

**MRCET CAMPUS**

# **MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY** **(AUTONOMOUS INSTITUTION - UGC, GOVT. OF INDIA)**

Affiliated to JNTUH; Approved by AICTE, NBA-Tier 1 & NAAC with A-GRADE | ISO 9001:2015  
Maisammaguda, Dhulapally, Komapally, Secunderabad - 500100, Telangana State, India

## **LABORATORY MANUAL & RECORD**

Name: .....

Roll No: ..... Branch: .....

Year: ..... Sem: .....







**MRCET CAMPUS**

# **MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY** **(AUTONOMOUS INSTITUTION - UGC, GOVT. OF INDIA)**

Affiliated to JNTUH; Approved by AICTE, NBA-Tier 1 & NAAC with A-GRADE | ISO 9001:2015  
Maisammaguda, Dhulapally, Komapally, Secunderabad - 500100, Telangana State, India

## **Certificate**

Certified that this is the Bonafide Record of the Work Done by  
Mr./Ms.....Roll.No.....of  
B.Tech I year ..... Semester for Academic year 2023 - 2024  
in.....Laboratory.

Date:

Faculty Incharge

HOD

Internal Examiner

External Examiner



# INDEX

[illegible]



## PROGRAM OUTCOMES

A B.Tech –graduate should possess the following program outcomes.

- 1 **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
- 2 **Problem analysis:** Identify ,formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3 **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4 **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5 **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6 **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7 **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8 **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9 **Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams ,and in multi disciplinary settings.
- 10 **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large ,such as, being able to comprehend and write effective reports and design documentation ,make effective presentations ,and give and receive clear instructions.
- 11 **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.
- 12 **Lifelong learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**I Year B. Tech-I Sem**

**L/ T/P/C**

-/ - /3/1.5

**(R22A0581) PROGRAMMING FOR PROBLEM SOLVING LAB**

**COURSE OBJECTIVES:**

1. To work with an IDE to create, edit, compile, run and debug programs.
2. To analyze the various steps in program development.
3. To develop programs to solve basic problems by understanding basic concepts in C like operators, control statements etc.
4. To develop modular, reusable and readable C Programs using the concepts like functions, arrays etc.
5. To create, read from and write to text and binary files.

**Week 1:**

- a) Write a program to find sum and average of three numbers.
- b) Write a program to calculate simple interest (SI) for a given principal (P), time (T), and rate of interest (R) ( $SI = P \cdot T \cdot R / 100$ ).

**Week 2:**

- a) Write a program to swap two variable values with and without using third variable.
- b) Write a program to find the roots of a quadratic equation.

**Week 3:**

- a) Write a program to find the sum of individual digits of a given positive integer.
- b) Write a program, which takes two integer operands and one operator from the user, performs the operation and then prints the result.  
(Consider the operators +, -, \*, /, % and use Switch Statement)

**Week 4:**

- a) Write a program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
- b) A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.

**Week 5:**

- a) Write a program to find both the largest and smallest number in a list of integers.
- b) Write a program to find the sum of integer array elements.

**Week 6:**

- a) Write a program to perform addition of two matrices.
- b) Write a program to perform multiplication of two matrices.



**Week 7:**

- a) Write a function that returns the sum of two numbers.
- b) Write a function to find the factorial of a given integer.

**Week 8:**

- a) Write a function to swap the values of two variables using call by value.
- b) Write a function to swap the values of two variables using call by reference.

**Week 9:**

- a) Write a recursive function to find the factorial of a given number.
- b) Write a recursive function to find GCD of two integers.

**Week 10:**

- a) Write a program to check whether the given string is palindrome or not.
- b) Write a program to sort the given names in alphabetical order.

**Week 11:**

- a) Write a program to find the sum of integer array elements using pointers.
- b) Write a program to find the length of the given string using pointers.

**Week 12:**

- a) Write a program to create a structure named book and display the contents of a book.
- b) Write a program to calculate total and percentage marks of a student using structure.

**Week 13:**

- a) Write a program to create a structure Complex and use functions to perform the following operations:
  - i) Reading a complex number
  - ii) Writing a complex number
  - iii) Addition of two complex numbers
  - iv) Multiplication of two complex numbers
- b) Write a program to write given text to a file and also display the contents of a file.

**Week 14:**

- a) Write a program to copy the contents of one file to another.
- b) Write a program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third).

**TEXT BOOKS**

1. Mastering C, K.R.Venugopal and S.R.Prasad, TMH Publishers.
2. C Programming, E. Balagurusamy, 3rd edition, TMHPublishers.
3. Computer programming in C.V.RAJaraman, PHI Publishers.
4. C Programming, M.V.S.S.N Venkateswarlu and E.V.Prasad,S.Chand Publishers

**COURSE OUTCOMES:**

After completion of the course, Students will be able to:

1. Formulate the algorithms for simple problems.
2. Translate given algorithms to a working and correct program.
3. Identify and correct logical errors encountered during execution.
4. Represent and manipulate data with arrays, strings and structures and pointers.
5. Create, read and write to and from simple text and binary file

## CONTENTS

Week No.	S.No./ Program No.	List of Programs
1)	A)	Write a program to find sum and average of three numbers
	B)	Write a program to calculate simple interest (SI) for a given principal (P), time (T) ,and rate of interest (R) ( $SI = P \cdot T \cdot R / 100$ )
2)	A)	Write a program to swap two variables values with and without using third variable
	B)	Write a program to find the roots of a quadratic equation.
3)	A)	Write a program to find the sum of individual digits of a given positive integer.
	B)	Write a program, which takes two integer operands and one operator from the user, performs the operation and then prints the result.(Consider the operators +,-,*,/, % and use Switch Statement)
4)	A)	Write a program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
	B)	Write a C program to generate the first n terms of the sequence.
5)	A)	Write a program to find both the largest and smallest number in a list of integers.
	B)	Write a program to find the sum of integer array elements.
6)	A)	Write a program to perform addition of two matrices.
	B)	Write a program to perform multiplication of two matrices.
7)	A)	Write a function that returns the sum of two numbers.
	B)	Write a function to find the factorial of a given integer.
8)	A)	Write a function to swap the values of two variables using call by value.
	B)	Write a function to swap the values of two variables using call by reference.
9)	A)	Write a recursive function to find the factorial of a given number.
	B)	Write a recursive function to find GCD of two integers.
10)	A)	Write a program to check whether the given string is palindrome or not.
	B)	Write a program to sort the given names in alphabetical order.

11)	A)	Write a program to find the sum of integer array elements using pointers.
	B)	Write a program to find the length of the given string using pointers.
12)	A)	Write a program to create a structure named book and display the contents of a book.
	B)	Write a Program to Calculate Total and Percentage marks of a student using structure.
13)	A)	Write a program that uses functions to perform the following operations: i) Reading a complex number ii) Writing a complex number iii) Addition of two complex numbers iv) Multiplication of two complex numbers
	B)	Write a program to write given text to a file and also display the contents of a file.
14)	A)	Write a program to copy the contents of one file to another.
	B)	Write a program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third.

## **INSTRUCTIONS TO STUDENTS**

1. Students should bring lab Manual/Record for every laboratory session and should enter the readings /observations in the manual while performing the experiment.
2. The group- wise division made in the beginning should be adhered to, and no mix up of students among different groups will be permitted later.
3. The components required pertaining to the experiment should be collected from stores in –charge after duly filling in the requisition form.
4. When the experiment is completed, students should disconnect the setup made by them, and should return all the components/instruments taken for the purpose.
5. Any damage to the apparatus that occurs during the experiment should be brought to the notice of lab in-charge, consequently, the cost of repair or new apparatus should be brought by the students.
6. After completion of the experiment, certification of the concerned staff in –charge in the observation book is necessary.
7. Students should be present in the labs for the total scheduled duration.
8. Students should not carry ant food items inside the laboratory.
9. Use of cell phones and IPODs is forbidden.
10. Students should not write on or deface any lab desks, computers, or any equipment provided to them during the experiment.
11. Every student should keep his/her work area properly before leaving the laboratory.

### **What are Computer Programming Languages?**

Computer programming languages allow us to give instructions to a computer in a language the computer understands. Just as many human-based languages exist, there are an array of computer programming languages that programmers can use to communicate with a computer. The portion of the language that a computer can understand is called a "binary." Translating programming language into binary is known as "compiling." Each language, from C Language to Python, has its own distinct features, though many times there are commonalities between programming languages.

### **What is a Compiler?**

A compiler is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses. Typically, a programmer writes language statements in a language such as Pascal or C one line at a time using an editor. The file that is created contains what are called the source statements. The programmer then runs the appropriate language compiler, specifying the name of the file that contains the source statements.



