PYTHON PROGRAMMING LAB

MASTER OF COMPUTER APPLICATIONS (MCA)

FIRST YEAR, SEMESTER-II, PAPER-VI

LESSON WRITER

Dr. U Surya Kameswari
Assistant Professor
Department of CS&E
Acharya Nagarjuna University

EDITOR

Dr. Kampa Lavanya
Assistant Professor
Department of CS&E
Acharya Nagarjuna University

ACADEMIC ADVISOR

Dr. Kampa Lavanya
Assistant Professor
Department of CS&E
Acharya Nagarjuna University

DIRECTOR, I/c.

Prof. V. Venkateswarlu
M.A., M.P.S., M.S.W., M.Phil., Ph.D.

CENTRE FOR DISTANCE EDUCATION
ACHARYA NAGARJUNA UNIVERSITY
NAGARJUNA NAGAR 522 510

Ph: 0863-2346222, 2346208
0863- 2346259 (Study Material)
Website www.anucde.info
E-mail: anucdedirector@gmail.com

PYTHON PROGRAMMING LAB

First Edition : 2025

No. of Copies :

This book is exclusively prepared for the use of students of MASTER OF COMPUTER APPLICATIONS (MCA), Centre for Distance Education, Acharya Nagarjuna University and this book is meant for limited circulation only.

Published by:

Prof. V. VENKATESWARLU
Director, I/c
Centre for Distance Education,
Acharya Nagarjuna University

Printed at:

FOREWORD

Since its establishment in 1976, Acharya Nagarjuna University has been forging ahead in the path of progress and dynamism, offering a variety of courses and research contributions. I am extremely happy that by gaining 'A+' grade from the NAAC in the year 2024, Acharya Nagarjuna University is offering educational opportunities at the UG, PG levels apart from research degrees to students from over 221 affiliated colleges spread over the two districts of Guntur and Prakasam.

The University has also started the Centre for Distance Education in 2003-04 with the aim of taking higher education to the door step of all the sectors of the society. The centre will be a great help to those who cannot join in colleges, those who cannot afford the exorbitant fees as regular students, and even to housewives desirous of pursuing higher studies. Acharya Nagarjuna University has started offering B.Sc., B.A., B.B.A., and B.Com courses at the Degree level and M.A., M.Com., M.Sc., M.B.A., and L.L.M., courses at the PG level from the academic year 2003-2004 onwards.

To facilitate easier understanding by students studying through the distance mode, these self-instruction materials have been prepared by eminent and experienced teachers. The lessons have been drafted with great care and expertise in the stipulated time by these teachers. Constructive ideas and scholarly suggestions are welcome from students and teachers involved respectively. Such ideas will be incorporated for the greater efficacy of this distance mode of education. For clarification of doubts and feedback, weekly classes and contact classes will be arranged at the UG and PG levels respectively.

It is my aim that students getting higher education through the Centre for Distance Education should improve their qualification, have better employment opportunities and in turn be part of country's progress. It is my fond desire that in the years to come, the Centre for Distance Education will go from strength to strength in the form of new courses and by catering to larger number of people. My congratulations to all the Directors, Academic Coordinators, Editors and Lesson-writers of the Centre who have helped in these endeavors.

Prof. K. Gangadhara Rao
M.Tech., Ph.D.,
Vice-Chancellor I/c
Acharya Nagarjuna University.

MASTER OF COMPUTER APPLICATIONS (MCA)

FIRST YEAR, Semester-II, Paper-VI

206MC24: PYTHON PROGRAMMING LAB

SYLLABUS

Simple Programs

1. Write a program using print Pascal triangle.
2. write a program to find out the roots of the quadratic equations.
3. write a program to display the Fibonacci series using generators.
4. write a program to check the given number is palindrome or not.
5. Write a program to find the sum of digits of a given number
6. write a Python program to calculate $X = \frac{1}{2!} = \frac{1}{2!} + \frac{2}{4!} + \frac{4}{8!} + \frac{8}{16!} \frac{2}{4!} + \frac{4}{8!} + \frac{8}{16!}$
7. write a Python program to remove the punctuations from a string.
8. Write a Python program to implement the simple calculator.
9. write a Python program to print the lower and upper triangles of a matrix.
10. Write a Python program to merge two mails.

Functions

1. Write a recursive Python function that has a parameter representing a list of integers and returns the maximum stored in the list.
2. Write a recursive Python function to that generates the top n prime numbers in the range 1 to 1000.
3. write a python function to calculate the multiplication of two matrices.
4. Write a Python function to reverse the given string.
5. Write a Python function that takes an integer n and a character c, returns a string and displays as "xxxxx" (Ex: the length of the retuned string is 5, then the output as XXXXX)
6. Write Python function that the search the given number in the list of numbers by using binary search.
7. Write a Python function to convert the given decimal number into binary number by using recursion.
8. Write a Python function to sort the list of records in a file.

GUI Programs

1. Construct a GUI application to generate the employee pay slip
2. construct a GUI application to generate a Bar Graph for a excel data
3. construct a GUI application to perform the Arithmetic operations
4. Read Input Values through input window
5. Choose choice and Operation through following windows
6. Display the result in Message Box.

206MC24-PYTHON PROGRAMMING LAB

AIM AND OBJECTIVES

To develop students' problem-solving ability through the design and implementation of efficient programs using Python programming language, applying structured, object-oriented, and GUI-based approaches.

