even
85 is odd
86 is even
87 is odd
88 is even
89 is odd
90 is even
91 is odd
92 is even
93 is odd
94 is even
95 is odd
96 is even
97 is odd
98 is even
99 is odd
100 is even

(base) C:\Users\Admin\Downloads\python_ahhhh>
```

5

```
def sum_of_numbers():
    i = 0
    total = 0
    while i <= 100:
        total = total + i
        i = i + 1
    print("The sum of all natural numbers from 0 to 100 is", total)

sum_of_numbers()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p4_t5.py
The sum of all natural numbers from 0 to 100 is 5050

(base) C:\Users\Admin\Downloads\python ahhhh>
```

6

```
def sum_even_and_odd():
    i = 0
    sum_even = 0
    sum_odd = 0
    while i <= 100:
        if i % 2 == 0:
            sum_even = sum_even + i
        else:
            sum_odd = sum_odd + i
        i = i + 1
    print("The sum of all even numbers from 0 to 100 is", sum_even)
    print("The sum of all odd numbers from 0 to 100 is", sum_odd)

sum_even_and_odd()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p4_t6.py
The sum of all even numbers from 0 to 100 is 2550
The sum of all odd numbers from 0 to 100 is 2500
(base) C:\Users\Admin\Downloads\python ahhhh>
```

7

```
def divisible_by_5_or_7():
    i = 0
    while i <= 100:
        if i % 5 == 0 or i % 7 == 0:
            print(i)
        i = i + 1

divisible_by_5_or_7()
```

```
Anaconda Prompt
35
40
42
45
49
50
55
56
60
63
65
70
75
77
80
84
85
90
91
95
98
100

(base) C:\Users\Admin\Downloads\python_ahhhh>
```

8

```
def calculate_factorial():
    number = int(input("Enter an integer: "))
    if number < 0:
        print("Factorial is not defined for negative numbers.")
        return
    factorial = 1
    i = 1
    while i <= number:
        factorial = factorial * i
        i = i + 1
    print("The factorial of", number, "is", factorial)

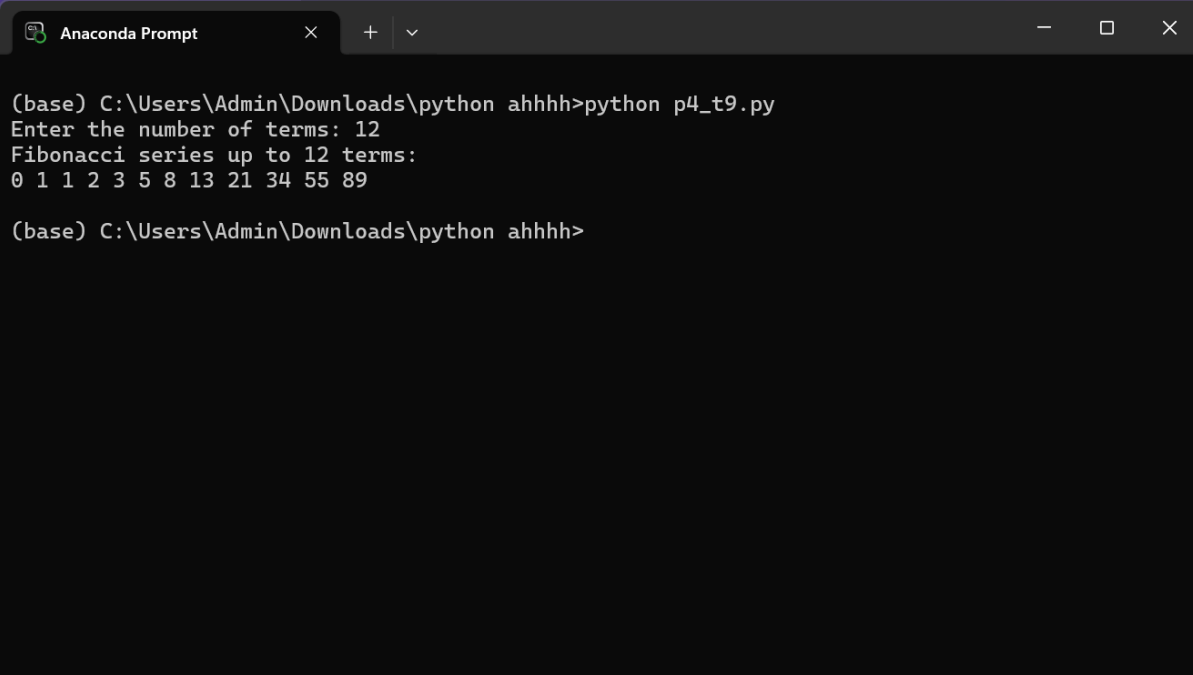
calculate_factorial()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p4_t8.py
Enter an integer: 12
The factorial of 12 is 479001600
(base) C:\Users\Admin\Downloads\python ahhhh>
```

9

```
def generate_fibonacci():
    n = int(input("Enter the number of terms: "))
    if n <= 0:
        print("Please enter a positive integer.")
        return
    a, b = 0, 1
    i = 0
    print("Fibonacci series up to", n, "terms:")
    while i < n:
        print(a, end=" ")
        temp = a + b
        a = b
        b = temp
        i = i + 1
    print()
```

generate_fibonacci()



```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p4_t9.py
Enter the number of terms: 12
Fibonacci series up to 12 terms:
0 1 1 2 3 5 8 13 21 34 55 89

(base) C:\Users\Admin\Downloads\python ahhhh>
```

10

def is_prime_while():

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

 if n <= 1:

 print(n, "is not prime")

 return

 i = 2

 while i < n:

 if n % i == 0:

 print(n, "is not prime")

 return

 i = i + 1

 print(n, "is prime")

is_prime_while()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p4_t10.py
Enter a number: 12
12 is not prime

(base) C:\Users\Admin\Downloads\python ahhhh>
```

Practical 5:

Question:

1. Write a Python program that takes a decimal number as input and converts it to its binary representation using a while loop.
2. Write a Python program that takes a binary number as input and converts it to its decimal representation using a while loop.
3. Write a Python program that takes a hexadecimal number as input and converts it to its binary representation using a while loop.
4. Write a Python program that takes a binary number as input and converts it to its hexadecimal representation using a while loop.
5. Write a Python program that takes an octal number as input and converts it to its binary representation using a for loop and a mapping dictionary.
6. Write a Python program that takes a binary number as input and converts it to its octal representation using a while loop.
7. Write a Python program that takes a decimal number as input and converts it to its binary representation using a for loop and bitwise operations.
8. Write a Python program that takes a binary number as input and converts it to its decimal representation using a for loop.
9. Write a Python program that takes a hexadecimal number as input and converts it to its binary representation using a for loop and a mapping dictionary.
10. Write a Python program that takes a binary number as input and converts it to its hexadecimal representation using a for loop and a mapping dictionary, ensuring proper padding.
11. Write a Python program that takes an octal number as input and converts it to its binary representation using a for loop and a mapping dictionary.