### **Why use a Compiler?**

- Compiler verifies entire program, so there are no syntax or semantic errors
- The executable file is optimized by the compiler, so it executes faster
- Allows you to create internal structure in memory
- There is no need to execute the program on the same machine it was built
- Translate entire program in other language
- Generate files on disk
- Link the files into an executable format
- Check for syntax errors and data types
- Helps you to enhance your understanding of language semantics
- Helps to handle language performance issues
- Opportunity for a non-trivial programming project
- The techniques used for constructing a compiler can be useful for other purposes as well

C programming language was developed in 1972 by Dennis Ritchie at bell laboratories of AT&T (American Telephone & Telegraph), located in the U.S.A. Dennis Ritchie is known as the founder of the c language. It was developed to overcome the problems of previous languages such as B, BCPL, etc.

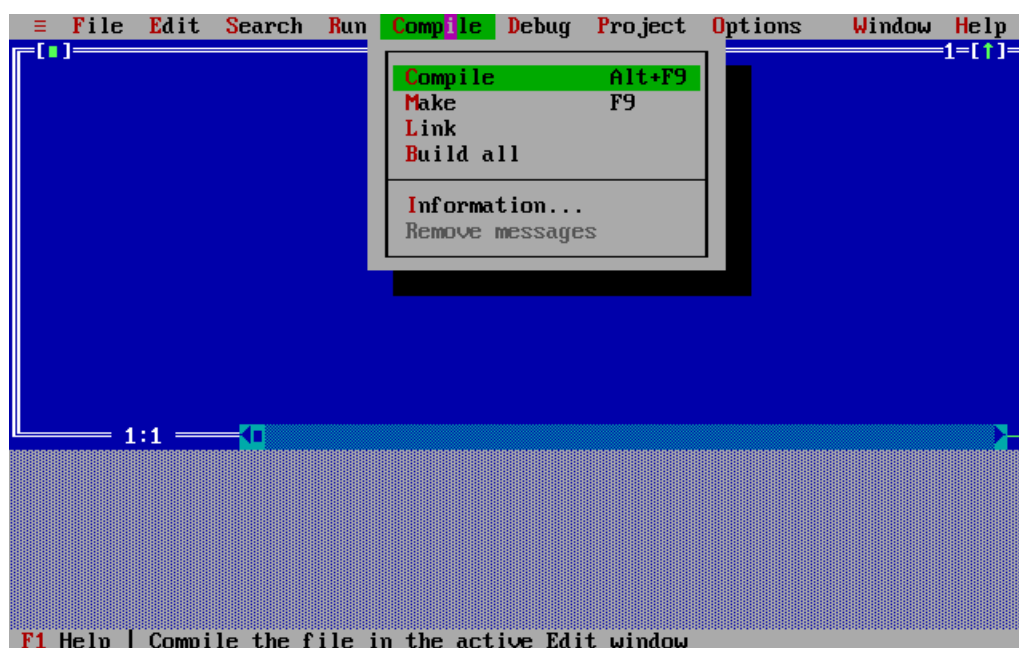
## Standardization of C language

Language	Year	Developed By
Algol	1960	International Group
BCPL	1967	Martin Richard
B	1970	Ken Thompson
Traditional C	1972	Dennis Ritchie
K & R C	1978	Kernighan & Dennis Ritchie
ANSI C	1989	ANSI Committee
ANSI/ISO C	1990	ISO Committee
C99	1999	Standardization Committee

Widely used compilers are

1. **Turbo C (16 bit compiler)**
2. **GCC based compilers (32 bit compiler)**

## 1. Compilation Process in Turbo C In Windows Operating System

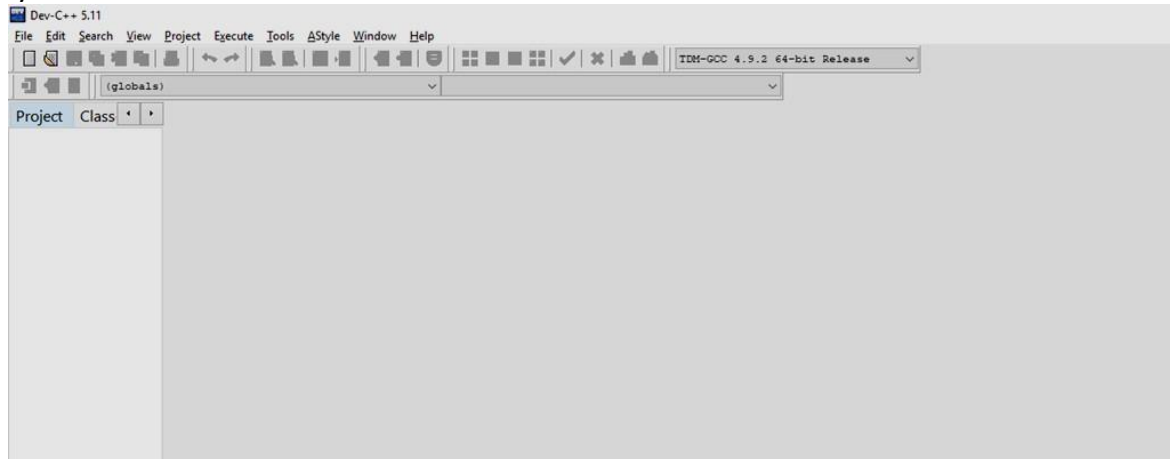


Open Turbo C Editor.

1. Select **"File"** from Menu bar and select option **"New"**
2. Save C program in filename .C extension.
3. To do compiling – Select -> **Compile** from menu and click-> **compile**.  
If the compilation is success – you will see a **"success"** message. Else you will see the number of errors.
4. To RUN the program – you may select -> **Run** from menu and click -> **Run**  
Now you will see the output screen.

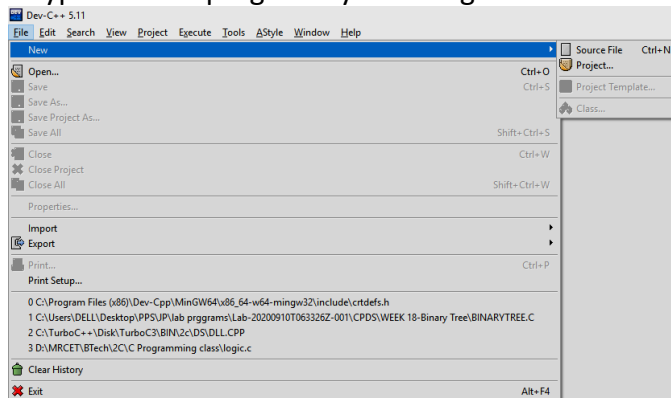
## 2. Compilation of C program in DEV C++ editor in Windows

Dev-C++ is a free full-featured integrated development environment (IDE) distributed under the GNU General Public License for programming in C and C++ for windows operating system.

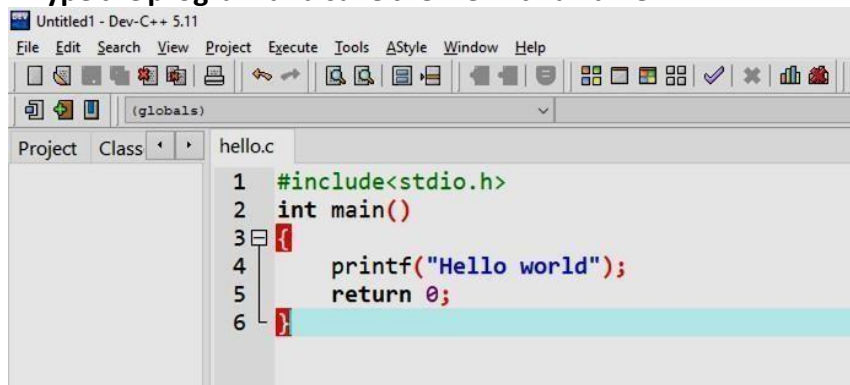


Procedure for compilation and execution:

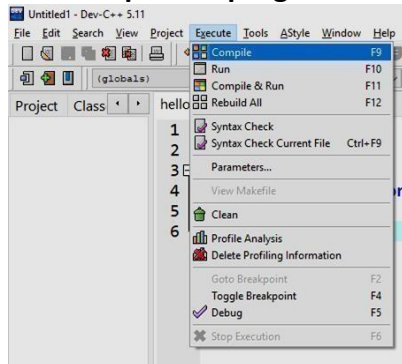
### 1. Type the new program by selecting new source file from file menu