After successful completion of this lab course, students will be able to:

1. Understand and apply the fundamentals of Python syntax, semantics, and data structures.
2. Develop problem-solving skills through the implementation of algorithms using Python.
3. Use functions, modules, and recursion to design modular and reusable code.
4. Apply object-oriented principles such as classes, objects, inheritance, and polymorphism.
5. Handle files, exceptions, and data structures effectively in Python applications.
6. Design GUI applications using tkinter for interactive software development.
7. Integrate Python with real-world applications such as databases, graphical interfaces, and web interactions.

STRUCTURE

A. Simple Python Programs

1. Program to print Pascal's triangle.
2. Program to find roots of a quadratic equation.
3. Program to display Fibonacci series using generators.
4. Program to check if a number is palindrome or not.
5. Program to find the sum of digits of a number.
6. Program to calculate X^n.
7. Program to remove punctuations from a string.
8. Program to implement a simple calculator.
9. Program to print lower and upper triangles of a matrix.
10. Program to merge two mail strings.

B. Function-Based Programs

1. Recursive function to find the maximum in a list.
2. Recursive function to generate top *n* prime numbers between 1–1000.
3. Function to multiply two matrices.
4. Function to reverse a string.
5. Function that returns a string of repeated characters.
6. Function to search a number in a list using binary search.
7. Recursive function to convert decimal to binary.
8. Function to sort list of records in a file.

C. GUI Programs (tkinter Applications)

1. GUI application to generate an employee pay slip.
2. GUI application to display a bar graph for Excel data.
3. GUI application to perform arithmetic operations with input and message boxes.
4. Input and operation selection through tkinter windows.
5. Display result using message box or output label.

SIMPLE PROGRAMS

Programs -1

Program to print Pascal's Triangle

```
from math import factorial
n = int(input("Enter number of rows: "))
for i in range(n):
    for j in range(n - i + 1):
        print(" ", end="")
    for j in range(i + 1):
        print(factorial(i)//(factorial(j)*factorial(i-j)), end=" ")
    print()
```

Test Cases

Test Case	Input (n)	Expected Output
1	1	1
2	3	1 1 1 1 2 1
3	5	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1
4	0	No output
5	7	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1

Output (Sample)
Input:
Enter number of rows: 5
Output:

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

Output (For n = 7)

Input:
Enter number of rows: 7
Output:
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

Viva Questions

Question	**Answer**
1. What is Pascal's Triangle?	A triangular arrangement of binomial coefficients where each number is the sum of the two above it.
2. What formula is used in the program?	$C(i, j) = i! / (j! \times (i-j)!)$.
3. What is the purpose of the factorial function?	It calculates the product of all positive integers up to a number.
4. Why is Pascal's Triangle symmetric?	Because $C(i, j) = C(i, i-j)$.
5. What is the time complexity of the program?	$O(n^2)$ due to nested loops for rows and elements.

SIMPLE PROGRAMS

Programs -2

Program to find roots of a Quadratic Equation

```
import cmath
a = float(input("Enter a: "))
b = float(input("Enter b: "))
c = float(input("Enter c: "))
d = (b**2) - (4*a*c)
root1 = (-b + cmath.sqrt(d)) / (2*a)
root2 = (-b - cmath.sqrt(d)) / (2*a)
print("The roots are:", root1, "and", root2)
```

Test Cases

Test Case	Input (a, b, c)	Expected Output (Roots)
1	1, 5, 6	(-2+0j) and (-3+0j)
2	1, 2, 1	(-1+0j) and (-1+0j)
3	1, 4, 5	(-2+1j) and (-2-1j)
4	2, 3, 1	(-0.5+0j) and (-1+0j)
5	1, -7, 10	(5+0j) and (2+0j)

Output (Sample)

Input:
Enter a: 1
Enter b: 5
Enter c: 6
Output:
The roots are: (-2+0j) and (-3+0j)

Additional Outputs

Input:
Enter a: 1
Enter b: 2
Enter c: 1
Output:
The roots are: (-1+0j) and (-1+0j)

Input:
Enter a: 1
Enter b: 4
Enter c: 5
Output:
The roots are: (-2+1j) and (-2-1j)

Input:
Enter a: 2
Enter b: 3
Enter c: 1
Output:
The roots are: (-0.5+0j) and (-1+0j)

Input:
Enter a: 1
Enter b: -7
Enter c: 10
Output:
The roots are: (5+0j) and (2+0j)

Viva Questions

Question	Answer
1. What is the quadratic formula?	$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
2. What is the discriminant?	$D = b^2 - 4ac$
3. What does the cmath module do?	Handles complex numbers.
4. What are the possible types of roots?	Real & distinct, real & equal, or complex.
5. Why divide by (2*a)?	It comes from solving the quadratic equation $ax^2 + bx + c = 0$.

SIMPLE PROGRAMS

Programs -3

Program to display Fibonacci Series using Generators

```
def fibonacci():
    a, b = 0, 1
    while True:
        yield a
        a, b = b, a + b
n = int(input("Enter number of terms: "))
f = fibonacci()
for i in range(n):
    print(next(f), end=" ")
```

Test Cases

Test Case	Input (n)	Expected Output
1	1	0
2	5	0 1 1 2 3
3	7	0 1 1 2 3 5 8
4	10	0 1 1 2 3 5 8 13 21 34
5	0	*(No output)*

Output (Sample)
Input:
Enter number of terms: 5
Output:
0 1 1 2 3

Additional Outputs
Input:
Enter number of terms: 7
Output:
0 1 1 2 3 5 8

Input:
Enter number of terms: 10

Output:
0 1 1 2 3 5 8 13 21 34

Input:
Enter number of terms: 1
Output:
0

Input:
Enter number of terms: 0
Output:
(No output)

Viva Questions

Question	**Answer**
1. What is a Fibonacci sequence?	A series where each number is the sum of the previous two, starting from 0 and 1.
2. What does yield do in Python?	It turns a function into a generator that produces values one at a time.
3. What is the difference between return and yield?	return ends the function, yield pauses and resumes it to produce a sequence of values.
4. What is the time complexity of generating Fibonacci numbers?	O(n) for n terms.
5. Why use a generator for Fibonacci series?	It saves memory by generating numbers one by one instead of storing the entire sequence.

