



COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous college under VTU)



DEPARTMENT OF COMPUTER & APPLICATIONS

VISION

To develop logical, analytical, and problem-solving skills in students by providing strong foundations in C programming, enabling them to design efficient programs and adapt to advanced computing technologies.

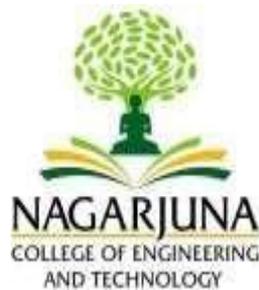
MISSION

- M1:** To impart fundamental knowledge of C programming concepts such as data types, control structures, functions, arrays, pointers, and structures.
- M2:** To enable students to analyze real-world problems and implement effective solutions using structured programming techniques.
- M3:** To enhance coding proficiency through hands-on laboratory exercises, debugging practices, and algorithmic thinking.
- M4:** To prepare students for advanced programming languages, software development, and competitive examinations by strengthening their core programming skills.

NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous College under VTU) Venkatagiri
post, Devanahalli, Bengaluru-562164

Department of Computer & Applications



LAB MANUAL

Semester: I

Course Name: Programming In C Laboratory

Course Code: 25BCL107

NAME OF THE STUDENT _____
USN _____
SEMESTER _____
SECTION _____

Program Outcome:

- **PO1:** Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
 - **PO2:** Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
 - **PO3:** Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
 - **PO4:** Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modeling, analysis & interpretation of data to provide valid conclusions. (WK8).
 - **PO5:** Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modeling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
 - **PO6:** The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, WK7).
 - **PO7:** Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
 - **PO8:** Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
 - **PO9:** Communication: Communicate effectively and inclusively within the community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
 - **P10:** Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
 - **P11:** Life-Long Learning : Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)
-

Program Educational Objectives (PEOs)

- **PEO1:** To progress into the software industry, academia, research, entrepreneurial pursuit, government, consulting firms, and other recent trends in technology-enabled services.
- **PEO2:** To achieve peer recognition of an individual or team; by adopting ethics and professionalism and communicating effectively to perform well in cross-culture and inter-disciplinary teams.
- **PEO3:** To continue lifelong professional development in computing for the benefit of self- and societal growth

PROGRAMMING IN C LABORATORY

PROGRAMMING IN C LABORATORY		SEMESTER	I
Course Code	25BCL107	CIE Marks	50
Teaching Hours/Week (L: P: SDA)	1:2:0	SEE Marks	50
Total Hours of Pedagogy	14 Sessions	Total Marks	100
Credits	02	Exam Hours	03
Type of the Course	Practical		

Course Objectives:

This course will enable students to :

- Write C programs to perform basic operations such as calculations, comparisons, and data manipulation.
- Implement and demonstrate control structures like loops, conditionals, and switch-case statements in C.
- Manipulate arrays and strings, including operations like finding duplicates, reversing, and applying string functions.
- Demonstrate the use of pointers for advanced operations like swapping numbers and matrix manipulation, and work with structures to manage complex data types.

Syllabus

Part A

1. Print the value of y for given x=2 & z=4 and analyze the output.
 - a. `y = x++ + ++x;`
 - b. `y= ++x + ++x;`
 - c. `y= ++x + ++x + ++x;`
 - d. `y = x>z;`
 - e. `y= x>z? x:z;`
 - f. `y = x&z;`
2. Program to read two numbers and find the largest (demonstration on if else).
3. Program to read percentage of marks and to display appropriate message (demonstration of switch case statement).

4. Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers (demonstration of do-while loop).
5. Write a program to print sums of even numbers and sum of odd numbers from array of positive integers (demonstration of 1D array).
6. Program to implement built-in string functions.
7. Program to demonstrate call by value and call by reference.

Part B

1. Program to demonstrate library functions in math.h (demonstration of built-in functions).
2. Program to find the roots of quadratic equation (demonstration of else-if ladder)..
3. Program to read a number, find the sum of the digits, reverse the number and check it for palindrome (demonstration of while loop).
4. Program to generate n Fibonacci sequence (demonstration of for loop).
5. Program to perform addition and subtraction of Matrices (demonstration of 2D array).
6. Program to check a number for prime by defining isprime() function (demonstration of userdefined function).
7. Program to accept USN, Student Name, marks of any 6 Subjects and calculate total marks, Percentage, grade and print the all the details in marks card format of a particular student (demonstration of structure).

