

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: Business Systems & Applications
Year: 1st Year

Subject Code-BCA101
Semester: First

| Module Number | Topics | Number of Lectures |
|-----------------------------------|---|---------------------------|
| 1 | Introduction: | 5L |
| | Use of computers for managerial applications, Technology issues and data processing in organisations, Introduction to Information Systems, shift in Information system thinking, latest trends in Information Technology. | 5 |
| 2 | Information System: | 4L |
| | Computer Based Information Systems- office automation systems. Decision making and MIS, transaction processing systems. | 4 |
| 3 | Decision Support System: | 4L |
| | Decision support system, Group Decision Support, Executive Information systems, DSS generator. | 4 |
| 4 | Modern computation: | 4L |
| | Introduction to: Artificial Intelligence Based Systems, End user computing, Distributed data processing. | 4 |
| 5 | IS architecture: | 4L |
| | Deciding on IS architecture, IT leadership & IS strategic planning. | 4 |
| 6 | IS strategy: | 3L |
| | Introduction to: IS strategy and effects of IT on competition. | 3 |
| 7 | ERP: | 5L |
| | Introduction to: ERP, re-engineering work processes for IT applications, Business Process Redesign. | 5 |
| 8 | Knowledge Based: | 3L |
| | Knowledge engineering and data warehouse. | 3 |
| Total Number Of Hours = 32 | | |

Assignment:

Module-1:

1. Describe the use of computer in managerial applications.

Module-2:

1. Explain Management Information System with examples.

Module-3:

1. Discuss Decision support system.

Module-4:

1. What do you mean by artificial intelligence (AI)? How AI involves in business?

Module-5:

1. Describe IS architecture with proper diagram.

Module-6:

1. Discuss about various IS strategies.

Module-7:

1. Define Enterprise resource planning. What do you mean by re-engineering?

Module-8:

1. Define Data warehouse? How we can Mine data?

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: Digital Electronics
Year: 1st Year

Subject Code-BCA102
Semester: First

| Module Number | Topics | Number of Lectures |
|----------------------|--|---------------------------|
| 1 | Number Systems and Codes | 5L |
| | 1. Binary, octal and hexadecimal number systems, and conversion of number with one radix to another | 2L |
| | 2. Different binary codes | 3L |
| 2 | Logic Functions | 7L |
| | 3. Boolean algebra and Boolean operators | 2L |
| | 4. Logic Functions | 1L |
| | 5. Minimization of logic functions using Karnaugh -map | 3L |
| | 6. Quine-McClausky method of minimization of logic functions | 1L |
| 3 | Combinational Circuits | 8L |
| | 7. Introduction to combinational circuits, logic convention, and realization of simple combinational functions using gates | 2L |
| | 8. Implications of delay and hazard | 1L |
| | 9. Realization of adders and subtractors | 2L |
| | 10. Design of code converters, comparators, and decoders | 2L |
| | 11. Design of multiplexers, demultiplexers, | 1L |
| 4 | Analysis of Sequential Circuits | 8L |
| | 12. Introduction to sequential circuits: Moore and Mealy machines | 1L |
| | 13. Introduction to flip-flops like SR, JK, D & T with truth tables, logic diagrams, and timing relationships | 2L |
| | 14. Conversion of Flip-Flops, Excitation table | 2L |
| | 15. State tables, and realization of state tables | 3L |
| 5 | A/D and D/A Converter | 3L |
| | Different types of converters | 3L |
| 6 | Logic Families | 5L |
| | 16. Introduction to Logic families | 1L |
| | 17. TTL family | 1L |
| | 18. CMOS family | 2L |
| | 19. Electrical characteristics of logic families | 1L |

Faculty In-Charge

HOD, ECE Dept.

Assignment

Module 1

Perform the following number system conversions:

- (a) $(1110010)_2 = (?)_{10}$
- (b) $(10000111)_2 = (?)_8$
- (c) $(DACB6)_{16} = (?)_8$

Module 2

1. A system of logic is to be designed which has two outputs & three inputs. One output will be TRUE if an odd number of inputs are TRUE. The other output will be TRUE if only one input alone is TRUE. Draw the truth table & write the corresponding Boolean equation.
2. (a) Minimize the following using K-map and realize the simplified expression using basic gates only.

$$Y = \sum (0, 2, 3, 6, 7, 8, 10, 11, 12, 15)$$

(b) What is the difference between combinational circuit and sequential circuit?