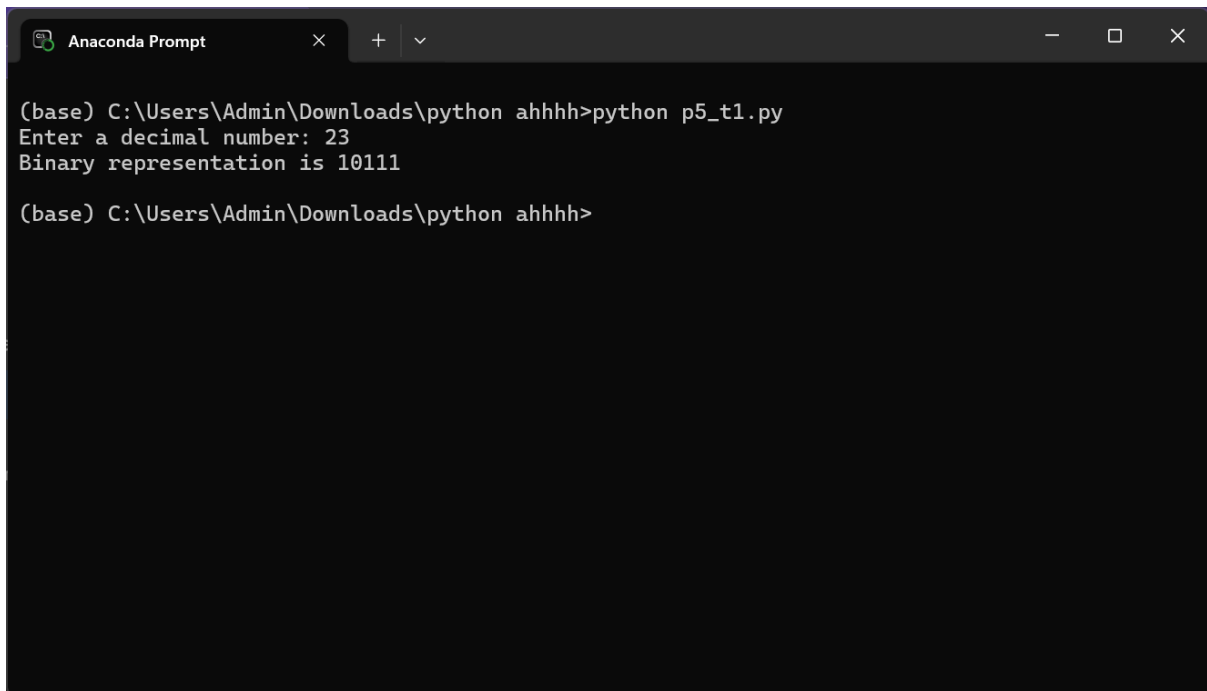
12. Write a Python program that takes a binary number as input and converts it to its octal representation using a for loop and a mapping dictionary, ensuring proper padding.

Code:

1

```
def decimal_to_binary():  
    num = int(input("Enter a decimal number: "))  
    binary = ""  
    while num > 0:  
        binary = str(num % 2) + binary  
        num = num // 2  
    print("Binary representation is", binary)
```

decimal_to_binary()



```
Anaconda Prompt  
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t1.py  
Enter a decimal number: 23  
Binary representation is 10111  
(base) C:\Users\Admin\Downloads\python ahhhh>
```

2

```
def binary_to_decimal():  
    binary = input("Enter a binary number: ")
```

```
decimal = 0

i = 0

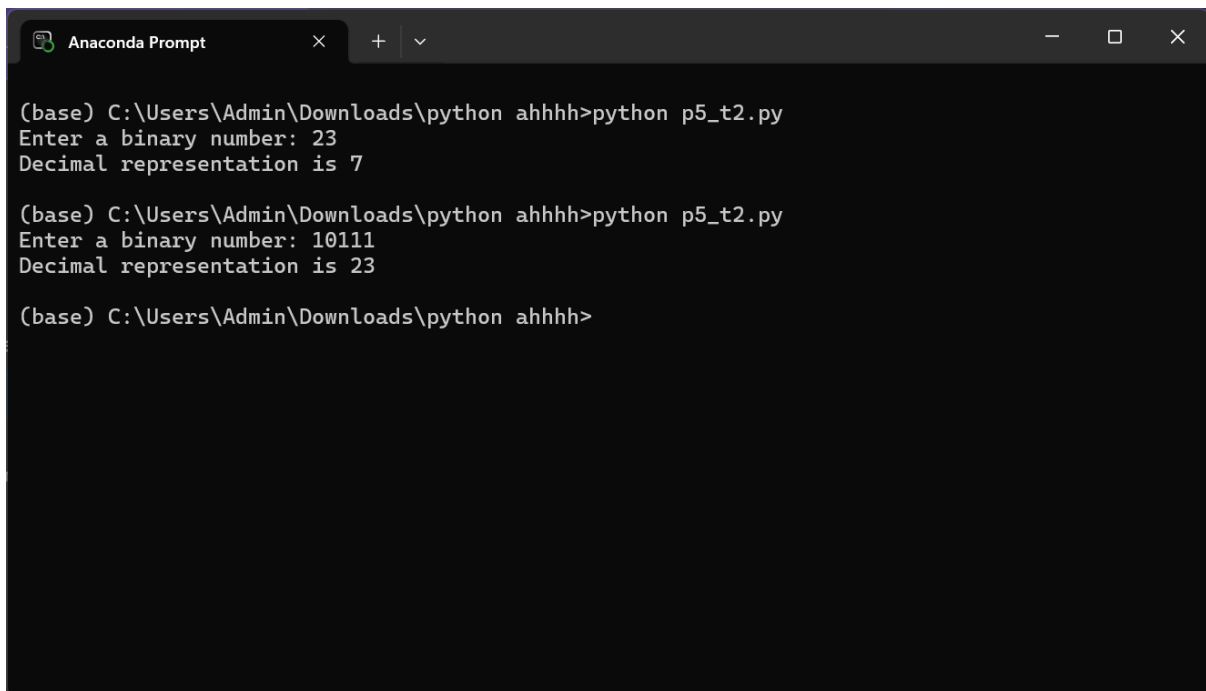
while i < len(binary):

    decimal = decimal * 2 + int(binary[i])

    i = i + 1

print("Decimal representation is", decimal)
```

binary_to_decimal()



```
Anaconda Prompt

(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t2.py
Enter a binary number: 23
Decimal representation is 7

(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t2.py
Enter a binary number: 10111
Decimal representation is 23

(base) C:\Users\Admin\Downloads\python ahhhh>
```