Course Outcomes:

After the completion of the Course, students will be able to

- CO1.** Develop the C Program which includes the structure of a C program, Tokens, Expressions, Operators etc.
- CO2.** Demonstrate conditional and iterative statements to write C programs.
- CO3.** Construct C programs that use arrays and strings.
- CO 4.** Design user defined functions to solve real time problems.
- CO5.** Demonstration of Structure concepts & Pointers.

PART-A

1. Print the value of y for given x=2 & z=4 and analyze the output.

a. `y = x++ ++x`

b. `y= ++x ++x;`

c. `y= ++x+++x+++x;`

d. `y = x>z;`

e. `y= x>z? x:z;`

f. `y = x&z;`

```
#include <stdio.h>

#include <conio.h>

#include<stdlib.h>

int main() {

    int x, z, y;

    clrscr();

    // (a) y = x++ + ++x

    x = 2; z = 4;

    y = x++ + ++x;

    printf("(a) y = %d, x = %d\n", y, x);

    // (b) y = ++x + ++x

    x = 2; z = 4;

    y = ++x + ++x;

    printf("(b) y = %d, x = %d\n", y, x);

    // (c) y = ++x + ++x + ++x

    x = 2; z = 4;

    y = ++x + ++x + ++x;

    printf("(c) y = %d, x = %d\n", y, x);
```

```
// (d) y = x > z
x = 2; z = 4;
y = x > z;
printf("(d) y = %d\n", y);

// (e) y = x > z ? x : z
x = 2; z = 4;
y = x > z ? x : z;
printf("(e) y = %d\n", y);

// (f) y = x & z
x = 2; z = 4;
y = x & z;
printf("(f) y = %d\n", y);

getch();
return 0;
}
```

Output

```
(a) y = 6, x = 4
(b) y = 8, x = 4
(c) y = 13, x = 5
(d) y = 0
(e) y = 4
(f) y = 0
```

```
=== Code Execution Successful ===
```

2. Program to read two numbers and find the largest (demonstration on if else).

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int num1, num2;
    // Input two numbers
    printf("Enter first number: ");
    scanf("%d", &num1);
    printf("Enter second number: ");
    scanf("%d", &num2);
    // Compare numbers using if-else
    if(num1 > num2)
    {
        printf("The largest number is: %d\n", num1);
    }
    else if(num2 > num1)
    {
        printf("The largest number is: %d\n", num2);
    }
    else
    {
        printf("Both numbers are equal.\n");
    }
    getch();
    return 0;
}
```

Output

```
Enter first number: 25
Enter second number: 10
The largest number is: 25
```

```
=== Code Execution Successful ===
```

3. Program to read percentage of marks and to display appropriate message (demonstration of switch case statement).

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int main()
{
int per, grade;
clrscr();
printf("Enter percentage of marks: ");
scanf("%d", &per);
    // Step 1: Convert percentage into grade number
if(per >= 90)
    grade = 1;
else if(per >= 80)
    grade = 2;
else if(per >= 70)
    grade = 3;
else if(per >= 60)
    grade = 4;
else if(per >= 50)
    grade = 5;
else
    grade = 6;

    // Step 2: Use switch–case on grade
switch(grade)
{
case 1: printf("Grade: Excellent"); break;
case 2: printf("Grade: Very Good"); break;
case 3: printf("Grade: Good"); break;
case 4: printf("Grade: Average"); break;
```

```
case 5: printf("Grade: Pass"); break;
case 6: printf("Grade: Fail"); break;
}
getch();
return 0;
}
```

Output

```
Enter percentage of marks: 86
Grade: Very Good
```

```
=== Code Execution Successful ===
```

4. Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers (demonstration of do-while loop).

```
#include <stdio.h>
#include <conio.h>
int main() {
    int num;
    int sum = 0;
    clrscr();
    printf("Enter numbers (press 999 to stop):\n");
    do
    {
        scanf("%d", &num);
        if (num > 0 && num != 999)
        {
            sum = sum + num; // add only positive numbers
        }
    }
    while (num != 999); // stop when 999 is entered

    printf("Sum of all positive numbers = %d\n", sum);
    getch();
    return 0;
}
```