3. Prove that

$$(a) \quad BCD + A \bar{C} \bar{D} + ABD = ABD + A \bar{C} \bar{D} + AB \bar{C}$$

$$(b) \quad (A + B)(\bar{A} \bar{C} + C)(\overline{\bar{B} + AC}) = \bar{A} B$$

4. Plot the logical expression $ABCD + A \bar{B} \bar{C} \bar{D} + A \bar{B} C + AB$ on a 4-variable K-map; obtain the simplified expression from the map.
5. Minimize the function using K-map and realize the simplified expression using basic gates. $f(a, b, c, d) = \sum m(9, 10, 12) + d(3, 5, 6, 7, 11, 13, 14, 15)$
6. Minimize the function using K-map. $\Pi M(3, 6, 8, 11, 13, 14) \bullet d(1, 5, 7, 10)$.
7. Let $f(A, B) = \bar{A} + B$, then find the value of $f(f(x+y, y), z)$.
8. Let $X * Y = \bar{x} + y$ and $z = X * Y$, then find the value of $z * x$.
9. Write the canonical SOP form of $f = A + \bar{B}C$ and implement using decoder

Module 3

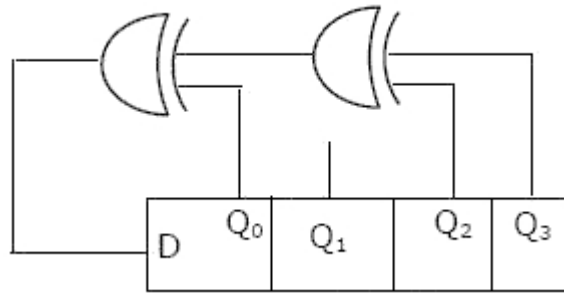
1. Implement a Full Adder circuit using two Half Adder. Write the truth table for Half Subtractor.
2. Implement the following functions using 3:8 Decoders. $F(A, B, C) = \sum m(0, 1, 4, 5, 7)$
3. What is Multiplexer?
4. Why multiplexer is called as “data selector”?
5. Implement $f(A, B, C) = \sum m(1, 2, 3, 5, 6, 7)$ using 4:1 MUX with AB as select line.
6. Write the truth table of 4:1 MUX and implement the circuit using logic gates.
7. Design XOR and OR gate using 2:1 MUX.
8. Write the canonical SOP form of $f = A + \bar{B}C$ and implement using decoder.

Module 4

1. Explain Ring counter. Draw the circuit diagram and wave form of Ring counter.
2. Perform the conversion from T flip-flop to JK flip-flop.

Lecture-wise Plan

3. Write down the difference between Synchronous and Asynchronous counter. What is modulus in a counter (explain with an example)? Design 3 bit asynchronous ripple counter (up counter) and draw the output waveform.
4. What is register?
5. What is shift register?
6. What are the types of shift register? Sketch the block diagrams of each type.
7. A 4 bit shift register, which shifts 1 bit to the right at every clock pulse, is initialized to values 1000 for ($Q_0Q_1Q_2Q_3$). The D input is derived from Q_0 , Q_2 and Q_3 through two XOR gates as shown in figure. Write the 4 bit values ($Q_0Q_1Q_2Q_3$) after each clock pulse till the pattern (1000) reappears on ($Q_0Q_1Q_2Q_3$).



8. Using proper truth table and logic diagram find the characteristic equation of SR Flip-Flop. Design a 2-bit up/down synchronous counter using SR flip flop. Use one directional control input M. For $M=0$, the counter will count up and for $M=1$, counter will count down.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: PC Software
Year: 1st Year

Subject Code- BCA103
Semester: First

| Module Number | Topics | Number of Lectures |
|---------------|---|--------------------|
| 1 | Basics of Computer Hardware | 12L |
| | 1. Microsoft Windows. Functioning of a Computer Components of a computer and their role. | 3L |
| | 2. Windows Feature. Memory System Type of memories and their characteristics | 4L |
| | 3. Desktop, creation of folder and shortcut, feature of windows explorer. I/O devices and their functions I/O devices | 5L |
| 2 | Basics of Computer Software | 17L |
| | 1. Toolbars, work area, save, save as, shortcut key | 3L |
| | 2. Menus, standard toolbar option | 2L |
| | 3. Mail Merge, clipart, style, design | 2L |
| | 4. Function, Formula, Formatting, protection, Data validation, Macros | 2L |
| | 5. Animation, design, slide show, other | 2L |
| | 6. Create database | 2L |
| | 7. Query and use of database | 2L |
| | 8. Task Manager, System Tools, etc | 2L |
| 3 | Internet Technologies | 19L |
| | 1. Networking and Internet Basic of Networking Concepts | 3L |
| | 2. Advantages of Networking | 3L |
| | 3. Basic model of Networks, Network Devices | 3L |
| | 4. TCP/IP, Web addresses, DNS, IP addresses | 3L |
| | 5. Web Applications I Browsing, E-mail, Messenger/Chat | 3L |
| | 6. Web Applications II Blogging, E-Learning and wiki, Collaboration, Social Networking | 4L |
| Total= | | 48 |