3

```
def hex_to_binary():

    hex_num = input("Enter a hexadecimal number: ")

    binary = ""

    i = 0

    while i < len(hex_num):

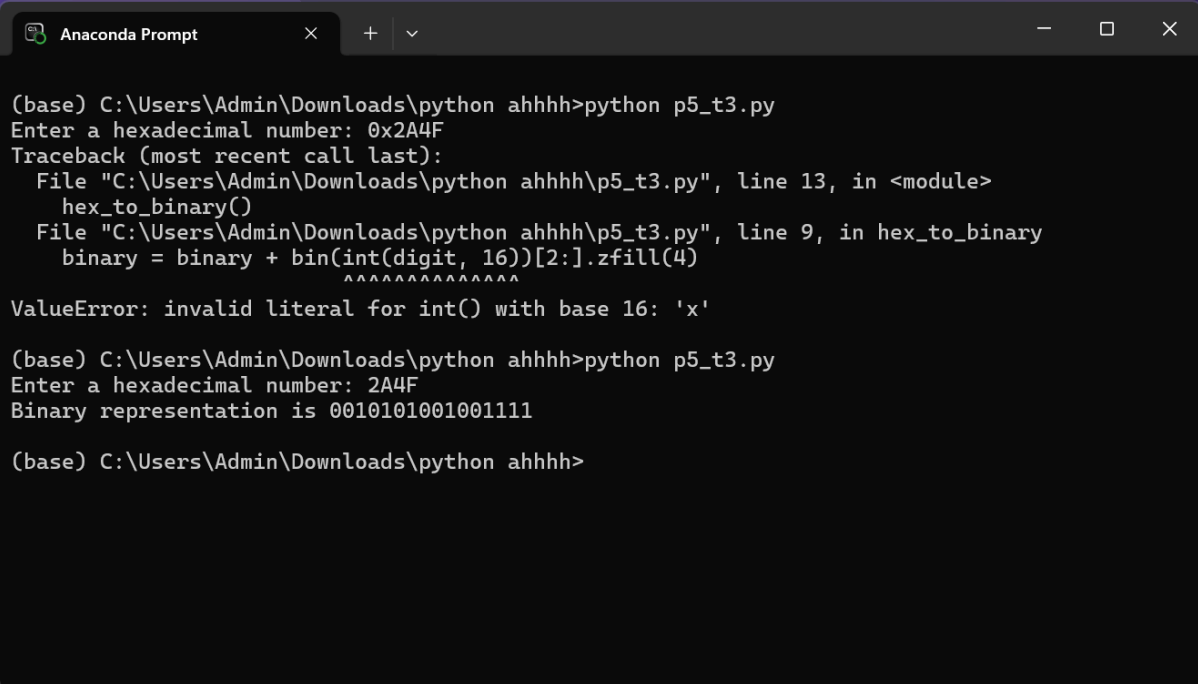
        digit = hex_num[i]

        binary = binary + bin(int(digit, 16))[2:].zfill(4)

        i = i + 1
```

```
print("Binary representation is", binary)
```

```
hex_to_binary()
```



```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t3.py
Enter a hexadecimal number: 0x2A4F
Traceback (most recent call last):
  File "C:\Users\Admin\Downloads\python ahhhh\p5_t3.py", line 13, in <module>
    hex_to_binary()
  File "C:\Users\Admin\Downloads\python ahhhh\p5_t3.py", line 9, in hex_to_binary
    binary = binary + bin(int(digit, 16))[2:].zfill(4)
                        ^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 16: 'x'

(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t3.py
Enter a hexadecimal number: 2A4F
Binary representation is 0010101001001111

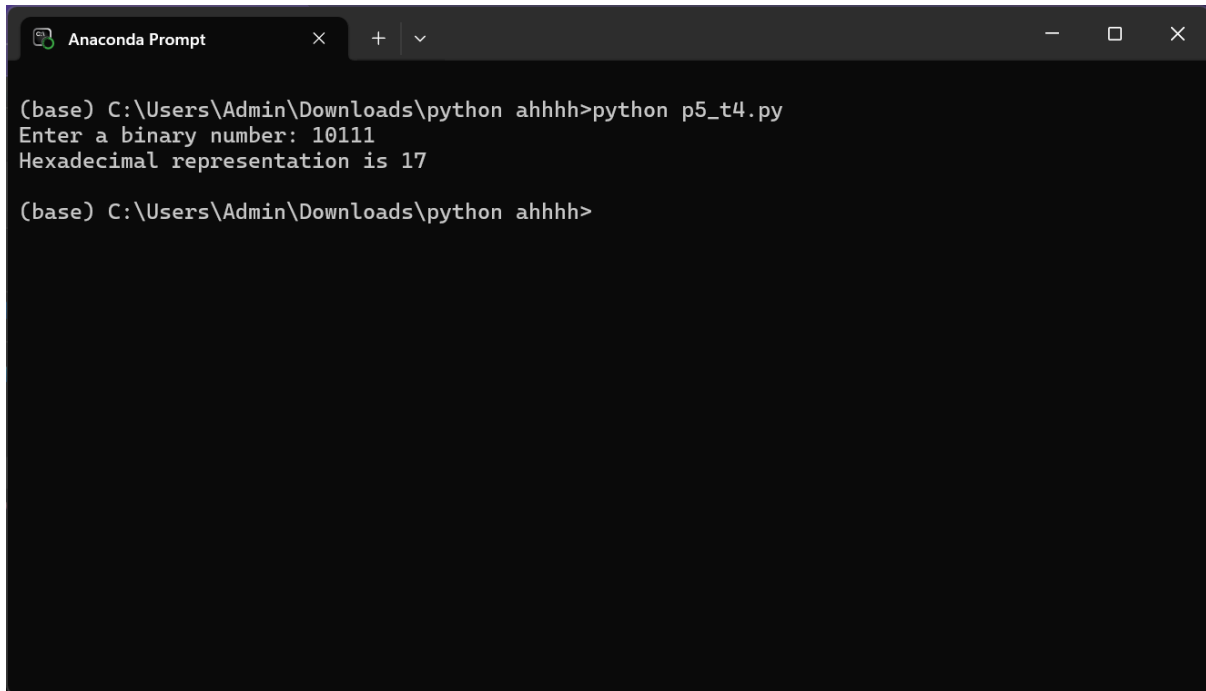
(base) C:\Users\Admin\Downloads\python ahhhh>
```

4

```
def binary_to_hex():
    binary = input("Enter a binary number: ")
    decimal = 0
    i = 0
    while i < len(binary):
        decimal = decimal * 2 + int(binary[i])
        i = i + 1
    hex_num = ""
    while decimal > 0:
        remainder = decimal % 16
        if remainder < 10:
            hex_num = str(remainder) + hex_num
        else:
```

```
hex_num = chr(55 + remainder) + hex_num  
decimal = decimal // 16  
print("Hexadecimal representation is", hex_num)
```

binary_to_hex()

A screenshot of an Anaconda Prompt terminal window. The window title is "Anaconda Prompt". The terminal shows the following text:

```
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t4.py  
Enter a binary number: 10111  
Hexadecimal representation is 17  
(base) C:\Users\Admin\Downloads\python ahhhh>
```

5

```
def octal_to_binary():  
    octal = input("Enter an octal number: ")  
    binary_map = {  
        '0': '000',  
        '1': '001',  
        '2': '010',  
        '3': '011',  
        '4': '100',  
        '5': '101',  
        '6': '110',  
        '7': '111'
```

```

}

binary = ""

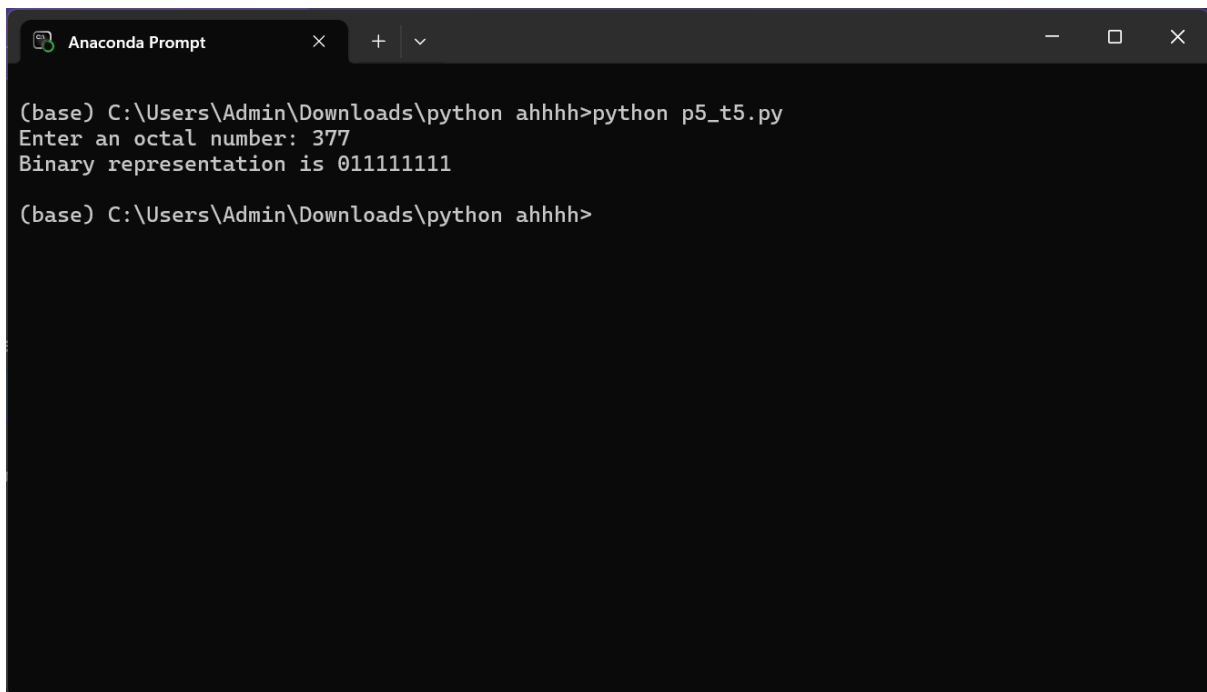
for digit in octal:

    binary = binary + binary_map[digit]

print("Binary representation is", binary)

```

octal_to_binary()



The screenshot shows an Anaconda Prompt window with a dark background. The terminal text is as follows:

```

(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t5.py
Enter an octal number: 377
Binary representation is 011111111

(base) C:\Users\Admin\Downloads\python ahhhh>