Output

```
Enter numbers (press 999 to stop):
345
120
12
5
999
Sum of all positive numbers = 482

=== Code Execution Successful ===
```

5. Write a program to print sums of even numbers and sum of odd numbers from an array of positive integers (demonstration of 1D array).

```
#include <stdio.h>
#include <conio.h>
int main() {
int n, i;
int numbers[100]; // 1D array
int evenSum = 0, oddSum = 0;
printf("Enter how many numbers: ");
scanf("%d", &n);
printf("Enter %d positive integers:\n", n);
for (i = 0; i < n; i++)
{
scanf("%d", &numbers[i]);
}
for (i = 0; i < n; i++)
{
if (numbers[i] % 2 == 0)
evenSum += numbers[i]; // add even numbers
else
oddSum += numbers[i]; // add odd numbers
}
printf("\nSum of Even Numbers = %d", evenSum);
printf("\nSum of Odd Numbers = %d\n", oddSum);
return 0;
}
```

Output

```
Enter how many numbers: 4
Enter 4 positive integers:
2 6 3 7

Sum of Even Numbers = 8
Sum of Odd Numbers = 10

=== Code Execution Successful ===
```

6. Program to implement built-in string functions.

```
#include <stdio.h>
#include <string.h>
//#include <conio.h>
int main() {
char str1[50], str2[50], str3[100];
int len, cmp;
printf("Enter first string: ");
scanf("%s",str1); // reads a string including spaces
printf("Enter second string: ");
scanf("%s",str2);
    // 1. Length of first string
    len = strlen(str1);
    printf("\nLength of first string = %d", len);

    // 2. Copy string 1 to string 3
    strcpy(str3, str1);
    printf("\nCopied string = %s", str3);

    // 3. Concatenate string 1 and string 2
    strcat(str1, str2);
    printf("\nConcatenated string = %s", str1);

    // 4. Compare string 1 and string 2
    cmp = strcmp(str2, str3);
    if (cmp == 0)
        printf("\nStrings are equal");
    else
        printf("\nStrings are not equal");

return 0;
```

Output

```
Enter first string: mounisha
Enter second string: Chandra

Length of first string = 8
Copied string = mounisha
Concatenated string = mounishaChandra
Strings are not equal

} === Code Execution Successful ===
```

7. Program to demonstrate call by value and call by reference.

```
#include <stdio.h>

#include <conio.h>

// Function declaration

void swapByValue(int a, int b);

void swapByReference(int *x, int *y);

int main() {

    int num1, num2;

    printf("Enter two numbers: ");

    scanf("%d %d", &num1, &num2);

    printf("\nBefore swapping: num1 = %d, num2 = %d", num1, num2);

    // Call by Value

    swapByValue(num1, num2);

    printf("\nAfter Call by Value: num1 = %d, num2 = %d", num1, num2);

    // Call by Reference

    swapByReference(&num1, &num2);

    printf("\nAfter Call by Reference: num1 = %d, num2 = %d\n", num1, num2);

    return 0;

}
```

```

// Function definition – Call by Value

void swapByValue(int a, int b) {

    int temp;

    temp = a;

    a = b;

    b = temp;

    printf("\nInside swapByValue: a = %d, b = %d", a, b);

}

// Function definition – Call by Reference

void swapByReference(int *x, int *y) {

    int temp;

    temp = *x;

    *x = *y;

    *y = temp;

    printf("\nInside swapByReference: x = %d, y = %d", *x, *y);

}

```

Output

```

Enter two numbers: 2 4

Before swapping: num1 = 2, num2 = 4
Inside swapByValue: a = 4, b = 2
After Call by Value: num1 = 2, num2 = 4
Inside swapByReference: x = 4, y = 2
After Call by Reference: num1 = 4, num2 = 2

=== Code Execution Successful ===

```

PART B

1. Program to demonstrate library functions in math.h

```
#include <stdio.h>
#include <math.h>
int main() {

double num = 25.0;

printf("Square root of %.2lf = %.2lf\n", num, sqrt(num));   printf("Power (2^3)
= %.2lf\n", pow(2, 3));
printf("Ceil of 3.2 = %.2lf\n", ceil(3.2));
printf("Floor of 3.8 = %.2lf\n", floor(3.8));   printf("Absolute
value of -10 = %.2lf\n", fabs(-10));

return 0;
}
```