SIMPLE PROGRAMS

Programs -4

Program to check Palindrome Number

```
num = input("Enter a number: ")
if num == num[::-1]:
    print("Palindrome Number")
else:
    print("Not a Palindrome Number")
```

Test Cases

Test Case	Input (num)	Expected Output
1	121	Palindrome Number
2	123	Not a Palindrome Number
3	4554	Palindrome Number
4	98789	Palindrome Number
5	120	Not a Palindrome Number

Output (Sample)
Input:
Enter a number: 121
Output:
Palindrome Number

Additional Outputs
Input:
Enter a number: 123
Output:
Not a Palindrome Number

Input:
Enter a number: 4554
Output:
Palindrome Number

Input:
Enter a number: 98789
Output:
Palindrome Number

Input:
Enter a number: 120
Output:
Not a Palindrome Number

Viva Questions

Question	**Answer**
1. What is a palindrome number?	A number that reads the same forward and backward.
2. What does num[::-1] do?	It reverses the string num using slicing.
3. Why is input taken as a string here?	To easily reverse and compare digits.
4. Can this logic be used for words too?	Yes, the same code works for palindrome strings.
5. What is the time complexity of this program?	O(n), where n is the number of digits.

SIMPLE PROGRAMS

Programs -5

Program to find Sum of Digits

```
n = int(input("Enter a number: "))
sum = 0
while n > 0:
    sum += n % 10
    n //= 10
print("Sum of digits =", sum)
```

Test Cases

Test Case	Input (n)	Expected Output
1	123	Sum of digits = 6
2	987	Sum of digits = 24
3	1005	Sum of digits = 6
4	9	Sum of digits = 9
5	4567	Sum of digits = 22

Output (Sample)
Input:
Enter a number: 123
Output:
Sum of digits = 6

Additional Outputs
Input:
Enter a number: 987
Output:
Sum of digits = 24

Input:
Enter a number: 1005
Output:
Sum of digits = 6

Input:
Enter a number: 9
Output:
Sum of digits = 9

Input:
Enter a number: 4567
Output:
Sum of digits = 22

Viva Questions

Question	**Answer**
1. What is the purpose of n % 10?	It extracts the last digit of the number.
2. What does n //= 10 do?	It removes the last digit from the number.
3. What is the initial value of sum and why?	0, to start adding digits from zero.
4. What is the time complexity of this program?	O(d), where d is the number of digits in the number.
5. Can this code handle negative numbers?	Not correctly; it needs modification to handle negatives.

SIMPLE PROGRAMS

Programs -6

Program to calculate X^n

```
x = int(input("Enter base: "))
n = int(input("Enter power: "))
print("Result =", x ** n)
```

Test Cases

Test Case	Input (x, n)	Expected Output
1	2, 3	Result = 8
2	5, 0	Result = 1
3	7, 2	Result = 49
4	10, 5	Result = 100000
5	3, 4	Result = 81

Output (Sample)
Input:
Enter base: 2
Enter power: 3
Output:
Result = 8

Additional Outputs
Input:
Enter base: 5
Enter power: 0
Output:
Result = 1

Input:
Enter base: 7
Enter power: 2
Output:
Result = 49

Input:
Enter base: 10
Enter power: 5

Output:
Result = 100000

Input:
Enter base: 3
Enter power: 4
Output:
Result = 81

Viva Questions

Question	**Answer**
1. What does the operator ** do in Python?	It performs exponentiation (raises a number to a power).
2. What is x and n in this program?	x is the base and n is the power (exponent).
3. What is the result when any number is raised to power 0?	The result is always 1 (except 0^0 which is undefined).
4. How can you calculate power without using **?	By using a loop or the built-in pow(x, n) function.
5. What is the time complexity of x ** n?	O(log n) internally, since Python uses fast exponentiation.

SIMPLE PROGRAMS

Programs -7

Program to remove punctuations from a string

```
punc = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''
s = input("Enter string: ")
no_punc = ""
for ch in s:
    if ch not in punc:
        no_punc += ch
print(no_punc)
```

Test Cases

Test Case	Input (s)	Expected Output
1	Hello, World!	Hello World
2	Good-morning!	Goodmorning
3	Python@3.11	Python311
4	Welcome!!!	Welcome
5	Wow! Such *clean* code.	Wow Such clean code

Output (Sample)

Input:

Enter string: Hello, World!

Output:

Hello World

Additional Outputs

Input:

Enter string: Good-morning!

Output:

Goodmorning

Input:

Enter string: Python@3.11

Output:

Python311

Input:

Enter string: Welcome!!!

Output:

Welcome

Input:

Enter string: Wow! Such clean code.

Output:

Wow Such clean code

Viva Questions

Question	**Answer**
1. What is the purpose of this program?	To remove punctuation marks from a given string.
2. What does the variable punc contain?	A list of punctuation characters to be removed.
3. What is the role of the condition if ch not in punc:?	It checks if the character is not punctuation before adding it to the result string.
4. What data type is used to store the cleaned string?	A regular Python string (str).
5. Can this program handle spaces and numbers?	Yes, they are kept unchanged as they are not in the punctuation list.