Faculty In-Charge

HOD, CSE Dept.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lecture-wise Plan

Assignment:

Module-1(Basics of Computer Hardware):

1. Explain the different generations of computer? List some of the applications areas of computer.
2. Explain the purpose of a motherboard in a computer?
3. What is need of memory in a computer? Also explain the need of a memory hierarchy.
4. Explain with examples, the different types of main memory in computer systems.
5. What are different input/output (I/O) devices in a computer? Explain any two output devices.

Module-2 (Basics of Computer Software):

1. Why do you need Computer Software? What are the software trends?
2. Make a Presentation for a Travel agency which would include the following topics:
 - Introduction on the Agency and its management
 - International Destinations
 - European Tours
 - Australian Tours
 - Asian Tours
 - Airline Information along with Fares
3. Create a Word document and type the text as given below. The headings, footer, formatting and layout should be as given in the following sample.

Organizational Research: Determining Appropriate Sample Size

The determination of sample size is a common task for many organizational researchers. Inappropriate, inadequate, or excessive sample sizes continue to influence the and accuracy of research. This manuscript describes the procedures for determining sample size for continuous and categorical variables using frequency analysis. Procedures for determining appropriate sample size for regression analysis, and common issues in sample size determination are examined.

A common goal of research is to collect data representative of a population. The researcher uses information gathered from the sample to generalize findings from a drawn sample back to a population, within the limits of random error. However, when critiquing business education Wunsch (1986) stated that "Two of the most consistent flaws included (1) disregard for sampling error when determining sample size, and (2) disregard for response and nonresponse bias" (31).

Within a quantitative design, determining sample size and dealing with no-response bias is essential. The question then is, how large of a

sample is required to infer research findings back to a population?

Basic Sample Size Determination Continuous Data

Before proceeding with sample size calculations, assuming continuous data, the researcher should determine if a categorical variable will play a role in data analysis. Assume that a researcher has set the alpha level a priori at .05, plans to use a seven point scale, has set the level of acceptable error at 3%, and has estimated the standard deviation of the scale as 1.161'. Cohen's sample size formula for continuous data and an example of its use is presented here along with the explanations as to these decisions were made.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: PC Software
Year: 1st Year

Subject Code- BCA103
Semester: First

Module-3(Internet Technologies):

1. You are asked to design a network for a Distance Learning University.

This network is to be used for the purpose of maintenance of records of the students and online delivery of learning contents. Assume that the network designers have proposed to have LANs at each study centers which are linked to the Regional Centers through a WAN. Answer the following questions in this context:

- i) What are the different network media that may be used for this network?

Give reason for selection of various media.

- ii) What topology will you suggest for the LANs in the network?

Justify your answer.

- iii) What are the different network devices that will be used for this network?

Explain the function of each of the device that you suggest to use.

2. Find the IP address of the device through which you are connected to

Internet service provider. Also Find the IP address and Subnet mask of your Computer. Is the IP address of your computer is address of a private network? Explain the term DNS and its purpose.

3. List the functions of an email and chat clients on the Internet.
4. What are the different security threats when you are connected to Internet?
5. What are the different features required in an e-learning platform?

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Title of Course: Introduction to Programming