Output

```
Square root of 25.00 = 5.00
Power (2^3) = 8.00
Ceil of 3.2 = 4.00
Floor of 3.8 = 3.00
Absolute value of -10 = 10.00
```

```
=== Code Execution Successful ===
```

2. Program to find the roots of a quadratic equation (else-if ladder)

```
#include <stdio.h>

#include <math.h>
int main() {

float a, b, c, d, root1, root2;   printf("Enter
coefficients a, b, c: ");   scanf("%f %f %f",
&a, &b, &c);

d = b*b - 4*a*c;

if (d > 0) {
    root1 = (-b + sqrt(d)) / (2*a);
    root2 = (-b - sqrt(d)) / (2*a);
    printf("Roots are real and distinct: %.2f , %.2f", root1, root2);
}
else if (d == 0) {
    root1 = -b / (2*a);
    printf("Roots are real and equal: %.2f , %.2f", root1, root1); }
else {
    printf("Roots are imaginary.");
}
return 0;
}
```

Output

```
Enter coefficients a, b, c: 2 6 9
Roots are imaginary.
```

```
=== Code Execution Successful ===
```

Output

```
Enter coefficients a, b, c: 2 3 1
Roots are real and distinct: -0.50 , -1.00
```

```
=== Code Execution Successful ===
```

3. Program to find sum of digits, reverse number and check palindrome (while loop)

```
#include <stdio.h>
int main() {

    int num, temp, rem, sum = 0, rev = 0;

    printf("Enter a number: ");

    scanf("%d", &num);

    temp = num;

    while (temp != 0) {

        rem = temp % 10;

        sum += rem;

        rev = rev * 10 + rem;

        temp/= 10;

    }

    printf("Sum of digits = %d\n", sum);

    printf("Reversed number = %d\n", rev);

    if (rev == num)

        printf("The number is a palindrome.");

    else

        printf("The number is not a palindrome.");

    return 0;
}
```

Output

```
Enter a number: 1 2 1
Sum of digits = 1
Reversed number = 1
The number is a palindrome.
```

=== Code Execution Successful ===

Output

```
Enter a number: 321
Sum of digits = 6
Reversed number = 123
The number is not a palindrome.
```

=== Code Execution Successful ===

4. Program to generate Fibonacci sequence (for loop)

```
#include <stdio.h>
int main() {
int i, n;

// initialize first and second terms
int t1 = 0, t2 = 1;

// initialize the next term (3rd term)
int nextTerm = t1 + t2;

// get no. of terms from user
printf("Enter the number of terms: ");
scanf("%d", &n);

// print the first two terms t1 and t2
printf("Fibonacci Series: %d, %d, ", t1, t2);

// print 3rd to nth terms
for (i = 3; i <= n; ++i) {
printf("%d, ", nextTerm);
t1 = t2;
t2 = nextTerm;
nextTerm = t1 + t2;
}
return 0;
}
```

Output	Output
Enter the number of terms: 5 Fibonacci Series: 0, 1, 1, 2, 3;	Enter the number of terms: 6 Fibonacci Series: 0, 1, 1, 2, 3, 5;
=== Code Execution Successful ===	=== Code Execution Successful ===

5. Program for addition and subtraction of matrices (2D array)

```
#include <stdio.h>
```

```
int main() {
```

```
int a[10][10], b[10][10], sum[10][10], sub[10][10];
```

```
int r, c, i, j;
```

```
printf("Enter number of rows and columns: ");
```

```
scanf("%d %d", &r, &c);
```

```
printf("Enter elements of first matrix:\n");
```

```
for (i=0; i<r; i++)
```

```
    for (j=0; j<c; j++)
```

```
        scanf("%d", &a[i][j]);
```

```
printf("Enter elements of second matrix:\n");
```

```
for (i=0; i<r; i++)
```

```
    for(j=0;j<c;j++)
```

```
        scanf("%d", &b[i][j]);
```

```
printf("\nSum of matrices:\n");
```

```
for (i=0; i<r; i++) {
```

```
    for (j=0; j<c; j++) {
```

```
        sum[i][j] = a[i][j] + b[i][j];
```

```
        printf("%d\t", sum[i][j]);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
printf("\nSubtraction of matrices:\n");
```

```
for (i=0; i<r; i++) {
```

```
    for (j=0; j<c; j++) {
```

```
        sub[i][j] = a[i][j] - b[i][j];
```

```
        printf("%d\t", sub[i][j]);
```

```
    }
```

```
    printf("\n");  
}  
  
return 0;  
}
```

```
Output  
Enter number of rows and columns: 2  
2  
Enter elements of first matrix:  
2  
3 4 1  
  
Enter elements of second matrix:  
3 4 6 7  
  
Sum of matrices:  
5 7  
10 8  
  
Subtraction of matrices:  
-1 -1  
-2 -6  
  
=== Code Execution Successful ===
```