```

6

```

def binary_to_octal():

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

    decimal = 0

    i = 0

    while i < len(binary):

        decimal = decimal * 2 + int(binary[i])

        i = i + 1

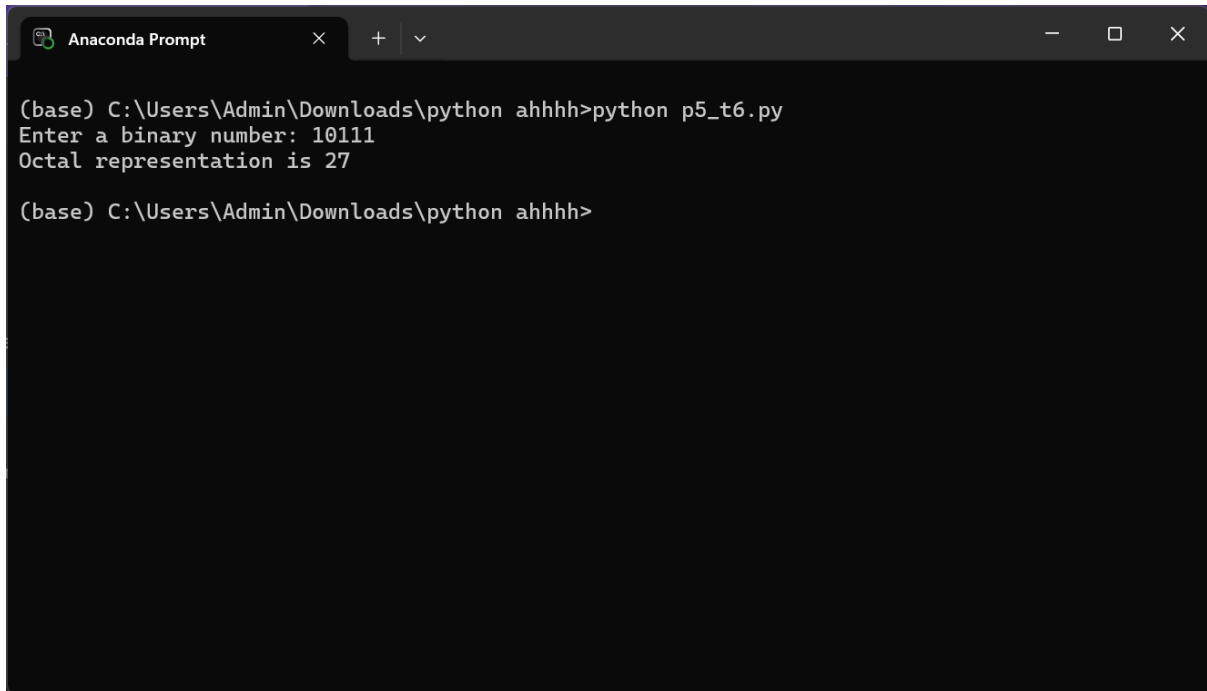
    octal = ""

    while decimal > 0:

```

```
octal = str(decimal % 8) + octal  
  
decimal = decimal // 8  
  
print("Octal representation is", octal)
```

binary_to_octal()

A screenshot of an Anaconda Prompt window. The title bar shows 'Anaconda Prompt' with standard window controls. The terminal content shows a command prompt at 'C:\Users\Admin\Downloads\python ahhhh>' where the command 'python p5_t6.py' has been entered. The program prompts 'Enter a binary number: 10111' and outputs 'Octal representation is 27'. The prompt returns to 'C:\Users\Admin\Downloads\python ahhhh>'.

7

```
def decimal_to_binary_bitwise():  
    num = int(input("Enter a decimal number: "))  
    binary = ""  
    for i in range(31, -1, -1):  
        bit = (num >> i) & 1  
        binary = binary + str(bit)  
    binary = binary.lstrip("0") # Remove leading zeros  
    print("Binary representation is", binary)
```

decimal_to_binary_bitwise()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t7.py
Enter a decimal number: 34
Binary representation is 100010
(base) C:\Users\Admin\Downloads\python ahhhh>
```

8

```
def binary_to_decimal_for():
    binary = input("Enter a binary number: ")
    decimal = 0
    for i in range(len(binary)):
        decimal = decimal * 2 + int(binary[i])
    print("Decimal representation is", decimal)
```

```
binary_to_decimal_for()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t8.py
Enter a binary number: 10111
Decimal representation is 23

(base) C:\Users\Admin\Downloads\python ahhhh>
```

9

```
def hex_to_binary_for():
    hex_num = input("Enter a hexadecimal number: ")
    binary_map = {
        '0': '0000', '1': '0001', '2': '0010', '3': '0011',
        '4': '0100', '5': '0101', '6': '0110', '7': '0111',
        '8': '1000', '9': '1001', 'A': '1010', 'B': '1011',
        'C': '1100', 'D': '1101', 'E': '1110', 'F': '1111'
    }
    binary = ""
    for digit in hex_num:
        binary = binary + binary_map[digit.upper()]
    print("Binary representation is", binary)

hex_to_binary_for()
```



```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t9.py
Enter a hexadecimal number: 2A4F
Binary representation is 0010101001001111
(base) C:\Users\Admin\Downloads\python ahhhh>
```

10

```
def binary_to_hex_for():
    binary = input("Enter a binary number: ")
    binary = binary.zfill((len(binary) + 3) // 4 * 4) # Ensure proper padding
    hex_map = {
        '0000': '0', '0001': '1', '0010': '2', '0011': '3',
        '0100': '4', '0101': '5', '0110': '6', '0111': '7',
        '1000': '8', '1001': '9', '1010': 'A', '1011': 'B',
        '1100': 'C', '1101': 'D', '1110': 'E', '1111': 'F'
    }
    hex_num = ""
    for i in range(0, len(binary), 4):
        hex_num = hex_num + hex_map[binary[i:i+4]]
    print("Hexadecimal representation is", hex_num)

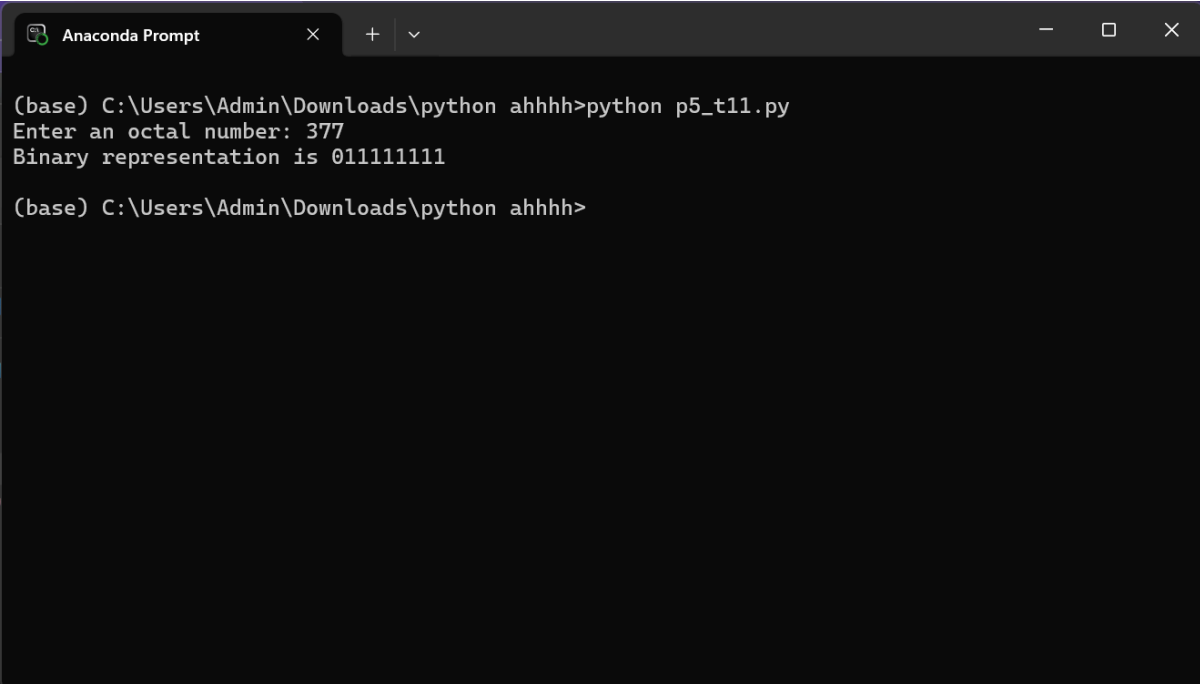
binary_to_hex_for()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t10.py
Enter a binary number: 10111
Hexadecimal representation is 17
(base) C:\Users\Admin\Downloads\python ahhhh>
```

11

```
def octal_to_binary_for():
    octal = input("Enter an octal number: ")
    binary_map = {
        '0': '000',
        '1': '001',
        '2': '010',
        '3': '011',
        '4': '100',
        '5': '101',
        '6': '110',
        '7': '111'
    }
    binary = ""
    for digit in octal:
        binary = binary + binary_map[digit]
    print("Binary representation is", binary)
```

octal_to_binary_for()

A screenshot of an Anaconda Prompt window. The title bar shows 'Anaconda Prompt' with standard window controls. The terminal content shows a command prompt at '(base) C:\Users\Admin\Downloads\python ahhhh>' where the user has run 'python p5_t11.py'. The program prompts 'Enter an octal number: 377' and outputs 'Binary representation is 011111111'. The prompt returns to '(base) C:\Users\Admin\Downloads\python ahhhh>'.