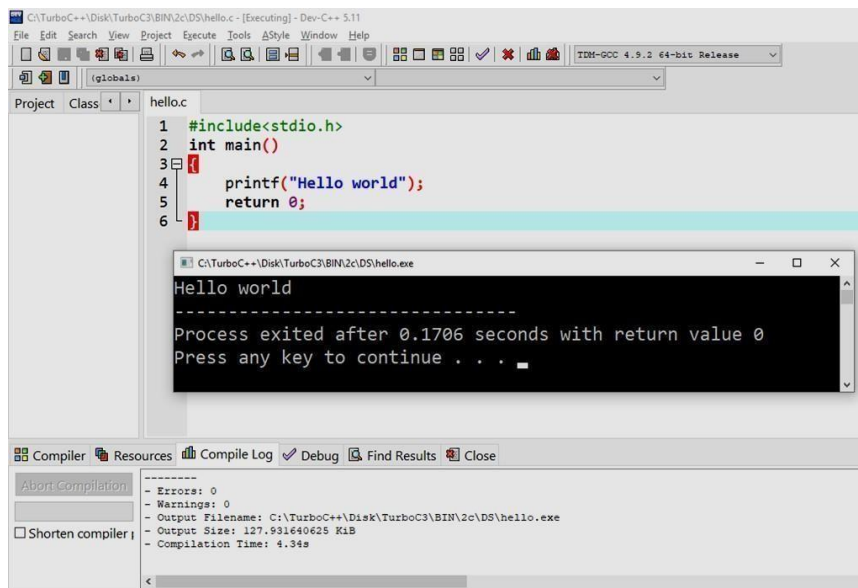
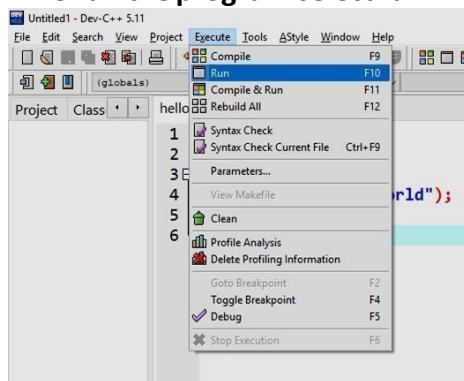
### 2. Type the program and save the file with a name



### 3. To compile the program select compile from execute menu or press F9



### 4. To run the program select run from execute menu or press F10





### 3. Compiling C program using GCC in Linux

Released by the Free Software Foundation. gcc is a Linux-based C compiler usually operated via the command line. It often comes distributed with a Linux installation.

Compilation process of a C program

#### **\$gcc filename**

The default executable output of gcc is "a.out",

Running the output file

#### **\$/a.out**

It is also possible to specify a name for the executable file at the command line by using the syntax -o outfile , as shown in the following example : -

gcc filename -o outfile

### **Program execution in Linux Environment using vi editor**

1.To Type the program use vi editor

[jayapalmedida@webminal.org ~]\$vi sample.c

To Type the program press i

```
#include<stdio.h>
int main()
{
    printf("Hello");
    return 0;
}
```

To save the program

- Press ESC
- press :wq (save and quit)

### **2. Compilation**

[jayapalmedida@webminal.org ~]\$gcc sample.c -o output

### **3. Running the program**

[jayapalmedida@webminal.org ~]\$./output

Hello

### **Program execution in Linux Environment using gedit editor**

1.To Type the program use gedit editor

[jayapalmedida@webminal.org ~]\$gedit sample.c

```
#include<stdio.h>
int main()
{
    printf("Hello");
    return 0;
}
```

Type the program .

Save the program .

Close the editor.

## 2. Compilation

[jayapalmedida@webminal.org ~]\$gcc sample.c -o output

## 3. Running the program

[jayapalmedida@webminal.org ~]\$. /output  
hello

**Week 1****1. a) Write a program to find the sum and average of three numbers.**

**Aim:** Find the sum and average of three numbers

**Algorithm:**

Step 1: Start

Step 2: Read values num1, num2, num3

Step 3: Add num1, num2, num3 and assign the result to sum.

$sum \leftarrow num1 + num2 + num3$

$average \leftarrow sum / 3$

Step 4: Display sum and average

Step 5: Stop

**Program:**

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

```
int main( )
```

```
{
```

```
    int a,b,c;
```

```
    int sum,average;
```

```
    printf("Enter any three integers: ");
```

```
    scanf("%d%d %d",&a,&b,&c);
```

```
    sum = a+b+c;
```

```
    average=sum/3;
```

```
    printf("Sum and average of three integers: %d %d",sum,average);
```

```
    return 0;
```

```
}
```

**SAMPLE INPUT:**

Enter any three integers: 2 4 5

**EXPECTED OUTPUT:**

Sum and average of three integers: 11 3

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

1. b) Write a program to calculate simple interest(SI) for a given principal (P), time (T), and rate of interest (R)  $(SI = P \cdot T \cdot R / 100)$

Aim: To find the simple interest

**Algorithm:**

Step 1: Start.

Step 2 : Read Principal Amount, rate and time.

Step 3 : **Calculate Interest** using formula  $SI = ((\text{amount} \cdot \text{rate} \cdot \text{time}) / 100)$

Step 4 : Print **Simple Interest SI**.

Step 5 : Stop

**Program:**

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

```
int main()
```

```
{
```

```
    float p, r, t, si;
```

```
    printf("Input principle:");
```

```
    scanf("%f", &p);
```

```
    printf("Rate of interest:");
```

```
    scanf("%f", &r);
```

```
    printf("Enter time(in years):");
```

```
    scanf("%f", &t);
```

```
    si=(p*r*t)/100;
```

```
    printf("Simple interest = %f", si);
```

```
    return 0;
```

```
}
```

**SAMPLE INPUT:**

Input principle: 10000

Rate of interest: 12

Enter time(in years): 2

**EXPECTED OUTPUT:**

Simple interest = 2400

**Record at least 2 results****Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐**Signature of faculty with date****Exercise**

1. Write a program to find distance when initial velocity, acceleration and time is given.
2. Write a program to find compound interest.
3. Write a program to find amount of memory required by different types of variables.
4. Write a program to evaluate algebraic expression  $(ax+b)/(ax-b)$ .
5. Write a program to find area and perimeter of circle.



**RECORD NOTES**



**RECORD NOTES**

**RECORD NOTES**

**Week 2****2 a)Write a program to swap two variables values with and without using third variable**

**AIM:** To swap two variable values using a third variable

**DESCRIPTION:**

Swap the values of the variable using temporary variable

t = a

a = b

b = t

**ALGORITHM:**

➤ **using a third variable**

Step 1 : Start

Start 2 : READ num1, num2

Start 3 : temp = num1

Start 4 : num1 = num2

Start 5 : num2 = temp

Start 6 : PRINT num1, num2

Start 7 : Stop

➤ **without using a third variable**

Step 1 : Start

Start 2 : READ num1, num2

Start 3 : num1 = num1 + num2

Start 4 : num2 = num1 - num2

Start 5 : num1 = num1 - num2

Start 6 : PRINT num1, num2

Start 7 : Stop

**PROGRAM:**

using a third variable

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

```
int main()
```

```
{
```

```
    int x, y, t;
```

```
        printf("Enter two integers: ");
```

```
        scanf("%d%d", &x, &y);
```

```
        printf("Before Swapping\nFirst integer = %d\nSecond integer = %d\n", x, y);
```

```
        t = x;
```

```
        x = y;
```

```
        y = t;
```

```
        printf("After Swapping\nFirst integer = %d\nSecond integer = %d\n", x, y);
```

```
    return 0;
```

```
}
```

**SAMPLE INPUT:**

Enter two integers: 10 20

**EXPECTED OUTPUT:**

Before Swapping

First integer = 10

Second integer = 20

After Swapping

First integer = 20

Second integer = 10

**Record at least 2 results****Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**RECORD NOTES**

**PROGRAM: Without using a third variable**

```
#include <stdio.h>
int main()
{
    int a, b;

    printf("Enter two integers: ");
    scanf("%d%d", &a, &b);
    printf("Before Swapping\nFirst integer = %d\nSecond integer = %d\n", a, b);
    a = a + b;
    b = a - b;
    a = a - b;
    printf("After Swapping\nFirst integer = %d\nSecond integer = %d\n", a, b);
    return 0;
}
```

**SAMPLE INPUT:**

Enter two integers: 23 45

**EXPECTED OUTPUT:**

Before Swapping  
First integer = 23  
Second integer = 45  
After Swapping  
First integer = 45  
Second integer = 23

**Record at least 2 results**

**Assessment**

Not Satisfactory

Needs Improvement

Partially Executed

Executed Successfully

☐☐☐☐

**Signature of faculty with date**

2 b) Write a program to find the roots of a quadratic equation.