6. Program to check if a number is prime (user-defined function)

```
#include <stdio.h>
int isPrime(int n) {
    int i;

    if (n <= 1)
        return 0;

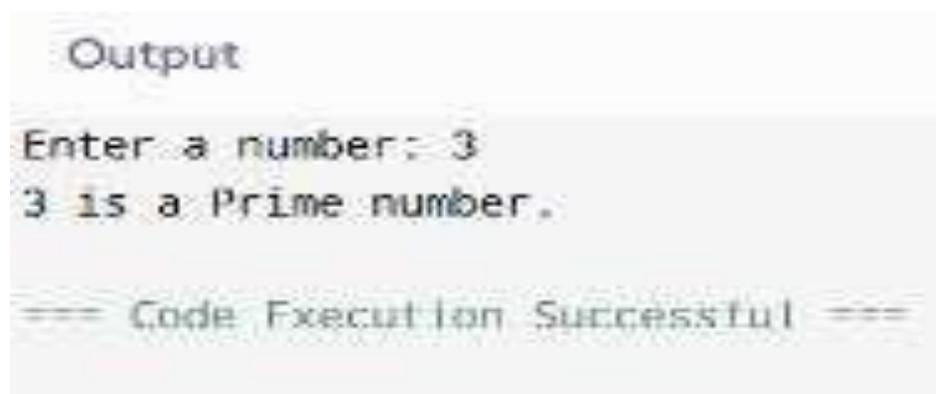
    for (i = 2; i <= n/2; i++) {
        if (n % i == 0)
            return 0;
    }
    return 1;
}

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    if (isPrime(num))
        printf("%d is a Prime number.", num);
    else
        printf("%d is not a Prime number.", num);

    return 0;
}
```



```
Output
Enter a number: 3
3 is a Prime number.
=== Code Execution Successful ===
```

7. Program to accept USN, Student Name, marks of any 6 Subjects and calculate total marks, Percentage, grade and print the all the details in marks card format of a particular student (demonstration of structure).

```
#include <stdio.h> struct
Student {
char usn[20];
char name[50];
int marks[6];
int total;
float percentage;
char grade;
};

int main() {
struct Student s;

int i;
printf("Enter USN: ");
scanf("%s", s.usn);
printf("Enter Name: ");
scanf("%s", s.name);
s.total = 0;
printf("Enter marks of 6 subjects: ");
for (i = 0; i < 6; i++) {
scanf("%d", &s.marks[i]);
s.total += s.marks[i];
}
s.percentage = s.total / 6.0;
if (s.percentage >= 85)
s.grade = 'A';
```

```
else if (s.percentage >= 70)
    s.grade = 'B';
else if (s.percentage >= 55)
    s.grade = 'C';
else if (s.percentage >= 40)
    s.grade = 'D';
else
    s.grade = 'F';

printf("\n----- MARKS CARD ----- \n");
printf("USN: %s\n", s.usn);
printf("Name: %s\n", s.name);
printf("Total Marks: %d\n", s.total);
printf("Percentage: %.2f\n", s.percentage);
printf("Grade: %c\n", s.grade);

return 0;
}
```

Output

```
Enter USN: 45
Enter Name: mouni
Enter marks of 6 subjects: 89 99 86 93 82 79

----- MARKS CARD -----
USN: 45
Name: mouni
Total Marks: 528
Percentage: 88.00
Grade: A

=== Code Execution Successful ===
```