```
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t11.py
Enter an octal number: 377
Binary representation is 011111111
(base) C:\Users\Admin\Downloads\python ahhhh>
```

12

def binary_to_octal_for():

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

 binary = binary.zfill((len(binary) + 2) // 3 * 3) # Ensure proper padding

 octal_map = {

 '000': '0', '001': '1', '010': '2', '011': '3',

 '100': '4', '101': '5', '110': '6', '111': '7'

 }

 octal = ""

 for i in range(0, len(binary), 3):

 octal = octal + octal_map[binary[i:i+3]]

 print("Octal representation is", octal)

binary_to_octal_for()

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p5_t12.py
Enter a binary number: 10111
Octal representation is 27

(base) C:\Users\Admin\Downloads\python ahhhh>
```

Practical 6:

Question:

1. Write a Python function that takes the string "Hello World" and prints each character on a new line.
2. Write a Python function that takes the string "Hello World" and prints the index of each vowel in the string.
3. Write a Python function that takes the string "Hello World" and prints the character immediately to the left of each vowel in the string.
4. Write a Python function that takes a word as input and checks if it's a palindrome using loops. The function should print whether the word is a palindrome or not.
5. Write a Python function that takes two words as input and checks if they are anagrams. The function should print whether the words are anagrams or not. If the words are of different lengths, it should print an appropriate message.
6. Write a Python function that takes a word as input and removes repeated characters from it. The function should print the resulting string.
7. Write a Python function that takes a word as input and prints all possible substrings of that word.
8. Write a Python function that takes a word as input and prints all substrings of length 3.
9. Write a Python function that takes a list of strings as input (the list length and elements are provided by the user) and prints all strings with a length of 3 or more.
10. Write a Python function that takes a list of strings as input (e.g., ["IITRAM", "is", "in", "Gujarat"]) and prints the strings in order of their length, from shortest to longest.
11. Write a Python function that takes a 3x3 matrix as input and prints the elements row-wise and column-wise.
12. Write a Python function that takes a 3x3 matrix as input and checks if it's an identity matrix. The function should print "True" if it is, and "False" otherwise.

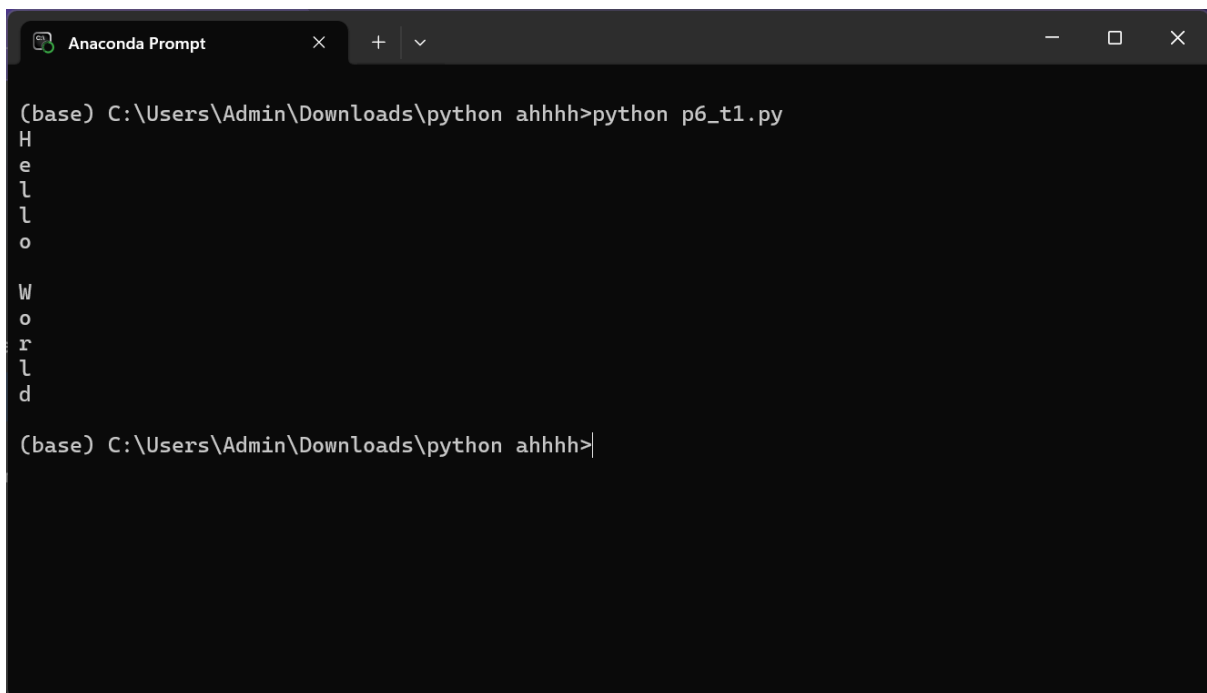
13. Write a Python function that takes a list of strings representing numbers (e.g., ["5", "1", "3", "4", "2"]) and prints them in order of their length (which will result in sorted order in this specific case).
14. Write a Python function that takes a list of strings as input (the list length and elements are provided by the user) and an element to search for. The function should print the index of the element if found, or an appropriate message if not found.

Code:

1

```
def print_characters():  
    string = "Hello World"  
    for char in string:  
        print(char)
```

print_characters()



```
Anaconda Prompt  
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t1.py  
H  
e  
l  
l  
o  
  
W  
o  
r  
l  
d  
  
(base) C:\Users\Admin\Downloads\python ahhhh>
```

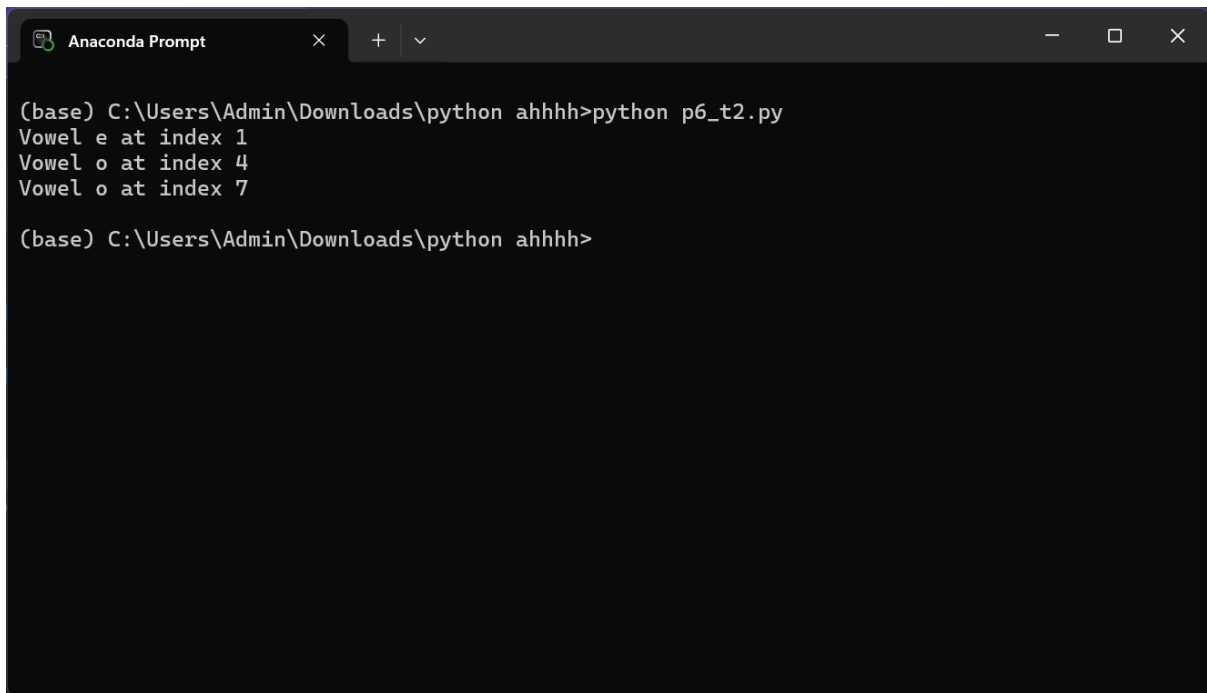
2

```
def print_vowel_indices():  
    string = "Hello World"
```

```
vowels = "aeiouAEIOU"

for i in range(len(string)):
    if string[i] in vowels:
        print("Vowel", string[i], "at index", i)
```

```
print_vowel_indices()
```



```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t2.py
Vowel e at index 1
Vowel o at index 4
Vowel o at index 7
(base) C:\Users\Admin\Downloads\python ahhhh>
```