**AIM:** To find the roots of a quadratic equation.

**Description:** roots of quadratic equation are  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

**ALGORITHM:**

Step 1: Start

Step 2: Read a,b,c

Step 3: calculate disc =  $b*b-4*a*c$

Step 4: if(disc>0)

    Begin

Step 5: root1= $(-b+\sqrt{\text{disc}})/(2*a)$

Step 6: root2= $(-b-\sqrt{\text{disc}})/(2*a)$

Step 7: Print "Root1" , "Root2"

    End

Step 8: else if(disc=0)

    Begin

Step 9: root1= $-b/(2*a)$

Step 10: root2=root1;

Step 11: Print "Root1" , "Root2"

    End

Step 12: else

Step 13: Print Roots are imaginary

Step 14: Stop

**PROGRAM:**

```
#include<stdio.h>
#include<math.h>
int main()
{
int a,b,c;
float disc, root1, root2;
printf("ENTER VALUES FOR a,b,c:\n");
scanf("%d%d%d",&a,&b,&c);
disc=(float)b*b-4*a*c;
if(disc>0)
{
printf("THE ROOTS ARE REAL & UNEQUAL:\n");
root1=(-b+sqrt(disc))/(2*a);
root2=(-b-sqrt(disc))/(2*a);
printf("Root1=%f\n",root1);
printf("Root2=%f\n",root2);
}
else if(disc==0)
{
printf("THE ROOTS ARE REAL AND EQUAL:\n");
root1=-b/(2*a);
root2=root1;
printf("Root1=%f\n",root1);
printf("Root2=%f\n",root2);
}
else
{
printf("THE ROOTS ARE IMAGINARY:\n");
}
return 0;
}
```

**SAMPLE INPUT:**

```
ENTER VALUES FOR a, b, c
1      4      4
```

**EXPECTED OUTPUT:**

```
THE ROOTS ARE EQUAL AND THEY ARE..      Root1=-2      Root2=-2
```



Record at least 2 results

**Assessment**

Not Satisfactory

Needs Improvement

Partially Executed

Executed Successfully

☐☐☐☐**Signature of faculty with date****Exercise:**

- 1) Write a program to check whether the entered year is leap year or not (a year is leap if it is divisible by 4 and divisible by 100 or 400)
- 2) Write a program to check whether given number is even or odd
- 3) Write a program to find largest of three numbers
- 4) Write a program to calculate percentage and grade of a student given marks in 5 subjects

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**



**Week 3**

3 a) Write a C program to find the sum of individual digits of a given positive integer.

**AIM:** To find the sum of individual digits of positive integer.

**Description:**

Summation of digits of a number

Ex: 1234

Summation =  $1+2+3+4=10$

**ALGORITHM:**

Step 1: Start

Step 2: Read n

Step 3: Initialize  $\text{sum} \leftarrow 0$

Step 4: while( $n \neq 0$ )

    Begin

Step 5:  $r \leftarrow n \% 10$

Step 6:  $\text{sum} \leftarrow \text{sum} + r$

Step 7:  $n \leftarrow n / 10$

    End

Step 8: Print "sum"

Step 9: Stop

**PROGRAM:**

```
#include<stdio.h>
int main()
{
    int n,r,sum=0;
    printf("ENTER A POSITIVE INTEGER \n");
    scanf("%d",&n);
    while(n!=0)
    {
        r=n%10;
        sum=sum+r;
        n=n/10;
    }
    printf("THE SUM OF INDIVIDUAL DIGITS OF A POSITIVE INTEGER IS..%d", sum);
    return 0;
}
```

**SAMPLE INPUT:**

ENTER A POSITIVE INTEGER  
5321

**EXPECTED OUTPUT:**

THE SUM OF INDIVIDUAL DIGITS OF A POSITIVE INTEGER IS..11

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

3 b) Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,\*, /, % and use Switch Statement)

**AIM:**To perform arithmetic operations using switch statement.

**Algorithm:**

Step 1: Read a,b

Step 2: Print "Menu Options"

Step 3: do

Begin Step 4: Read ch

Step 5: switch(ch)

Begin Step 6:

case '+': Begin

Calculate  $c = a + b$

Print "c"

break;

End

case '-': Begin

Calculate  $c = a - b$

Print "c"

break;

End

case '\*':Begin

Calculate  $c = a * b$

Print "c"

break;

End

case '/':Begin

Calculate  $c = a / b$

Print "c"

break;

End

case '%':Begin

Calculate  $c = a \% b$

Print "c"

break;

End

default:

Print "Invalid choice"

End



**Program:**

```
#include<stdio.h>
int main()
{
    int a,b,c;
    char ch,t;
    printf("ENTER TWO VALUES FOR a & b\n");
    scanf("%d %d",&a,&b);
    scanf("%c",&t); // to skip the newline character
    printf("MENU OPTIONS \n");
    printf("*****\n");
    printf("Addition\n");
    printf("Subtraction\n");
    printf("Multiplication\n");
    printf("Division\n");
    printf("Modulus\n");
    printf("\n");
    printf("ENTER Operator : \n");
    scanf("%c",&ch);
    switch(ch)
    {
        case '+':c=a+b;
            printf("The addition of %d and %d is..%d\n",a,b,c); break;
        case '-':c=a-b;
            printf("The subtraction of %d and %d is..%d\n",a,b,c); break;
        case '*':c=a*b;
            printf("The multiplication of %d and %d is..%d\n",a,b,c); break;
        case '/':c=a/b;
            printf("The division of %d and %d is..%d\n",a,b,c); break;
        case '%':c=a%b;
            printf("The modulus of %d and %d is..%d\n",a,b,c); break;
        default:printf("INVALID CHOICE\n"); }
    }
    return 0;
}
```

**SAMPLE INPUT:**

ENTER TWO VALUES FOR a & b: 20 16

**EXPECTED OUTPUT:**

MENU OPTIONS

\*\*\*\*\*

Addition

Subtraction

Multiplication

Division

Modulus

ENTER Operator : +

The addition of 20 and 16 is..36

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise:**

- 1) Write a program to check whether given number is palindrome or not.
- 2) Write a program to check whether given number is prime number or not.
- 3) Write a program to check for a vowel using switch statement.









**Week 4**

4 a) Write a C program to generate the first n terms of the fibonacci sequence.

**AIM:** To generate first n terms of fibonacci sequence

**Algorithm:**

Step 1 → Read n  
Step 2 → Set A = 0, B = 1  
Step 3 → DISPLAY A, B  
Step 4 → C = A + B  
Step 5 → DISPLAY C  
Step 6 → Set A = B, B = C  
Step 7 → REPEAT from 4 - 6, for n times  
Step 8 → STOP

**Program:**

```
#include <stdio.h>
int main()
{
    int i, n;
    int t1 = 0, t2 = 1;
    int nextTerm;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    printf("Fibonacci Series: %d, %d, ", t1, t2);
    for (i = 3; i <= n; ++i)
    {
        nextTerm = t1 + t2;
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
    }
    return 0;
}
```

**SAMPLE INPUT:**

Enter the number of terms: 8

**EXPECTED OUTPUT:**

Fibonacci Series: 0 1 1 2 3 5 8 13

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**



4 b) Write a program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

**AIM:** To generate prime numbers between 1 to N.

**ALGORITHM:**

**Step 1** – Read n value.

**Step 2** – Initialize count = 0

**Step 3** – for i = 2 to n

a. for j = 1 to i

b. if i % j = 0

c. then increment count

d. if count is equal to 2

e. then print i value

```
#include<stdio.h>
int main()
{
    int i, num, n, count;
    printf("Enter the range: ");
    scanf("%d", &n);
    printf("The prime numbers in between the range 1 to %d:",n);
    for(num = 1;num<=n;num++)
    {
        count = 0;
        for(i=2;i<=num/2;i++)
        {
            if(num%i==0)
            {
                count++;
                break;
            }
        }
        if(count==0 && num!= 1)
            printf("%d ",num);
    }
}

return 0;
}
```

**SAMPLE INPUT:**

Enter the range: 20

**EXPECTED OUTPUT:**

The prime numbers in between the range 1 to 20: 1 3 5 7 11 13 17 19

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise:**

1. Write a program to generate the following pattern

i. *	ii. 1
* *	1 2
* * *	1 2 3

2. Write a program to

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**

**Week 5**

5 a) Write a program to find both the largest and smallest number in a list of integers

**AIM:** To find the largest and smallest number in a list of integers.

**ALGORITHM:**

Step 1: start

Step 2: read n

Step 3: initialize i=0

Step 4: if i<n do as follows. If not goto step 5

    Read a[i]

    Increment i

    goto step 4

Step 5: small=a[0], large=a[0]

Step 6: initialize i=0

Step 7: if i<n do as follows. If not goto step 8

    If a[i]<small

        Assign small=a[i]

    If a[i]>large

        Assign large=a[i]