SIMPLE PROGRAMS

Programs -8

Simple Calculator

```
def add(x,y): return x+y
def sub(x,y): return x-y
def mul(x,y): return x*y
def div(x,y): return x/y
print("Select Operation: 1.Add 2.Sub 3.Mul 4.Div")
ch = int(input("Enter choice: "))
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))

if ch==1: print("Result =", add(a,b))
elif ch==2: print("Result =", sub(a,b))
elif ch==3: print("Result =", mul(a,b))
elif ch==4: print("Result =", div(a,b))
else: print("Invalid Choice")
```

Test Cases

Test Case	Input (ch, a, b)	Expected Output
1	1, 10, 5	Result = 15.0
2	2, 10, 5	Result = 5.0
3	3, 4, 6	Result = 24.0
4	4, 20, 5	Result = 4.0
5	7, 8, 2	Invalid Choice

Output (Sample)

Input:

Select Operation: 1.Add 2.Sub 3.Mul 4.Div

Enter choice: 1

Enter first number: 10

Enter second number: 5

Output:

Result = 15.0

Additional Outputs

Input:

Enter choice: 2

Enter first number: 10

Enter second number: 5

Output:

Result = 5.0

Input:

Enter choice: 3

Enter first number: 4

Enter second number: 6

Output:

Result = 24.0

Input:

Enter choice: 4

Enter first number: 20

Enter second number: 5

Output:

Result = 4.0

Input:

Enter choice: 7

Enter first number: 8

Enter second number: 2

Output:

Invalid Choice

Viva Questions

Question	**Answer**
1. What is the purpose of defining functions here?	To perform arithmetic operations modularly (add, sub, mul, div).
2. Why are float values used for inputs?	To handle both integers and decimal numbers.
3. What does if-elif-else do in this program?	It checks which operation the user selected and executes the corresponding function.
4. What happens if the user enters division by zero?	A runtime error (ZeroDivisionError) occurs.
5. How can the program be improved?	By adding input validation and error handling for invalid or zero division cases.

SIMPLE PROGRAMS

Programs -9

Print Lower and Upper Triangles of a Matrix

```
import numpy as np
n = int(input("Enter matrix size: "))
mat = np.array([list(map(int, input().split())) for _ in range(n)])
print("Lower Triangle:")
for i in range(n):
    for j in range(n):
        if j <= i: print(mat[i][j], end=" ")
        else: print("0", end=" ")
    print()
print("Upper Triangle:")
for i in range(n):
    for j in range(n):
        if j >= i: print(mat[i][j], end=" ")
        else: print("0", end=" ")
    print()
```

Test Cases

Test Case	Input	Expected Output
1	n = 3 1 2 3 4 5 6 7 8 9	Lower Triangle: 1 0 0 4 5 0 7 8 9 Upper Triangle: 1 2 3 0 5 6 0 0 9
2	n = 2 5 6 7 8	Lower Triangle: 5 0 7 8 Upper Triangle: 5 6 0 8
3	n = 3 2 0 1 3 5 4 6 7 8	Lower Triangle: 2 0 0 3 5 0 6 7 8 Upper Triangle: 2 0 1 0 5 4 0 0 8

4	n = 1 9	**Lower Triangle:** 9 **Upper Triangle:** 9
5	n = 4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	**Lower Triangle:** 1 0 0 0 5 6 0 0 9 10 11 0 13 14 15 16 **Upper Triangle:** 1 2 3 4 0 6 7 8 0 0 11 12 0 0 0 16

Output (Sample)

Input:
Enter matrix size: 3
1 2 3
4 5 6
7 8 9
Output:
Lower Triangle:
1 0 0
4 5 0
7 8 9

Upper Triangle:
1 2 3
0 5 6
0 0 9

Additional Outputs

Input:
Enter matrix size: 2
5 6
7 8
Output:
Lower Triangle:
5 0
7 8

Upper Triangle:
5 6
0 8

Input:
Enter matrix size: 3
2 0 1
3 5 4
6 7 8
Output:
Lower Triangle:
2 0 0
3 5 0
6 7 8

Upper Triangle:
2 0 1
0 5 4
0 0 8

Viva Questions

Question	**Answer**
1. What is the difference between upper and lower triangular matrices?	In a lower triangular matrix, elements above the main diagonal are zero; in an upper triangular matrix, elements below the diagonal are zero.
2. What does np.array() do?	It creates a NumPy array from a list or sequence.
3. How is input taken for the matrix?	Using list comprehension and map(int, input().split()) for each row.
4. What is the purpose of the condition j <= i and j >= i?	j <= i selects lower triangle elements, j >= i selects upper triangle elements.
5. Can this code work for non-square matrices?	No, it's designed only for square (n×n) matrices.

SIMPLE PROGRAMS

Programs -10

Merge Two Mails

```
template = """Dear {name},

We are pleased to inform you that your application for {position}
has been successfully received. Our HR team will contact you soon.

Thank you,
Best Regards,
{sender}
"""
n = int(input("Enter number of recipients: "))
for i in range(n):
    name = input("Enter recipient name: ")
    position = input("Enter position applied for: ")
    sender = input("Enter sender name: ")

    message = template.format(name=name, position=position, sender=sender)
    print("\n--- Merged Mail ---")
    print(message)
    print("--------------------\n")
```

Test Cases

Test Case	Input	Expected Output (Merged Mail)
1	n=1 name=John position=Software Engineer sender=HR Manager	Dear John, We are pleased to inform you that your application for Software Engineer has been successfully received. Our HR team will contact you soon. Thank you, Best Regards, HR Manager
2	n=2 name=Alice, Bob position=Data Analyst, Developer sender=HR Team	Two personalized letters generated — one for Alice and one for Bob.
3	n=1 name=Ravi position=Intern sender=Recruitment Team	Personalized message for Ravi.
4	n=0	No output (no recipients).
5	n=1 name=Priya position=Research Assistant sender=Dean	Personalized letter for Priya.