3

```
def print_left_of_vowels():
    string = "Hello World"
    vowels = "aeiouAEIOU"
    for i in range(1, len(string)):
        if string[i] in vowels:
            print("Character to the left of", string[i], "is", string[i - 1])

print_left_of_vowels()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t3.py
Character to the left of e is H
Character to the left of o is l
Character to the left of o is W
(base) C:\Users\Admin\Downloads\python ahhhh>
```

4

```
def check_palindrome():
    word = input("Enter a word: ")
    i = 0
    j = len(word) - 1
    while i < j:
        if word[i] != word[j]:
            print(word, "is not a palindrome")
            return
        i = i + 1
        j = j - 1
    print(word, "is a palindrome")

check_palindrome()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t4.py
Enter a word: husain
husain is not a palindrome
(base) C:\Users\Admin\Downloads\python ahhhh>
```

5

```
def check_anagrams():
    word1 = input("Enter the first word: ")
    word2 = input("Enter the second word: ")
    if len(word1) != len(word2):
        print("The words are of different lengths, so they cannot be anagrams.")
        return
    if sorted(word1) == sorted(word2):
        print("The words are anagrams.")
    else:
        print("The words are not anagrams.")

check_anagrams()
```



```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t5.py
Enter the first word: racecar
Enter the second word: choco
The words are of different lengths, so they cannot be anagrams.
(base) C:\Users\Admin\Downloads\python ahhhh>
```

6

```
def remove_repeated_characters():
    word = input("Enter a word: ")
    result = ""
    for char in word:
        if char not in result:
            result = result + char
    print("String after removing repeated characters:", result)

remove_repeated_characters()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t6.py
Enter a word: husain
String after removing repeated characters: husain
(base) C:\Users\Admin\Downloads\python ahhhh>
```

7

```
def print_substrings():
    word = input("Enter a word: ")
    for i in range(len(word)):
        for j in range(i + 1, len(word) + 1):
            print(word[i:j])

print_substrings()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python_ahhhh>python p6_t7.py
Enter a word: wow
w
wo
wow
o
ow
w
(base) C:\Users\Admin\Downloads\python_ahhhh>
```

8

```
def print_substrings_length_3():
    word = input("Enter a word: ")
    for i in range(len(word) - 2):
        print(word[i:i+3])
```

```
print_substrings_length_3()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python_ahhhh>python p6_t8.py
Enter a word: wonderful
won
ond
nde
der
erf
rfu
ful
(base) C:\Users\Admin\Downloads\python_ahhhh>
```

9

```
def print_long_strings():
    n = int(input("Enter the number of strings: "))
    strings = []
    for i in range(n):
        strings.append(input("Enter string " + str(i + 1) + ": "))
    for string in strings:
        if len(string) >= 3:
            print(string)

print_long_strings()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t9.py
Enter the number of strings: 3
Enter string 1: wow
Enter string 2: hehehe
Enter string 3: lol
wow
hehehe
lol

(base) C:\Users\Admin\Downloads\python ahhhh>
```

10

```
def sort_strings_by_length():
    n = int(input("Enter the number of strings: "))
    strings = []
    for i in range(n):
        strings.append(input("Enter string " + str(i + 1) + ": "))
    strings.sort(key=len)
    print("Strings sorted by length:", strings)
```

```
sort_strings_by_length()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t10.py
Enter the number of strings: 2
Enter string 1: heheh
Enter string 2: wonderful
Strings sorted by length: ['heheh', 'wonderful']

(base) C:\Users\Admin\Downloads\python ahhhh>
```

11

```
def print_matrix():
    matrix = []
    print("Enter the elements of the 3x3 matrix row by row:")
    for i in range(3):
        row = list(map(int, input("Enter row " + str(i + 1) + ": ").split()))
        matrix.append(row)

    print("Row-wise elements:")
    for row in matrix:
        print(row)

    print("Column-wise elements:")
    for j in range(3):
        for i in range(3):
            print(matrix[i][j], end=" ")
        print()
```

```
print_matrix()
```

```
# 12
```

```
def is_identity_matrix():
```

```
    matrix = []
```

```
    print("Enter the elements of the 3x3 matrix row by row:")
```

```
    for i in range(3):
```

```
        row = list(map(int, input("Enter row " + str(i + 1) + ": ").split()))
```

```
        matrix.append(row)
```

```
    for i in range(3):
```

```
        for j in range(3):
```

```
            if i == j and matrix[i][j] != 1:
```

```
                print("False")
```

```
                return
```

```
            elif i != j and matrix[i][j] != 0:
```

```
                print("False")
```

```
                return
```

```
    print("True")
```

```
is_identity_matrix()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p6_t12.py
Enter the elements of the 3x3 matrix row by row:
Enter row 1: 123
Enter row 2: 234
Enter row 3: 432
False

(base) C:\Users\Admin\Downloads\python ahhhh>
```

13

```
def sort_numbers_by_length():
    n = int(input("Enter the number of strings: "))
    numbers = []
    for i in range(n):
        numbers.append(input("Enter number " + str(i + 1) + ": "))
    numbers.sort(key=len)
    print("Numbers sorted by length:", numbers)
```

```
sort_numbers_by_length()
```

14

```
def search_element():
    n = int(input("Enter the number of strings: "))
    strings = []
```



```

for i in range(n):
    strings.append(input("Enter string " + str(i + 1) + ": "))
element = input("Enter the element to search for: ")
if element in strings:
    print("Element found at index", strings.index(element))
else:
    print("Element not found in the list.")

search_element()

```

Practical 7:

Question:

1. Write a Python program that defines three classes: vehicle, electric, and car. The vehicle class should have attributes for brand, model, and year, and a method to display these attributes. The electric class should have an attribute for battery capacity and a method to display it. The car class should inherit from both vehicle and electric and have a method to display all its attributes. Create instances of these classes and demonstrate their usage.
2. Write a Python program that defines a vehicle class with public, protected, and private attributes for brand, model, and year, respectively. Create a car class that inherits from vehicle and electric classes, and demonstrate how to access and print these attributes from an instance of the car class. Explain the concept of access modifiers in Python.
3. Write a Python program that simulates a bank account system using OOP. Define a BankAccount base class with attributes for account number, account type (protected), and balance (private). Implement methods to get and set the balance (using protected methods). Create subclasses Deposit, Withdraw, GetBalance, AccountType, savings_account, and AddInterest to handle deposit, withdrawal, balance display, account type display, savings account creation, and interest addition, respectively. Demonstrate the functionality by creating an instance of the savings_account class and performing various operations.

Code:

```
# 1
```

```

class Vehicle:
    def __init__(self, brand, model, year):

```

```
self.brand = brand
```

```
self.model = model
```

```
self.year = year
```

```
def display_vehicle(self):
```

```
    print("Brand:", self.brand)
```

```
    print("Model:", self.model)
```

```
    print("Year:", self.year)
```

```
class Electric:
```

```
    def __init__(self, battery_capacity):
```

```
        self.battery_capacity = battery_capacity
```

```
    def display_battery(self):
```

```
        print("Battery Capacity:", self.battery_capacity, "kWh")
```

```
class Car(Vehicle, Electric):
```

```
    def __init__(self, brand, model, year, battery_capacity):
```

```
        Vehicle.__init__(self, brand, model, year)
```

```
        Electric.__init__(self, battery_capacity)
```

```
    def display_car(self):
```

```
        self.display_vehicle()
```

```
        self.display_battery()
```

```
car = Car("Tesla", "Model S", 2022, 100)
```

```
car.display_car()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p7_t1.py
Brand: Tesla
Model: Model S
Year: 2022
Battery Capacity: 100 kWh
(base) C:\Users\Admin\Downloads\python ahhhh>
```