    Increment i goto Step 7

Step 8: print small, large

Step 9: stop

**Program:**

```
#include<stdio.h>
int main()
{
    int a[10],i,n,small,large;
    printf("Enter The Array Size:");
    scanf("%d",&n);
    printf("Enter The Array elements:");
    for(i=0;i<n;i++)// read the elements of an array
        scanf("%d",&a[i]);
    small=a[0];
    large=a[0];
    for(i=1;i<n;i++)// read the elements of an array
    {
        if(a[i]<small)// check the condition for minimum value
            small=a[i];
        if(a[i]>large)//check the condition for maximum value
            large=a[i];
    }
    printf("largest value is:%d\n",large);
    printf("smallest value is:%d\n",small);
    return 0;
}
```

**SAMPLE INPUT:**

Enter The Array Size: 10

ENTER THE ELEMENTS OF ARRAY

7      10      9      8      6      5      2      3      4      1

**EXPECTED OUTPUT:**

largest value is : 10

smallest value is : 1

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

5 b) Write a C program to find the sum of integer array elements

**AIM:** To find the sum of integer array elements

**Description:** Consider an integer array. During each iteration, element of the array will be added to sum.

**ALGORITHM:**

Step 1: start

Step 2: Read n elements into array Step 3: initialize sum=0

Step 4: for(i=0;i<n;i++)

Step 5: sum=sum+A[i] step 6: print sum

step 7: stop

**PROGRAM:**

```
#include<stdio.h>
int main( )
{
    int A[50],sum=0,i,n;
    printf("Enter how many values to read");
    scanf("%d",&n);
    printf("enter elements into array");
    for(i=0;i<n;i++)
        scanf("%d",&A[i]);
    for(i=0;i<n;i++)
        sum=sum+A[i];
    printf("Addition of array elements is %d",sum);
    return 0;
}
```

**SAMPLE INPUT:**

Enter the array elements : 1 2 3 4 5 6 7 8 9 1

**EXPECTED OUTPUT:**

Addition of array elements is : 46

**Record at least 2 results**



**Assessment**

**Not Satisfactory**

**Needs Improvement**

**Partially Executed**

**Executed Successfully**

☐☐☐☐

**Signature of faculty with date**

### **Exercise**

- 1) Write a C program to generate the first n terms of the Fibonacci, use one dimensional array to store the series.
- 2) Write a program to search for a given element in an array using linear search.
- 3) Write a program to find Fibonacci prime numbers.
- 4) Write a C Program to Sort the Array in an Ascending Order.
- 5) Write a program to count a total number of duplicate elements in an array.

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**



## Week 6

6 a) Write a program to perform addition of two matrices.

**AIM:** To perform addition of two matrices.

**Description:** Consider two matrices and their order is  $R1 \times C1$  and  $R2 \times C2$ . The condition is  $R1 == R2$  and  $C1 == C2$ , then only the addition is possible.

**ALGORITHM:**

Step 1: Start

Step 2: Read the order of the two matrices  $R1, C1$  and  $R2, C2$

Step 3: if  $R1 \neq R2$  or  $C1 \neq C2$

    Print "Addition not possible"

    goto step 8

Step 4: for  $i$  is 0 to  $R1$  by step 1

    for  $j$  is 0 to  $C1$  by step 1

        read  $a[i][j]$

Step 5: for  $i$  is 0 to  $R2$  by step 1

    for  $j$  is 0 to  $C2$  by step 1

        read  $b[i][j]$

Step 6: for  $i$  is 0 to  $R1$  by step 1

    for  $j$  is 0 to  $C1$  by step 1

        calculate

$c[i][j] = a[i][j] + b[i][j]$

Step 7: for  $i$  is 0 to  $R1$  by step 1

    for  $j$  is 0 to  $C1$  by step 1

        print  $c[i][j]$

Step 8: Stop

**PROGRAM:**

```
#include<stdio.h>
int main()
{
int a[5][5],b[5][5],c[5][5];
int i,j,p,q,r,s;
    printf("ENTER ORDER OF A MATRIX\n");
    scanf("%d%d",&p,&q);
    printf("ENTER ORDER OF B MATRIX\n");
    scanf("%d%d",&r,&s);
    if(p==r&&q==s)
    {
        printf("ENTER A MATRIX\n");
        for(i=0;i<p;i++)
            for(j=0;j<q;j++)
                scanf("%d",&a[i][j]);
        printf("ENTER B MATRIX\n");
        for(i=0;i<p;i++)
            for(j=0;j<q;j++)
                scanf("%d",&b[i][j]);
        for(i=0;i<p;i++)
            for(j=0;j<q;j++)
                c[i][j]=a[i][j]+b[i][j];
        printf(" After Addition of two matrices :\n");
        for(i=0;i<p;i++)
        {
            for(j=0;j<q;j++)
            {
                printf("%d\t",c[i][j]);
            }
            printf("\n");
        }
    }
    else
    {
        printf("Addition not possible");
    }
return 0;
}
```

**SAMPLE INPUT:**

ENTER ORDER OF A MATRIX 2 2  
ENTER ORDER OF B MATRIX 2 2  
ENTER A MATRIX  
1 2  
3 4  
ENTER B MATRIX  
1 2  
3 4

**EXPECTED OUTPUT:**

After Addition of two matrices :  
2    4  
6    8

**Record at least 2 results****Assessment**

Not Satisfactory

Needs Improvement

Partially Executed

Executed Successfully

☐  
☐  
☐  
☐**Signature of faculty with date**



6 b) Write a C program to perform multiplication of two matrices.

**AIM:** To perform multiplication of two matrices.

**Description:** Consider two matrices and their order is  $R_1 \times C_1$  and  $R_2 \times C_2$ .

The condition is  $C_1 == R_2$ , then only the multiplication is possible.

**ALGORITHM:**

Step 1: Start

Step 2: Read order of two matrices  $R_1$ ,  $C_1$  and  $R_2$ ,  $C_2$

Step 3: if  $C_1 \neq R_2$

    print "Multiplication not possible"

    goto Step 8

Step 4: for i is 0 to  $R_1$  by step 1

    for j is 0 to  $C_1$  by step 1

        read  $a[i][j]$

Step 5: for i is 0 to  $R_2$  by step 1

    for j is 0 to  $C_2$  by step 1

        read  $b[i][j]$

Step 6: for i is 0 to  $R_1$  by step 1

    for j is 0 to  $C_2$  by step 1

$c[i][j] = 0$ ;

        for k is 0 to  $C_1$  by step 1

            calculate  $c[i][j] = c[i][j] + a[i][k] * b[k][j]$

Step 7: for i is 0 to  $R_1$  by step 1

    for j is 0 to  $C_2$  by step 1

        print  $c[i][j]$

Step 8: Stop

**Program:**

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

```
int main()
```

```
{ int a[5][5],b[5][5],c[5][5],m,n,p,q;
```

```
int i,j,k;
```

```
    printf("Enter the size of A Mtrix (Row and Col): \n");
```

```
    scanf("%d%d",&m,&n);
```

```
    printf("Enter the size of B Mtrix (Row and Col): \n");
```

```
    scanf("%d%d",&p,&q);
```

```
    if(n!=p)
```

```
    {        printf("Multiplication Not Possible\n Please re-enter\n");
```

```
            printf("correct size and try again ..... \n");
```

```
    }
```

```
    else
```

```
    {        printf("Enter Matrix A Values Row by Row\n");
```

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

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

```
scanf("%d",&a[i][j]);
printf("Enter Matrix B Values Row by Row\n");
for (i=0;i<p;i++)
    for(j=0;j<q;j++)
        scanf("%d",&b[i][j]);
//logic for multiplication
for (i=0;i<m;i++)
{
    for(j=0;j<q;j++)
    {
        c[i][j]=0;
        for(k=0;k<n;k++)
            c[i][j]+= a[i][k]*b[k][j];
    }
}
printf("A Matrix is :\n");
for (i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("%5d",a[i][j]);
    }
    printf("\n");
}
printf("B Matrix is :\n");
for (i=0;i<p;i++)
{
    for(j=0;j<q;j++)
    {
        printf("%5d",b[i][j]);
    }
    printf("\n");
}
printf("C Matrix is :\n");
for (i=0;i<m;i++)
{
    for(j=0;j<q;j++)
    {
        printf("%5d",c[i][j]);
    }
    printf("\n");
}
}
return 0;
```

```
}
```

**SAMPLE INPUT:**

Enter the size of A Mtrix (Row and Col): 2 2

Enter the size of B Mtrix (Row and Col): 2 2

Enter Matrix Value Row by Row

1 0

2 6

Enter Matrix Value Row by Row

3 4

4 2

**EXPECTED OUTPUT:**

A matrix is:

1 0

2 6

B Matrix is:

3 4

4 2

C matrix is:

2 4

24 20

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise:**

1. Write a program to find whether given matrix is symmetric or not.
2. Write a Program to perform transpose of a given Matrix.
3. Write a program to find sum of rows and columns of a Matrix.