Output (Sample)
Input:
Enter number of recipients: 1
Enter recipient name: John
Enter position applied for: Software Engineer
Enter sender name: HR Manager
Output:
Dear John,

We are pleased to inform you that your application for Software Engineer
has been successfully received. Our HR team will contact you soon.

Thank you,
Best Regards,
HR Manager

Additional Outputs
Input:
Enter number of recipients: 2
Enter recipient name: Alice
Enter position applied for: Data Analyst
Enter sender name: HR Team

Enter recipient name: Bob
Enter position applied for: Developer
Enter sender name: HR Team
Output:
Two personalized letters generated — one for Alice and one for Bob.

Input:
Enter number of recipients: 1
Enter recipient name: Ravi
Enter position applied for: Intern
Enter sender name: Recruitment Team
Output:
Personalized message for Ravi.

Input:
Enter number of recipients: 0
Output:
No output (no recipients).

Input:

Enter number of recipients: 1
Enter recipient name: Priya
Enter position applied for: Research Assistant
Enter sender name: Dean
Output:
Personalized letter for Priya.

Viva Questions

Question	**Answer**
1. What is Mail Merge?	A process of automatically generating personalized letters or emails using a template and a data source.
2. What does .format() do in Python?	It replaces placeholders (like {name}) in a string with actual values.
3. How is this program useful?	It saves time when sending customized messages to multiple recipients.
4. What data structure is used to store the template?	A multi-line string using triple quotes (""" """).
5. How can this program be extended?	By reading recipient details from a CSV or Excel file using the csv or pandas module.

FUNCTION-BASED PROGRAMS

Programs -1

Recursive Function – Maximum in a List

```
def find_max(lst):
    if len(lst) == 1:
        return lst[0]
    else:
        m = find_max(lst[1:])
        return m if m > lst[0] else lst[0]

print(find_max([3, 9, 1, 5, 7]))
```

Test Cases

Test Case	Input (List)	Expected Output
1	[3, 9, 1, 5, 7]	9
2	[10, 2, 30, 4]	30
3	[5]	5
4	[8, 6, 9, 9, 7]	9
5	[-5, -2, -9, -1]	-1

Output (Sample)
Input:
[3, 9, 1, 5, 7]
Output:
9

Additional Outputs
Input:
[10, 2, 30, 4]
Output:
30

Input:
[5]
Output:
5

Input:
[8, 6, 9, 9, 7]
Output:
9

Input:
[-5, -2, -9, -1]
Output:
-1

Viva Questions

Question	**Answer**
1. What concept is used in this program?	**Recursion** — the function calls itself to find the maximum.
2. What is the base case in this program?	When the list length is 1, it returns that single element.
3. What does lst[1:] mean?	It creates a sublist excluding the first element.
4. What happens if the list is empty?	It will cause an error (IndexError); needs a condition to handle empty lists.
5. What is the time complexity of this program?	O(n), as it compares each element once.

FUNCTION-BASED PROGRAMS

Programs -2

Recursive Function – First n Prime Numbers

```
def is_prime(n):
    if n < 2: return False
    for i in range(2, int(n**0.5)+1):
        if n % i == 0: return False
    return True

def print_primes(n):
    primes = [x for x in range(1, 1000) if is_prime(x)]
    print(primes[:n])

print_primes(10)
```

Test Cases

Test Case	Input (n)	Expected Output
1	5	[2, 3, 5, 7, 11]
2	10	[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
3	1	[2]
4	15	[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
5	0	[]

Output (Sample)
Input:
Enter n: 5
Output:
[2, 3, 5, 7, 11]

Additional Outputs
Input:
Enter n: 10

Output:
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

Input:
Enter n: 1
Output:
[2]

Input:
Enter n: 15
Output:
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

Input:
Enter n: 0
Output:
[]

Viva Questions

Question	**Answer**
1. What is a prime number?	A number greater than 1 that has no divisors other than 1 and itself.
2. Why check divisibility only up to √n?	Because if n has a factor greater than √n, the corresponding smaller factor has already been checked.
3. What does the list comprehension [x for x in range(1, 1000) if is_prime(x)] do?	It creates a list of all prime numbers below 1000.
4. What is the time complexity of the is_prime function?	O(√n) for each number checked.
5. How can this program be improved for larger ranges?	By using the Sieve of Eratosthenes algorithm to generate primes efficiently.

FUNCTION-BASED PROGRAMS

Programs -3

Matrix Multiplication Function

```
def matmul(A, B):
    result = [[sum(a*b for a,b in zip(A_row,B_col)) for B_col in zip(*B)] for A_row in A]
    return result


A = [[1,2,3],[4,5,6]]
B = [[1,2],[3,4],[5,6]]

for r in matmul(A,B):
    print(r)
```

Test Cases

Test Case	Input Matrices (A × B)	Expected Output (Resultant Matrix)
1	A = [[1,2,3],[4,5,6]] B = [[1,2],[3,4],[5,6]]	[22, 28] [49, 64]
2	A = [[2,4],[3,5]] B = [[1,0],[0,1]]	[2, 4] [3, 5]
3	A = [[1,0,0],[0,1,0],[0,0,1]] B = [[2,3,4],[5,6,7],[8,9,10]]	[2,3,4] [5,6,7] [8,9,10]
4	A = [[1,2],[3,4],[5,6]] B = [[7,8,9],[10,11,12]]	[27,30,33] [61,68,75] [95,106,117]
5	A = [[1,2,3]] B = [[4],[5],[6]]	[32]