2

```
class Vehicle:
```

```
    def __init__(self, brand, model, year):
```

```
        self.brand = brand
```

```
        self._model = model
```

```
        self.__year = year
```

```
    def get_year(self):
```

```
        return self.__year
```

```
class Electric:
```

```
    def __init__(self, battery_capacity):
```

```
        self.battery_capacity = battery_capacity
```

```
class Car(Vehicle, Electric):
```

```
    def __init__(self, brand, model, year, battery_capacity):
```

```
        Vehicle.__init__(self, brand, model, year)
```

```
Electric.__init__(self, battery_capacity)
```

```
def display_attributes(self):
```

```
    print("Brand (Public):", self.brand)
```

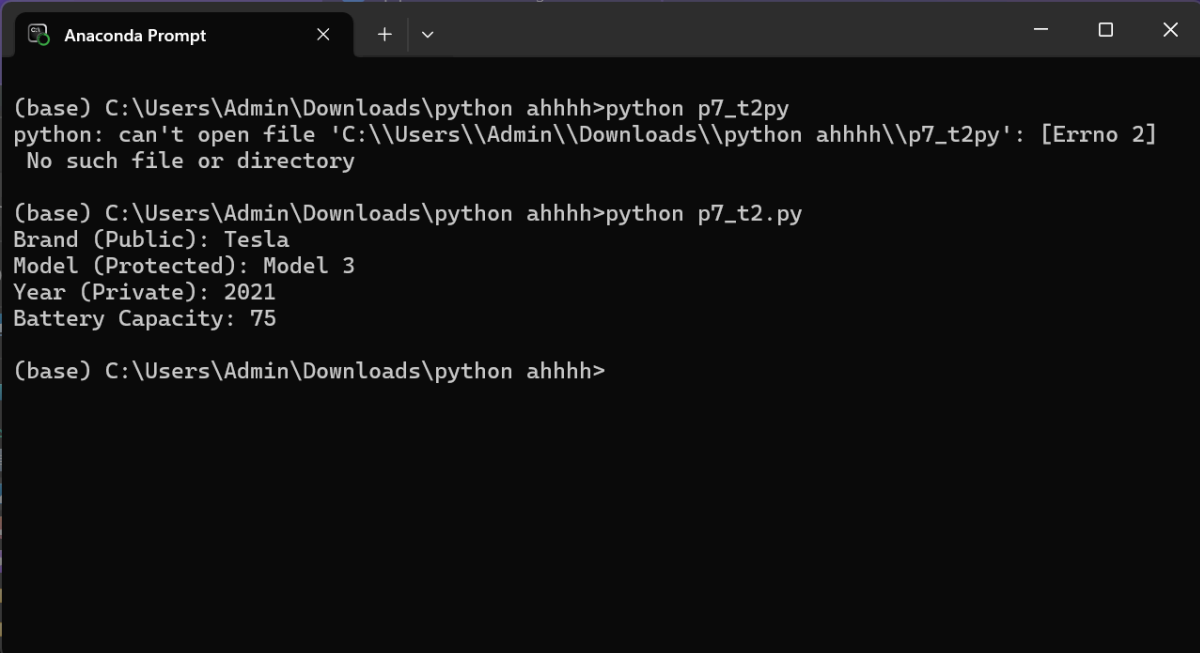
```
    print("Model (Protected):", self._model)
```

```
    print("Year (Private):", self.get_year())
```

```
    print("Battery Capacity:", self.battery_capacity)
```

```
car = Car("Tesla", "Model 3", 2021, 75)
```

```
car.display_attributes()
```



```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p7_t2py
python: can't open file 'C:\\Users\\Admin\\Downloads\\python ahhhh\\p7_t2py': [Errno 2]
No such file or directory

(base) C:\Users\Admin\Downloads\python ahhhh>python p7_t2.py
Brand (Public): Tesla
Model (Protected): Model 3
Year (Private): 2021
Battery Capacity: 75

(base) C:\Users\Admin\Downloads\python ahhhh>
```

3

```
class Vehicle:
```

```
    def __init__(self, brand, model, year):
```

```
        self.brand = brand
```

```
        self._model = model
```

```
        self.__year = year
```

```
def get_year(self):  
    return self.__year
```

```
class Electric:
```

```
    def __init__(self, battery_capacity):  
        self.battery_capacity = battery_capacity
```

```
class Car(Vehicle, Electric):
```

```
    def __init__(self, brand, model, year, battery_capacity):  
        Vehicle.__init__(self, brand, model, year)  
        Electric.__init__(self, battery_capacity)
```

```
    def display_attributes(self):  
        print("Brand (Public):", self.brand)  
        print("Model (Protected):", self._model)  
        print("Year (Private):", self.get_year())  
        print("Battery Capacity:", self.battery_capacity)
```

```
car = Car("Tesla", "Model 3", 2021, 75)
```

```
car.display_attributes()
```

```
Anaconda Prompt
(base) C:\Users\Admin\Downloads\python ahhhh>python p7_t3.py
Brand (Public): Tesla
Model (Protected): Model 3
Year (Private): 2021
Battery Capacity: 75
(base) C:\Users\Admin\Downloads\python ahhhh>
```

Practical 8:

Question:

1. Write a Python program using Tkinter to create a login window with fields for username and password. Implement a login function that checks if the username is "Train" and the password is "123". If the login is successful, hide the login window and display a calculator window.
2. Extend the previous program to create a calculator window with buttons for digits (0-9), addition (+), and equals (=). Implement functions to display the pressed digits on a label and calculate the result when the equals button is pressed. Include error handling for invalid calculations.
3. Explain the purpose and functionality of the Frame, Label, Entry, and Button widgets in Tkinter, as used in the provided code. Provide examples of how their attributes (e.g., width, height, font, borderwidth, bg, fg, text, command) are used to customize the GUI.

Code:

1

```
import tkinter as tk
```

```
from tkinter import messagebox
```

```
def login():  
    username = username_entry.get()  
    password = password_entry.get()  
    if username == "Train" and password == "123":  
        login_window.destroy()  
        open_calculator()  
    else:  
        messagebox.showerror("Login Failed", "Invalid username or password")
```

```
def open_calculator():  
    calc_window = tk.Tk()  
    calc_window.title("Calculator")  
    calc_window.geometry("300x400")  
    tk.Label(calc_window, text="Calculator", font=("Arial", 16)).pack()  
    calc_window.mainloop()
```

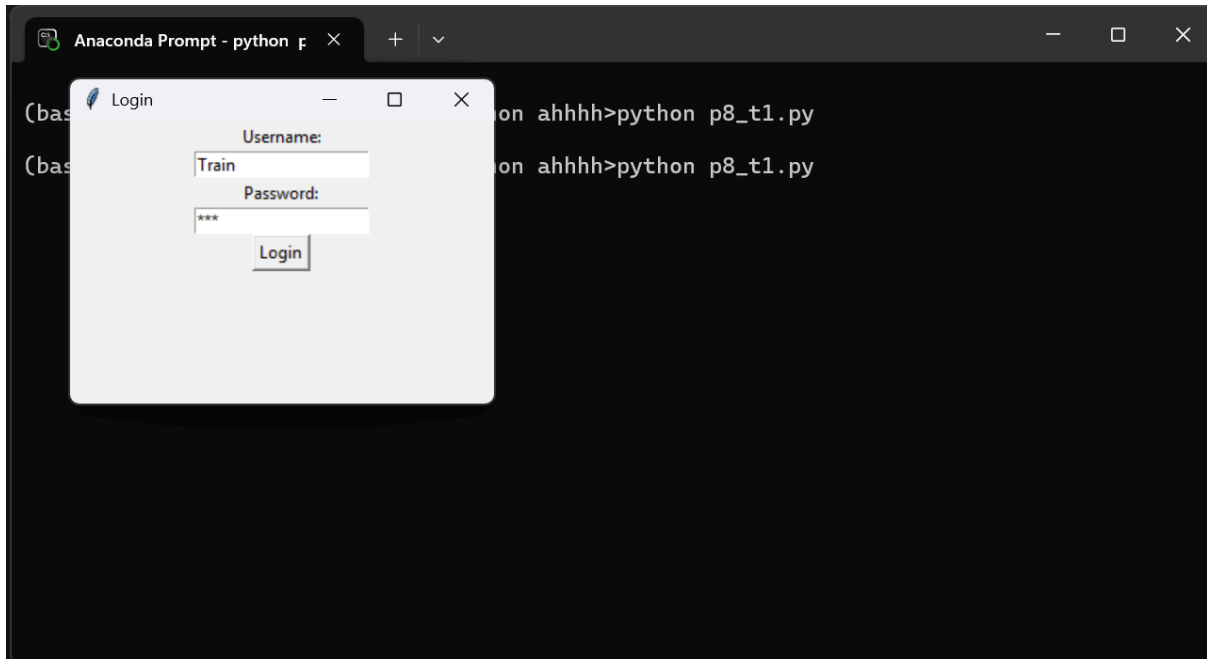
```
login_window = tk.Tk()  
login_window.title("Login")  
login_window.geometry("300x200")
```

```
tk.Label(login_window, text="Username:").pack()  
username_entry = tk.Entry(login_window)  
username_entry.pack()
```

```
tk.Label(login_window, text="Password:").pack()  
password_entry = tk.Entry(login_window, show="*")  
password_entry.pack()
```

```
tk.Button(login_window, text="Login", command=login).pack()
```

```
login_window.mainloop()
```