**RECORD NOTES**



**Week 7**

7a) Write a function that returns the sum of two numbers.

**ALGORITHM:**

Step 1: Start  
Step 2: Read a,b  
Step 3: Call add(a,b) goto step 6  
Step 4: Store result in "f"  
Step 5: Print "f" goto step 7  
Step 6: Begin               //sub program  
          return a+b  
          End  
Step 7: Stop

**Program:**

```
int add(int a,int b)
{
    return a+b
}
int main()
{
    int a,b;
    printf("Enter the two numbers:");
    scanf("%d%d",&a,&b);
    printf("Sum of %d and %d is %d",a,b,add(a,b)); // return to the sub program
    return 0;
}
```

**SAMPLE INPUT:**

Enter the two numbers: 6 12

**EXPECTED OUTPUT:**

Sum of 6 and 12 is 18

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**



7b) Write a C program to find the factorial of given number

**AIM:** To find the factorial of a given number using non-recursive function

**Description:**  $n! = n * (n-1) * (n-2) \dots * 1$

**ALGORITHM:**

Step 1: Start  
Step 2: Read n  
Step 3: Call fact(n) goto step 6  
Step 4: Store result in "f"  
Step 5: Print "f" goto step 10  
Step 6: Begin //sub program  
    Initialize f  $\leftarrow$  1  
Step 7: for i is 1 to n by step 2  
Step 8: Calculate f = f\*i  
Step 9: return "f"  
    End  
Step 10: Stop

**PROGRAM:**

```
#include<stdio.h>
int fact(int);
int main()
{
    int n,i,f;
    printf("ENTER A VALUE FOR n:");
    scanf("%d",&n);
    f=fact(n);
    printf("THE FACTORIAL OF A GIVEN NO IS..%d",f);
    return 0;
}
int fact(int n)
{
    int i,f=1;
    for(i=1;i<=n;i++)
        f=f*i;
    return f;
}
```

**SAMPLE INPUT:**

ENTER A VALUE FOR n: 5

**EXPECTED OUTPUT:**

THE FACTORIAL OF A GIVEN NUMBER IS..120

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise:**

1. Write menu driven program perform arithmetic operations. Use functions to perform arithmetic operations
2. Write a function to check whether a given number is palindrome or not. Function should return either true or false

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**

## Week 8

8a) Write a function to swap the values of two variables using call by value

**Aim:** To swap the values of two variables using Call by Value

**Algorithm:****Main Program:**

Step 1: start  
Step 2: read a,b  
Step 3: call swap ( a , b )  
Step 4: Stop

**Sub Program:**

Step 5: t=a;  
Step 6: a =b;  
Step 7: b=t;  
Step 8: print a ,b values  
Step 9: return to main program

**Program:**

```
#include<stdio.h>
void swap(int , int);           // Declaration of function
int main( )
{
    int a,b;
    printf("Enter any two integers:");
    // call by value
    swap(a,b);                  // a and b are actual parameters
}
void swap( int x, int y )       // x and y are formal parameters
{
    int t;
    t= x ;
    x= y ;
    y = t ;
    printf ( "\nx = %d y = %d", x, y ) ;
}
```

**SAMPLE INPUT:**

Enter any two integers:10 20

**EXPECTED OUTPUT:**

x=20 y=10

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**



**8) b)** Write a C Program to swap the values of two variables using Call by Reference

### Algorithm:

#### Main Program:

Step 1: start  
Step 2: read a,b  
Step 3: call swap ( &a , &b )  
Step 4: print a ,b values  
Step 5: Stop

#### Sub Program:

Step 5: t=\*a;  
Step 6: \*a=\*b;  
Step 7: \*b=t;  
Step 8: return to main program

### Program:

```
#include <stdio.h>
void swap(int *a, int *b)
{ int temp;
  temp=*a;
  *a=*b;
  *b=temp;
}
int main()
{
  int num1,num2;
  printf("Enter any Two Integers:");
  scanf("%d%d",&num1,&num2);
  swap(&num1,&num2);
  printf("Number1 = %d\n",num1);
  printf("Number2 = %d",num2);
  return 0;
}
```

### SAMPLE INPUT:

Enter any two integers: 2 3

### EXPECTED OUTPUT:

Number1 = 3  
Number2 = 2

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise:**

1. Write a program which consists of user defined function to reverse a string
2. Design a user defined function to convert decimal number to binary.

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**

**Week 9**

9a) Write a recursive function to find factorial of a given integer

**AIM:** To find the factorial of a given integer using recursive function.

**Description:**  $n! = n * (n-1) * (n-2) * \dots * 1$

**ALGORITHM:**

- Step 1: start
- Step 2: read n
- Step 3: call sub program  $f = \text{fact}(n)$
- Step 4: print the f value
- Step 5: stop

**Sub program fact(n):**

- Step 1: if  $n=0$  return 1 to main program
- Step 2: return  $n * \text{fact}(n-1)$  to main program

**PROGRAM:**

```
#include<stdio.h>
int fact(int);
int main()
{
    int n,res;
    printf("Enter any number:");
    scanf("%d",&n);
    res=fact(n);
    printf("Factorial of given number is..%d",res);
    return 0;
}
int fact(int n)
{
    if(n==0)
        return(1);
    else
        return(n*fact(n-1));
}
```

**SAMPLE INPUT:**

Enter any number: 5

**EXPECTED OUTPUT:**

Factorial of given number is..120

**Record at least 2 results**

Assessment	
Not Satisfactory	<input type="checkbox"/>
Needs Improvement	<input type="checkbox"/>
Partially Executed	<input type="checkbox"/>
Executed Successfully	<input type="checkbox"/>

**Signature of faculty with date**



9b) Write a recursive function to find GCD of two integers

**Aim:** To find the GCD of two given integers by using the recursive function

**Description:** The greatest common divisor (gcd) of two or more integers, when at least one of them is not zero, is the largest positive integer that divides the numbers without a remainder.

For example, the GCD of 8 and 12 is 4.

**Algorithm:**

**Main program:**

- Step 1: start
- Step 2: read a,b
- Step 3: call the sub program GCD(a,b) for print the value
- Step 4: stop

**Sub program: GCD(n,m)**

- Step 1: if  $n > m$  return GCD(n,m)
- Step 2: if  $n == 0$  return m else goto step 3
- Step 3: return GCD (n,m%n)
- Step 4: return to main program

**Program:**

```
#include<stdio.h>
int gcdrecursive(int m,int n)
{
    if(n>m)
        return gcdrecursive(n,m);
    if(n==0)
        return m;
    else
        return gcdrecursive(n,m%n);
}
```

```
int main()
{
    int a,b;
    printf("Enter the two numbers whose gcd is to be found:");
    scanf("%d%d",&a,&b);
    printf("GCD of a,b is %d",gcdrecursive(a,b)); // return to the sub program
    return 0;
}
```

**SAMPLE INPUT:**

Enter the two numbers whose gcd is to be found: 5 25

**EXPECTED OUTPUT:**

GCD of a,b is : 5

**Record at least 2 results**

Assessment	
Not Satisfactory	<input type="checkbox"/>
Needs Improvement	<input type="checkbox"/>
Partially Executed	<input type="checkbox"/>
Executed Successfully	<input type="checkbox"/>

**Signature of faculty with date**

**Exercise:**

1. Write a program to multiply two numbers using recursion.
2. Write a program to print Fibonacci numbers using recursion.
3. Write a program to find sum of natural numbers using recursion.

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**

**Week 10**

10a) Write a program to check whether the given string is palindrome or not.