Output (Sample)

Input:

A = [[1, 2, 3], [4, 5, 6]]

B = [[1, 2], [3, 4], [5, 6]]

Output:

[22, 28]

[49, 64]

Additional Outputs

Input:

A = [[2, 4], [3, 5]]

B = [[1, 0], [0, 1]]

Output:

[2, 4]

[3, 5]

Input:

A = [[1, 0, 0], [0, 1, 0], [0, 0, 1]]

B = [[2, 3, 4], [5, 6, 7], [8, 9, 10]]

Output:

[2, 3, 4]

[5, 6, 7]

[8, 9, 10]

Input:

A = [[1, 2], [3, 4], [5, 6]]

B = [[7, 8, 9], [10, 11, 12]]

Output:

[27, 30, 33]

[61, 68, 75]

[95, 106, 117]

Input:

A = [[1, 2, 3]]

B = [[4], [5], [6]]

Output:

[32]

Viva Questions

Question	**Answer**
1. What operation does this program perform?	It performs **matrix multiplication**.
2. What is the condition for matrix multiplication to be valid?	The number of **columns in A** must equal the number of **rows in B**.
3. What does zip(*B) do?	It transposes matrix B (converts rows into columns).
4. What is the time complexity of matrix multiplication?	O(m × n × p), where A is m×n and B is n×p.
5. What does sum(a*b for a,b in zip(A_row,B_col)) compute?	The **dot product** of a row from A and a column from B.

FUNCTION-BASED PROGRAMS

Programs -4

Reverse String Function

```
def reverse(s):
    return s[::-1]

print(reverse("Python"))
```

Test Cases

Test Case	Input (s)	Expected Output
1	"Python"	nohtyP
2	"Hello"	olleH
3	"12345"	54321
4	"madam"	madam
5	"AI"	IA

Output (Sample)

Input:

Python

Output:

nohtyP

Additional Outputs

Input:

Hello

Output:

olleH

Input:

12345

Output:

54321

Input:

madam

Output:

madam

Input:

AI

Output:

IA

Viva Questions

Question	**Answer**
1. What does s[::-1] do?	It reverses the string using slicing.
2. What is string slicing in Python?	Extracting a substring or reversing a string using index positions.
3. Is the original string modified by this operation?	No, strings in Python are **immutable**.
4. What is the time complexity of reversing a string?	O(n), where n is the length of the string.
5. Can this method be used for lists too?	Yes, lists can also be reversed using slicing list[::-1].

FUNCTION-BASED PROGRAMS

Programs -5

Function to Return a String of n Repeated Characters

```
def make_str(n, c):
    return c * n

print(make_str(5, 'X'))
```

Test Cases

Test Case	Input (n, c)	Expected Output
1	5, 'X'	XXXXX
2	3, '*'	***
3	7, '#'	#######
4	0, 'A'	*(Empty string)*
5	4, 'Hi'	HiHiHiHi

Output (Sample)

Input:

Enter n: 5

Enter c: X

Output:

XXXXX

Additional Outputs

Input:

Enter n: 3

Enter c: *

Output:

***

Input:

Enter n: 7

Enter c: #

Output:

#######

Input:

Enter n: 0

Enter c: A

Output:

(Empty string)

Input:

Enter n: 4

Enter c: Hi

Output:

HiHiHiHi

Viva Questions

Question	**Answer**
1. What does c * n do in Python?	It repeats the string c exactly n times.
2. What data type is returned by this function?	A **string**.
3. Can we use this method for other data types like integers?	No, it only works for strings and sequences.
4. What is the time complexity of this operation?	O(n × len(c)), proportional to the number of characters generated.
5. What happens if n is 0?	It returns an empty string ('').

FUNCTION-BASED PROGRAMS

Programs -6

Binary Search

```
def binary_search(lst, x):
    low, high = 0, len(lst)-1
    while low <= high:
        mid = (low + high)//2
        if lst[mid] == x: return mid
        elif lst[mid] < x: low = mid + 1
        else: high = mid - 1
    return -1

print(binary_search([1,3,5,7,9,11], 7))
```

Test Cases

Test Case	Input (List, x)	Expected Output
1	[1, 3, 5, 7, 9, 11], 7	3
2	[2, 4, 6, 8, 10], 8	3
3	[5, 10, 15, 20, 25], 5	0
4	[5, 10, 15, 20, 25], 25	4
5	[1, 2, 3, 4, 5], 6	-1

Output (Sample)

Input:

[1, 3, 5, 7, 9, 11], 7

Output:

3

Additional Outputs

Input:

[2, 4, 6, 8, 10], 8

Output:

3

Input:

[5, 10, 15, 20, 25], 5

Output:

0

Input:

[5, 10, 15, 20, 25], 25

Output:

4

Input:

[1, 2, 3, 4, 5], 6

Output:

-1

Viva Questions

Question	**Answer**
1. What is binary search?	An efficient algorithm to find an element in a **sorted list** by repeatedly dividing the search interval in half.
2. What is the prerequisite for binary search?	The list must be **sorted**.
3. What is the time complexity of binary search?	O(log n).
4. What happens if the element is not found?	The function returns **-1**.
5. What does (low + high) // 2 represent?	The **middle index** of the current search range.