2

```
import tkinter as tk
```

```
from tkinter import messagebox
```

```
def login():
```

```
    username = username_entry.get()
```

```
    password = password_entry.get()
```

```
    if username == "Train" and password == "123":
```

```
        login_window.destroy()
```

```
        open_calculator()
```

```
    else:
```

```
        messagebox.showerror("Login Failed", "Invalid username or password")
```

```
def open_calculator():
```

```
    def click_button(value):
```



```
current = display_label.cget("text")
display_label.config(text=current + value)
```

```
def calculate():
```

```
    try:
        result = eval(display_label.cget("text"))
        display_label.config(text=str(result))
    except Exception as e:
        display_label.config(text="Error")
```

```
calc_window = tk.Tk()
```

```
calc_window.title("Calculator")
```

```
calc_window.geometry("300x400")
```

```
display_label = tk.Label(calc_window, text="", font=("Arial", 16), bg="white", anchor="e")
```

```
display_label.pack(fill="both", padx=10, pady=10)
```

```
buttons = [
```

```
    ('7', '8', '9', '+'),
```

```
    ('4', '5', '6', ''),
```

```
    ('1', '2', '3', ''),
```

```
    ('0', '=', '', '')
```

```
]
```

```
for row in buttons:
```

```
    frame = tk.Frame(calc_window)
```

```
    frame.pack(expand=True, fill="both")
```

```
    for button in row:
```

```
        if button:
```

```
            tk.Button(frame, text=button, font=("Arial", 14), command=lambda b=button:
click_button(b) if b != '=' else calculate()).pack(side="left", expand=True, fill="both")
```

```
calc_window.mainloop()
```

```
login_window = tk.Tk()
```

```
login_window.title("Login")
```

```
login_window.geometry("300x200")
```

```
tk.Label(login_window, text="Username:").pack()
```

```
username_entry = tk.Entry(login_window)
```

```
username_entry.pack()
```

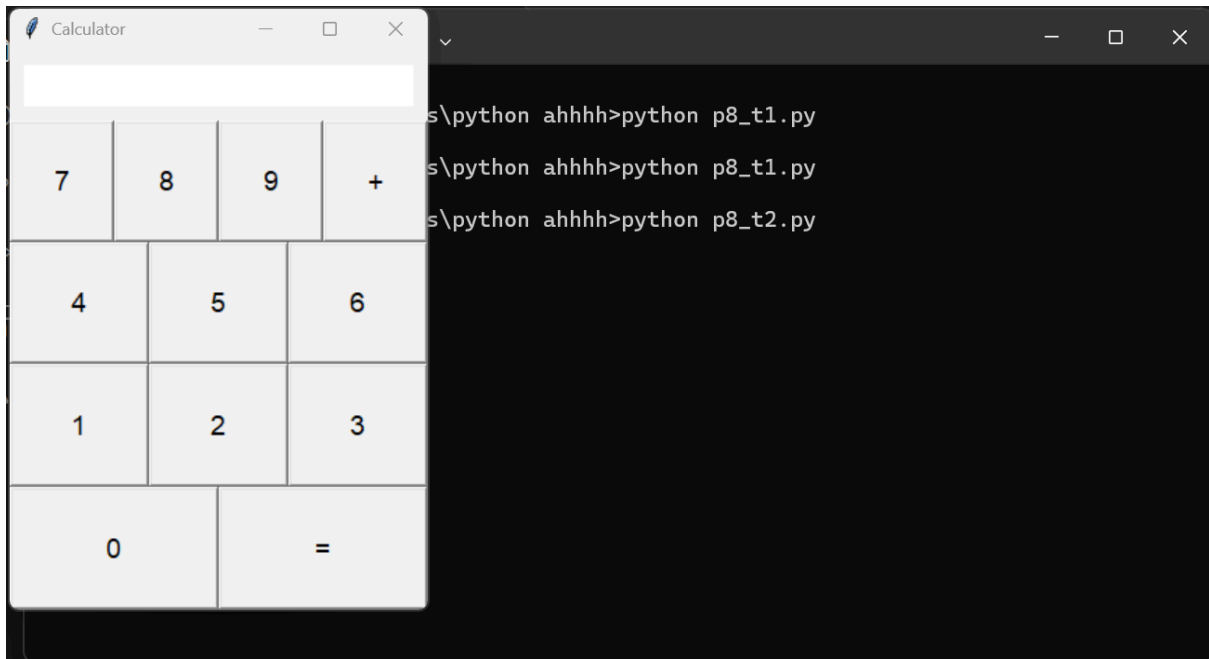
```
tk.Label(login_window, text="Password:").pack()
```

```
password_entry = tk.Entry(login_window, show="*")
```

```
password_entry.pack()
```

```
tk.Button(login_window, text="Login", command=login).pack()
```

```
login_window.mainloop()
```



3

```
import tkinter as tk
```

```
from tkinter import messagebox
```

```
def login():
```

```
    username = username_entry.get()
```

```
    password = password_entry.get()
```

```
    if username == "Train" and password == "123":
```

```
        login_window.destroy()
```

```
        open_calculator()
```

```
    else:
```

```
        messagebox.showerror("Login Failed", "Invalid username or password")
```

```
def open_calculator():
```

```
    def click_button(value):
```

```
        current = display_label.cget("text")
```

```
        display_label.config(text=current + value)
```

```

def calculate():
    try:
        result = eval(display_label.cget("text"))
        display_label.config(text=str(result))
    except Exception as e:
        display_label.config(text="Error")

calc_window = tk.Tk()
calc_window.title("Calculator")
calc_window.geometry("300x400")

display_label = tk.Label(calc_window, text="", font=("Arial", 16), bg="white", anchor="e")
display_label.pack(fill="both", padx=10, pady=10)

buttons = [
    ('7', '8', '9', '+'),
    ('4', '5', '6', '-'),
    ('1', '2', '3', '*'),
    ('0', '=', '/', 'C')
]

for row in buttons:
    frame = tk.Frame(calc_window, bg="lightgray")
    frame.pack(expand=True, fill="both")
    for button in row:
        if button == "C":
            tk.Button(frame, text=button, font=("Arial", 14), bg="red", fg="white",
                      command=lambda: display_label.config(text="")).pack(side="left", expand=True,
fill="both")

```

```
elif button == "=":  
    tk.Button(frame, text=button, font=("Arial", 14), bg="green", fg="white",  
              command=calculate).pack(side="left", expand=True, fill="both")  
else:  
    tk.Button(frame, text=button, font=("Arial", 14),  
              command=lambda b=button: click_button(b)).pack(side="left", expand=True,  
fill="both")  
  
calc_window.mainloop()
```

```
login_window = tk.Tk()  
login_window.title("Login")  
login_window.geometry("300x200")
```

```
tk.Label(login_window, text="Username:", font=("Arial", 12)).pack(pady=5)  
username_entry = tk.Entry(login_window, font=("Arial", 12))  
username_entry.pack(pady=5)
```

```
tk.Label(login_window, text="Password:", font=("Arial", 12)).pack(pady=5)  
password_entry = tk.Entry(login_window, show="*", font=("Arial", 12))  
password_entry.pack(pady=5)
```

```
tk.Button(login_window, text="Login", font=("Arial", 12), bg="blue", fg="white",  
command=login).pack(pady=10)
```

```
login_window.mainloop()
```