**Aim:** To determine if the given string is palindrome or not.

**Description :** Palindrome means string on reversal should be same as original

Ex: madam on reversal is also madam

**Algorithm:**

Step 1: start

Step 2: read string A

Step 3: copy string A into B

Step 4: reverse string B

Step 5: compare A & B

    If A equals B to got step 6

    else goto step 7

Step 6: print given string A is palindrome

Step 7: print given string is not palindrome

Step 8: stop

**Program:**

```
#include <stdio.h>
#include <string.h>
int main()
{
    char string[25], reverse_string[25] = {'\0'};
    int i, length = 0, flag = 0;
    printf("Enter a string \n");
    gets(string);
    for (i = 0; string[i] != '\0'; i++)
        length++;
    printf("The length of the string '%s' = %d\n", string, length);
    for (j=0, i = length - 1; i >= 0 ; i--, j++)
        reverse_string[j] = string[i];
    for (flag = 1, i = 0; i < length ; i++)
    {
        if (reverse_string[i] != string[i])
        {
            flag = 0;
            break;
        }
    }
    if (flag == 1)
        printf ("%s is a palindrome \n", string);
    else
        printf ("%s is not a palindrome \n", string);
    return 0;
}
```

**SAMPLE INPUT:**

Enter a string  
madam

**EXPECTED OUTPUT:**

The length of the string 'madam' = 5  
madam is a palindrome



**Record at least 2 results**

**Assessment**

Not Satisfactory

Needs Improvement

Partially Executed

Executed Successfully

☐☐☐☐

**Signature of faculty with date**

10 b) Write a program to sort the given names in alphabetical order

```
#include<stdio.h>
#include<string.h>
int main(){
    int i,j,n;
    char str[100][100],s[100];

    printf("Enter number of names :");
    scanf("%d",&n);

    printf("Enter names in any order:");

    for(i=0;i<n;i++){
        scanf("%s",str[i]);

        for(i=0;i<n;i++){
            for(j=i+1;j<n;j++){
                if(strcmp(str[i],str[j])>0){
                    strcpy(s,str[i]);
                    strcpy(str[i],str[j]);
                    strcpy(str[j],s);
                }
            }
        }

        printf("The sorted order of names are:");

        for(i=0;i<n;i++){
            printf("%s",str[i]);
        }
        return 0;
    }
```

**SAMPLE INPUT:**

Enter number of names:

5

Enter names in any order:

Pinky

Lucky

Ram

Appu

Bob

**EXPECTED OUTPUT:**

The sorted order of names is:

Appu

Bob

Lucky

Pinky

Ram

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise:**

1. Write a program to use function to insert a sub-string in to given main string from a given position.
2. Write a program to compare two strings without using library functions.
3. Write a program to concatenate two strings without using library functions.
4. Write a program to convert lowercase string into upper case without using library functions.
5. Write a program to convert upper case string into lower case without using library functions.

## Week 11

11 a) Write a program to find the length of the string using Pointer.

**AIM:** To find the length of the string using Pointer.

**Algorithm:**

Step 1: start

Step 2: read string

Step 3: count=0; i=0

Step 4: if string[i]!='\0'  
    count:= count +1  
    i:=i+1

    goto step 4

step 5: print count

step 6 stop

**Program:**

```
#include<stdio.h>
int string_Len(char*);
int main()
{
    char str[20];
    int length;
    printf("\nEnter any string : ");
    gets(str);
    length = string_Len (str);
    printf("The length of the given string %s is : %d", str, length);
    return 0;
}
int string_Len (char *p)    /* p=&str[0] */
{
    int count = 0;
    while (*p != '\0')
    {
        count++;
        p++;
    }
    return count;
}
```

**SAMPLE INPUT:**

Enter the String : pritesh

**EXPECTED OUTPUT:**

Length of the given string pritesh is : 7

**Record at least 2 Results****Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**AIM:** To find the sum of integer array elements

**Description:** Consider an integer array. During each iteration, element of the array will be added to sum.

**ALGORITHM:**

Step 1: start

Step 2: Read n elements into array

Step 3: initialize sum=0

Step 4: for(i=0;i<n;i++)

Step 5: sum=sum+\*(A+i)

step 6: print sum

Step 7: stop

**PROGRAM:**

```
#include<stdio.h>
int main( )
{
    int A[50],sum=0,i,n;
    printf("Enter how many values to read");
    scanf("%d",&n);
    printf("enter elements into array");
    for(i=0;i<n;i++)
        scanf("%d",&A[i]);
    for(i=0;i<n;i++)
        sum=sum+*(A+i);
    printf("Addition of array elements is %d",sum);
    return 0;
}
```

**SAMPLE INPUT:**

Enter the array elements : 1 2 3 4 5 6 7 8 9 1

**EXPECTED OUTPUT:**

Addition of array elements is : 46

**Record at least 2 results**

**Assessment**  
**Not Satisfactory**  
**Needs Improvement**  
**Partially Executed**  
**Executed Successfully**

☐  
☐  
☐  
☐

**Signature of faculty with date**

### **Exercise**

- 1 Write a C program to generate the first n terms of the Fibonacci, use one dimensional array to store the series.
- 2) Write a program to search for a given element in an array using linear search.
- 3) Write a program to count a total number of duplicate elements in an array.



**RECORD NOTES**

**RECORD NOTES**

**RECORD NOTES**



**Week 12**

**12) a) Write a program to create book structure and display the contents of a book.**

**AIM:** Write a program to create book structure and display the contents of a book.

**Program:**

```
#include<stdio.h>
struct book
{
    Char bname[50];
    int ssn;
    int pages;
    int rate;
};

int main()
{
    struct book b1;
    printf("Enter Book SSN Number:");
    scanf("%d",&b1.ssn);
    printf("Enter Number of pages:");
    scanf("%d",&b1.pages);
    printf("Enter price:");
    scanf("%d",&b1.rate);
    fflush(stdin);
    printf("Enter Book Name:");
    gets(b1.bname);
    printf("\nName of the Book : %s\n",b1.bname);
    printf("\nSSN of the Book : %d\n",b1.ssn);
    printf("\nPages in the Book : %d\n",b1.pages);
    printf("\nPrice of the Book : %d\n",b1.rate);

    return(0);
}
```

**SAMPLE INPUT:**

Enter Book SSN  
Number:123Enter  
Number of pages:200  
Enter price:100  
Enter Book Name:c programming

**EXPECTED OUTPUT:**

Name of the Book : c  
programmingSSN of the Book :  
123  
Pages in the Book :  
200Price of the  
Book : 100

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**12) b) Write a C Program to Calculate Total and Percentage marks of a student using structure.****Program:**

```
#include<stdio.h>
struct student
{
    int rl;
    char nm[20];
    int m1;
    int m2;
    int m3;
    int t;
    float per;
};
int main()
{
    struct student a;
    printf(" Enter RollNo, Name amd three sub marks\n");
    scanf("%d%s%d%d%d",&a.rl,&a.nm,&a.m1,&a.m2,&a.m3);
    a.t=a.m1+a.m2+a.m3;
    a.per=a.t/3.0;
    printf("rollno=%d\n",a.rl);
    printf("Name=%sk\n",a.nm);
    printf("m1=%d\n",a.m1);
    printf("m2=%d\n",a.m2);
    printf("m3=%d\n",a.m3);
    printf("total=%d\n",a.t);
    printf("per=%f\n",a.per);
    return 0;
}
```

**SAMPLE INPUT:**

Enter RollNo, Name and three sub marks  
12 rama 30 40 50

**EXPECTED OUTPUT:**

rollno=12  
Name=rama  
m1=30  
m2=40  
m3=50  
total=120  
per=40.000000



**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise**

1. Write a program to calculate salary of an employee using structure
2. Write a program to read N Items rate and quantity and Calculate total billed amount.









## Week 13

13) a) Write a C program that uses functions to perform the following operations:

- i) Reading a complex number      ii) Writing a complex number
- iii) Addition of two complex numbers      iv) Multiplication of two complex

**AIM:** To perform arithmetic operations on complex numbers

Complex numbers of type  $a+ib$

Addition:  $(a+ib)+(x+iy)=a+x+i(b+y)$

Subtraction:  $(a+ib)-(x+iy)=a-x+i(b-y)$

Multiplication:  $(a+ib)*(x+iy)=ax-by+i(ay+bx)$

Division

$$(a+ib)/(x-iy) = \frac{a+ib}{x+iy} * \frac{x-iy}{x-iy} = \frac{(a+ib)*(x-iy)}{x^2+y^2} = \frac{(ax+by)+i(bx-ay)}{x^2+y^2} = \frac{ax+by}{x^2+y^2} + i \frac{bx-ay}{x^2+y^2}$$