FUNCTION-BASED PROGRAMS

Programs -7

Decimal to Binary using Recursion

```
def dec_to_bin(n):
    if n == 0:
        return ""
    else:
        return dec_to_bin(n//2) + str(n%2)

print(dec_to_bin(10))
```

Test Cases

Test Case	Input (n)	Expected Output
1	10	1010
2	5	101
3	8	1000
4	15	1111
5	1	1

Output (Sample)

Input:

Enter a number: 10

Output:

1010

Additional Outputs

Input:

Enter a number: 5

Output:

101

Input:

Enter a number: 8

Output:

1000

Input:

Enter a number: 15

Output:

1111

Input:

Enter a number: 1

Output:

1

Viva Questions

Question	**Answer**
1. What concept is used in this program?	**Recursion** — the function calls itself to convert decimal to binary.
2. What does n // 2 do?	It performs integer division, reducing n for the next recursive step.
3. What does n % 2 represent?	The remainder when dividing by 2 — the current binary digit (bit).
4. What is the base case in this recursion?	When n == 0, the recursion stops.
5. What is the time complexity of this program?	O(log n), since the number is divided by 2 at each recursive step.

FUNCTION-BASED PROGRAMS

Programs -8

Sort List of Records in a File

```
with open("records.txt", "w") as f:
    f.write("Ram,85\nShyam,90\nAmit,75\n")
lines = open("records.txt").readlines()
lines.sort()
open("sorted_records.txt", "w").writelines(lines)
print("Records sorted and written to file.")
```

Test Cases

Test Case	Content of `records.txt` (Before Sorting)	Expected Content of `sorted_records.txt` (After Sorting)
1	Ram,85 Shyam,90 Amit,75	Amit,75 Ram,85 Shyam,90
2	Zoya,95 Arun,70 Meena,80	Arun,70 Meena,80 Zoya,95
3	Chetan,65 Bala,75 Anita,85	Anita,85 Bala,75 Chetan,65
4	Rohit,88 Karan,92	Karan,92 Rohit,88
5	Ravi,90	Ravi,90 *(only one record — remains same)*

Output (Sample)
Before Sorting (records.txt):
Ram,85
Shyam,90
Amit,75
After Sorting (sorted_records.txt):
Amit,75
Ram,85
Shyam,90
Additional Outputs
Before Sorting (records.txt):
Zoya,95

Arun,70
Meena,80
After Sorting (sorted_records.txt):
Arun,70
Meena,80
Zoya,95

Before Sorting (records.txt):
Chetan,65
Bala,75
Anita,85
After Sorting (sorted_records.txt):
Anita,85
Bala,75
Chetan,65

Before Sorting (records.txt):
Rohit,88
Karan,92
After Sorting (sorted_records.txt):
Karan,92
Rohit,88

Before Sorting (records.txt):
Ravi,90
After Sorting (sorted_records.txt):
Ravi,90

Viva Questions

Question	**Answer**
1. What is the purpose of this program?	To write student records to a file, then sort them alphabetically and save them to another file.
2. What does "w" mode mean in open()?	It opens the file for **writing**, overwriting existing content.
3. What does readlines() return?	A **list of strings**, each representing one line from the file.
4. What is the effect of lines.sort()?	It sorts the lines alphabetically (lexicographically).
5. What happens to the file when "w" mode is used on an existing file?	The existing content is **erased** before writing new data.

GUI Programs (tkinter Applications)

Programs -1

GUI Application – Employee Pay Slip

```
import tkinter as tk
from tkinter import messagebox

def generate_payslip():
    name = e1.get()
    salary = float(e2.get())
    hra = salary * 0.2
    da = salary * 0.1
    gross = salary + hra + da
    messagebox.showinfo("Pay Slip", f"Name: {name}\nGross Salary: {gross}")

root = tk.Tk()
root.title("Employee Pay Slip")

tk.Label(root, text="Employee Name").grid(row=0, column=0)
tk.Label(root, text="Basic Salary").grid(row=1, column=0)

e1 = tk.Entry(root)
e2 = tk.Entry(root)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)

tk.Button(root, text="Generate", command=generate_payslip).grid(row=2, column=1)
root.mainloop()
```

Test Cases

Test Case	Input (Name, Basic Salary)	Expected Output (Shown in Message Box)
1	Ram, 10000	Name: Ram Gross Salary: 13000.0
2	Sita, 15000	Name: Sita Gross Salary: 19500.0
3	Ravi, 20000	Name: Ravi Gross Salary: 26000.0
4	Meena, 25000	Name: Meena Gross Salary: 32500.0
5	Arjun, 0	Name: Arjun Gross Salary: 0.0

Output (Sample)
Input:
Employee Name: Ram
Basic Salary: 10000
Output (in Message Box):
Name: Ram
Gross Salary: 13000.0

Additional Outputs
Input:
Employee Name: Sita
Basic Salary: 15000
Output (in Message Box):
Name: Sita
Gross Salary: 19500.0

Input:
Employee Name: Ravi
Basic Salary: 20000
Output (in Message Box):
Name: Ravi
Gross Salary: 26000.0

Input:
Employee Name: Meena
Basic Salary: 25000
Output (in Message Box):
Name: Meena

Gross Salary: 32500.0

Input:
Employee Name: Arjun
Basic Salary: 0
Output (in Message Box):
Name: Arjun
Gross Salary: 0.0

Viva Questions

Question	**Answer**
1. What is Tkinter?	Tkinter is Python's standard GUI (Graphical User Interface) library used to create desktop applications.
2. What is the purpose of messagebox.showinfo()?	It displays a pop-up information dialog box with a message.
3. What do Entry widgets do in Tkinter?	They are input fields that allow the user to type text.
4. What does grid(row, column) do?	It arranges widgets in a tabular (row-column) layout on the window.
5. What are HRA and DA in this program?	HRA (House Rent Allowance) = 20% of salary, DA (Dearness Allowance) = 10% of salary, both added to get the gross salary.

GUI Programs (tkinter Applications)

Programs -2

GUI – Bar Graph from Excel Data
(requires matplotlib & pandas)

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_excel("data.xlsx")
plt.bar(data['Name'], data['Marks'])
plt.title("Student Marks Bar Graph")
plt.xlabel("Name")
plt.ylabel("Marks")
plt.show()
```

Test Cases

Test Case	Content of `data.xlsx`	Expected Output (Bar Graph)
1	Name Marks Ram	
2	Name Marks Asha	
3	Name Marks John	
4	Name Marks Raj	
5	Name Marks Sam	

Output (Sample)

Input File (data.xlsx):

Name	Marks
Ram	85

Name	Marks
Sita	90
Ravi	75
Meena	80
Arjun	70

Output (Bar Graph):

A bar chart displaying:

- **X-axis:** Names of students (Ram, Sita, Ravi, Meena, Arjun)
- **Y-axis:** Marks (85, 90, 75, 80, 70)
- **Title:** "Student Marks Bar Graph"

Additional Test Cases

Test Case	Content of data.xlsx	Expected Output (Bar Graph)
1	Name: Ram	Marks: 85
2	Name: Asha	Marks: 92
3	Name: John	Marks: 67
4	Name: Raj	Marks: 78
5	Name: Sam	Marks: 55

Viva Questions

Question	Answer
1. What is the purpose of the pandas library?	To read, manipulate, and analyze data easily (supports formats like Excel, CSV, etc.).
2. What function reads the Excel file?	pd.read_excel("filename.xlsx")
3. What does plt.bar() do?	It creates a vertical bar chart using the given x and y values.
4. What are the functions of xlabel() and ylabel()?	They label the X-axis and Y-axis of the graph, respectively.
5. What does plt.show() do?	It displays the plotted graph window.

GUI Programs (tkinter Applications)

Programs -3

GUI – Arithmetic Operations

```
import tkinter as tk
from tkinter import messagebox


def calc(op):
    a = float(e1.get())
    b = float(e2.get())
    if op == '+': res = a + b
    elif op == '-': res = a - b
    elif op == '*': res = a * b
    elif op == '/': res = a / b
    messagebox.showinfo("Result", f"Result: {res}")

root = tk.Tk()
root.title("Simple Calculator")

tk.Label(root, text="Enter A").grid(row=0, column=0)
tk.Label(root, text="Enter B").grid(row=1, column=0)
e1 = tk.Entry(root); e2 = tk.Entry(root)
e1.grid(row=0, column=1); e2.grid(row=1, column=1)

for i, op in enumerate(['+', '-', '*', '/']):
    tk.Button(root, text=op, command=lambda o=op: calc(o)).grid(row=2, column=i)

root.mainloop()
```

Test Cases

Test Case	Input (A, B, Operation)	Expected Output (Message Box)
1	10, 5, `+`	Result: 15.0
2	8, 3, `-`	Result: 5.0
3	6, 7, `*`	Result: 42.0
4	20, 4, `/`	Result: 5.0
5	5, 0, `/`	Error (ZeroDivisionError – should be handled ideally)

Output (Sample)
Input:
Enter A: 10
Enter B: 5
Operation: +
Output (Message Box):
Result: 15.0

Additional Outputs
Input:
Enter A: 8
Enter B: 3
Operation: -
Output (Message Box):
Result: 5.0

Input:
Enter A: 6
Enter B: 7
Operation: *
Output (Message Box):
Result: 42.0

Input:
Enter A: 20
Enter B: 4
Operation: /
Output (Message Box):
Result: 5.0

Input:
Enter A: 5
Enter B: 0
Operation: /
Output (Message Box):
Error (ZeroDivisionError – should be handled ideally)

Viva Questions

Question	Answer
1. What is the purpose of lambda o=op: calc(o)?	It passes the operator symbol (+, -, *, /) as an argument to the calc() function when the button is clicked.
2. What library is used for GUI programming in this code?	Tkinter, Python's built-in GUI library.
3. What does messagebox.showinfo() do?	Displays a pop-up window showing the result.
4. How are buttons arranged in the window?	Using the grid() layout manager, one row and multiple columns.
5. What happens if non-numeric values are entered?	A ValueError occurs when trying to convert input to float; should be handled using try-except.

Suggested Readings

1. Reema Thareja, *Python Programming*, Oxford University Press.
2. Jeeva Jose, *Introduction to Computing and Problem Solving using Python*, Khanna Publishers.
3. John Zelle, *Python Programming: An Introduction to Computer Science*, Franklin, Beedle & Associates.
4. Mark Lutz, *Learning Python*, O'Reilly Media.
5. Eric Matthes, *Python Crash Course*, No Starch Press.
6. Al Sweigart, *Automate the Boring Stuff with Python*, No Starch Press.
7. Charles Severance, *Python for Everybody*, Charles Severance Foundation.
8. Brad Miller and David Ranum, *Problem Solving with Algorithms and Data Structures using Python*, Franklin, Beedle & Associates.
9. Jake VanderPlas, *Python Data Science Handbook*, O'Reilly Media.
10. Wes McKinney, *Python for Data Analysis*, O'Reilly Media.
11. Burkhard A. Meier, *Python GUI Programming Cookbook*, Packt Publishing.
12. *Official Python Documentation* – https://docs.python.org/3
13. *Real Python Tutorials* – https://realpython.com/
14. *W3Schools Python Reference* – https://www.w3schools.com/python/
15. *GeeksforGeeks Python Programming* – https://www.geeksforgeeks.org/python-programming-language/