**ALGORITHM:**

Step 1: start

Step 2: Read Two complex numbers c1, c2

Step 3:  $c3=c1+c2$

Step 4: print c3

Step 5:  $c3=c1-c2$

Step 6: print c3

Step 7:  $c3=c1*c2$

Step 8: print c3

Step 9:  $c3=c1/c2$

Step 10: print c3

Step 11: print c

Step 12: stop

**PROGRAM:**

```
#include<stdio.h>
#include<stdlib.h>
struct complex
{
    float real,img;
};

/*code for reading complex number*/
struct complex read_complex()
{struct complex c;
    printf("enter real part of complex number");
    scanf("%f",&c.real);
    printf("enter Imaginary part of complex number");
    scanf("%f",&c.img);
    return c;
}

/*code for adding complex numbers*/
struct complex add_complex(struct complex c1,struct complex c2)
{
    struct complex c3;
    c3.real=c1.real+c2.real;
    c3.img=c1.img+c2.img;
    return c3;
}

/*code for subtraction of complex numbers*/
struct complex sub_complex(struct complex c1,struct complex c2)
{
    struct complex c3;
    c3.real=c1.real-c2.real;
    c3.img=c1.img-c2.img;
    return c3;
}

/*code for multiplication of complex numbers*/
struct complex mul_complex(struct complex c1,struct complex c2)
{
    struct complex c3;
    c3.real=c1.real*c2.real-c1.img*c2.img;
    c3.img= c1.img*c2.real+c2.img*c1.real;
    return c3;
}
```

```
/*code for division of complex numbers*/
struct complex div_complex(struct complex c1,struct complex c2)
{
    struct complex c3;
    c3.real= (c1.real*c2.real+c1.img*c2.img)/(c2.real*c2.real+c2.img*c2.img);
    c3.img= (c1.img*c2.real-c1.real*c2.img)/(c2.real*c2.real+c2.img*c2.img);
    return c3;
}

/*code for display of complex number*/
void display_complex(struct complex c)
{
    char sign;
    printf("The result is:");
    if(c.img<0)
    {
        sign='-';
        c.img=-c.img;
    }
    else
        sign='+';
    printf("%5f%ci%5f",c.real,sign,c.img);
}

int main()
{
    int choice;
    struct complex a,b,c;
    while(1)
    {
        printf("\n.....\n");
        printf("| Menu for operation complex numbers |\n ");
        printf(".....\n");
        printf("1.Addition \n ");
        printf("2.Subtraction \n ");
        printf("3.Multiplication \n ");
        printf("4.Division \n ");
        printf("5.Clear Screen \n ");
        printf("6.Exit Menu \n ");
        printf("Enter Your Choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("You Have Selected Addition operation on complex Numbers\n");
                    printf("Enter First complex number\n");
                    a=read_complex();
                    printf("Enter Second complex Number\n");
                    b=read_complex();
                    c=add_complex(a,b);
```



```
        display_complex(c);
        break;
    case 2:printf("You Have Selected Subtraction operation on complex Numbers\n");
        printf("Enter First complex number\n");
        a=read_complex();
        printf("Enter Second complex Number\n");
        b=read_complex();
        c=sub_complex(a,b);
        display_complex(c);
        break;
    case 3:printf("You Have Selected Multiplication operation on complex Numbers\n");
        printf("Enter First complex number\n");
        a=read_complex();
        printf("Enter Second complex Number\n");
        b=read_complex();
        c=mul_complex(a,b);
        display_complex(c);
        break;

    case 4:printf("You Have Selected Division operation on complex Numbers\n");
        printf("Enter First complex number\n");
        a=read_complex();
        printf("Enter Second complex Number\n");
        b=read_complex();
        c=div_complex(a,b);
        display_complex(c);
        break;

    case 5: system("cls");
        break;

    case 6: exit(0);
        default: printf("Invalid choice");
        }
    }
}
```

**SAMPLE INPUT:**

-----  
Menu For Operation Complex Numbers

1. Addition
- 2.Subtraction
- 3.Multiplication
- 4.Division
5. Clear Screen
6. Exit Menu

Enter Your Choice:

Enter Your Choice: 1

You Have Selected Addition Operation On Complex Numbers

Enter First Complex Number  
Enter Real Part Of Complex Number1  
Enter Imaginary Part Of Complex Number2  
Enter Second Complex Number  
Enter Real Part Of Complex Number1  
Enter Imaginary Part Of Complex Number2

**EXPECTED OUTPUT:**

THE RESULT IS:2.000000+I4.000000

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**13b ) Write a C program to reverse the first n characters in a file.**

**Aim:** To reverse the first n characters in a file

**Algorithm:**

- Step 1: Start
- Step 2: read the command line arguments
- Step 3: check if arguments=3 or not
  - If not print invalid no of arguments
- Step 4: open source file in read mode
- Step 5: if NULL pointer, then print file cannot be open
- Step 6: Store no of chars to reverse in k
  - K= \*argv[2]-48
- Step 7: read the item from file stream using fread
- Step 8: Store chars from last position to initial position in another string(temp)
- Step 9: print the temp string
- Step 10: Stop

**Program:**

```
#include <stdio.h>
#include <string.h>
#include <process.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    char a[15];
    char s[20];
    char n;
    int k;
    int j=0;
    int i;
    int len;
    FILE *fp;
    if(argc!=3)
    {
        puts("Improper number of arguments.");
        exit(0);
    }
    fp = fopen(argv[1], "r");
    if(fp == NULL)
    {
        puts("File cannot be opened.");
        exit(0);
    }
    k=atoi(argv[2]);
    n = fread(a,1,k,fp);
    a[n]='\0';
    len=strlen(a);
    for(i=len-1;i>=0;i--)
    {
        s[j]=a[i];
        printf("%c",s[j]);
        j=j+1;
    }
    s[j+1]='\0';
    return 0;
}
```

**SAMPLE INPUT:**

Input text file:

**source.txt:**

this is source

**EXPECTED OUTPUT:**

Command line arguments

C:\TURBOC~1\Disk\TurboC3\BIN>week11b source.txt 14

ecruos si siht

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise**

1. Write a Program to Store Information of N Students  
like Name, Roll Number, marks and process result Using Structure
2. Write program to calculate difference between two time periods using structures.











**Week 14**

**14)a) Write a C program to copy the contents of one file to another.**

**Aim:**Program which copies one file to another

**Algorithm:**

- Step 1: Start
- Step 2: read file1,file2
- Step 3: open source file in read mode
- Step 4: if NULL pointer, then print source file can not be open
- Step 5: open destination file in write mode
- Step 6: if NULL pointer, then print destination file can not be open
- Step 7: read a character from source file and write to destination file until EOF
- Step 8 : Close source file and destination file
- Step 9: Stop

**Program:**

```
#include<stdio.h>
#include<process.h>
int main()
{
    FILE *ft,*fs;
    int c=0;

    fs=fopen("a.txt","r");
    ft=fopen("b.txt","w");
    if(fs==NULL)
    {
        printf("Source file opening error\n");
        exit(1);
    }
    else
    if(ft==NULL)
    {
        printf("Target file opening error\n");
        exit(1);
    }
    while(!feof(fs))
    {
        fputc(fgetc(fs),ft);
        c++;
    }
    printf("%d bytes copied from 'a.txt' to 'b.txt'",c);
    c=fcloseall();
    printf("%d files closed",c);
    return 0;
}
```

**SAMPLE INPUT:**

a.txt

An array is a collection of elements of similar datatypes

**EXPECTED OUTPUT:**

57 bytes copied from 'a.txt' to 'b.txt'

2 files closed

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**14 b) Write a program to merge two files into a third file.****Algorithm:**

Step 1: Start  
Step 2: read file1,file2,file3  
Step 3: open file1 in read mode  
Step 4: if NULL pointer, then print source file cannot be open  
Step 5: open file3 in write mode  
Step 6: if NULL pointer, then print file3 cannot be open  
Step 7: read a character from file1 and write to file3 until EOF  
Step 8 : Close file1  
Step 9: open file2 in read mode  
Step 10: if NULL pointer, then print source file can not be open  
Step 11: read a character from file2 and write to file3 until EOF  
Step 12: Close file2 and file3  
Step 13: Stop

**Program :**

```
#include<stdio.h>
int main()
{
    FILE *fp1,*fp2,*fp3;
    char file1[20],file2[20],file3[20],ch;
    puts("Program to merge two files. ..\n");
    puts("Enter first file name:");
    gets(file1);
    puts("Enter Second file name:");
    gets(file2);
    puts("Enter Destination file name:");
    gets(file3);
    fp1=fopen(file1,"r");
    fp2=fopen(file2,"r");
    fp3=fopen(file3,"w");
    if(fp1==NULL&&fp2==NULL)
        printf("Error opening file1 and file2..... \n");
    else
    {
        if(fp3==NULL)
            printf("Error in creating destination file..... \n");
        else
        {
            while((ch=fgetc(fp1))!=EOF)
```

```
        putc(ch,fp3);
        while((ch=fgetc(fp2))!=EOF)
            putc(ch,fp3);
    }
    printf("File Merging Sucessfull ...");
    fcloseall();
}
return 0;
}
```

**Record at least 2 results**

**Assessment**

Not Satisfactory

☐

Needs Improvement

☐

Partially Executed

☐

Executed Successfully

☐

**Signature of faculty with date**

**Exercise**

1. Write program to read name and marks of n number of students and store them in a file.
2. Write a program to read name and marks of n number of students from keyboard and store them in a file. If the file previously exists, add the information to the file.
3. Write a program to write all the members of an array of structures to a file using fwrite(). Read the array from the file and display on the screen.

**RECORD NOTES**

**RECORD NOTES**



**RECORD NOTES**

**RECORD NOTES**