Course Code: BCA104

Year: 1st Year

Semester: 1st

| Module Number | Topics | Number of Lectures |
|----------------------|---|---------------------------|
| 1 | Introduction to Computers | 2 |
| | 1. Generations, Classifications, Applications, Basic Organization, Input and output devices | 1 |
| | 2. Basic concept of Computer memory, Computer software and networks | 1 |
| 2 | Number system | 4 |
| | 1. Decimal, Binary, Octal, Hexa-decimal, Conversion of numbers, Addition and subtraction of two numbers, Two's compliment | 2 |
| | 2. Multiplication and division of binary numbers, Working with fractions, signed number representation in binary form | 1 |
| | 3. Logic gates | 1 |
| 3 | Introduction to C | 4 |
| | 1. compiling and executing C programs, using comments, keywords, identifiers, Data type, variables, constants | 1 |
| | 2. input/output statements in C, operators in C | 2 |
| | 3. type conversion and type casting. | 1 |
| 4 | Decision Control and looping statements | 6 |
| | 1. conditional branching statement | 2 |
| | 2. iterative statements | 2 |
| | 3. nested loops, break and continue statements, goto statement | 2 |
| 5 | Arrays & Strings | 6 |
| | 1. Declaration, accessing elements of array, storing values | 1 |
| | 2. calculating the length of array, two dimensional arrays | 1 |
| | 3. reading and writing strings, suppressing input, string taxonomy | 1 |
| | 4. string operations – using and without using library function | 2 |
| | 5. array of strings | 1 |
| 6 | Functions | 5 |
| | 1. Declaration, prototype, definition, function call | 1 |
| | 2. return statement, passing parameters to the function | 1 |
| | 3. scope of variable, storage classes | 1 |
| | 4. recursive functions | 2 |
| 7 | Pointers | 7 |
| | 1. introduction, declaration, Pointer expression and arithmetic | 1 |
| | 2. null pointer, generic pointer, passing arguments to functions using pointer | 1 |
| | 3. pointers and arrays, passing an array to function, difference between array name and pointer | 2 |
| | 4. pointers and strings, array of pointers | 1 |
| | 5. function pointers, pointers to pointers | 1 |
| | 6. dynamic memory allocation, drawbacks | 1 |

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: **Basic computation and principles of computer programming**

Subject Code: **CS101**

Year: **1st Year**

Semester: **1st**

| | | |
|-------------------------------|--|----------|
| 8 | Structure-union, Files, Preprocessor directives | 4 |
| | 1. Structure, nested structure, array of structure | 1 |
| | 2. union, array of union variable, unions inside structure | 1 |
| | 3. Files – Reading –writing etc | 1 |
| | 4. Preprocessor directives | 1 |
| Total Lecture Hours – 37 l.h. | | |

Faculty In-Charge

HOD, CSE Dept.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Title of Course: Introduction to Programming

Course Code: BCA104

Year: 1st Year

Semester: 1st

Assignments :

Unit 1:-

1. How is OCR technology better than ordinary image scanner?
2. How does MICR technology help to detect fraud in check payment?
3. Which factors will you consider while purchasing a monitor for your personal computer?
4. What is a head crash? How does it occur?
5. What is USB flash drive?
6. What is a BIOS? Which kind of memory is preferred in it, and why?
7. How is application software different from system software?
8. Classify the operating systems based on their capabilities.
9. In what situation must the network have a gateway?
10. Which device will you prefer to form a network – hub or a switch? Justify your answer.

Unit 2 :-

1. How can two numbers be subtracted only through addition? Explain with example.
2. Convert the following –
 - a) $(10110101)_2 - (?)_{10}$
 - b) $(5674)_8 - (?)_{10}$
 - c) $(A1E2)_{16} - (?)_{10}$
 - d) $(289)_{10} - (?)_8$
 - e) $(593)_{10} - (?)_2$
 - f) $(36541)_8 - (?)_{16}$
3. $10101011 - 01111 = ?$
4. $11100001 + 11111110111 = ?$
5. $1111000 / 100 = ?$
6. Why NAND gate is known as universal gate?

Unit 3 :-

1. State true or false –
 - a> \$amount is a valid identifier in C,
 - b> The equality operators have higher precedence than the relational operators.
 - c> Signed variables can increase the maximum positive range.
 - d> `printf("%d", scanf("%d",&num));` is valid statement
 - e> The modulus operators can be used only with integers
2. Write a program to prepare a grocery bill. For that enter the name of the items purchased, quantity in which it is purchased, and its price per unit. Then display the bill in the following format

*****BILL*****

| Item | Quantity | Price | Amount |
|-------------------------|----------|-------|--------|
| <hr/> | | | |
| Total amount to be paid | | | |
| <hr/> | | | |

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: **Basic computation and principles of computer programming**

Subject Code: **CS101**

Year: **1st Year**

Semester: **1st**

3. Write a program to read two floating point numbers. Add these numbers and assign the result to an integer. Finally display the value of all the three variables.
4. Find the error(s) –
Int n;
float a b;
double = a, b;
complex a b;
a,b : INTEGER
long int a;b;
5. Find the error(s) –
int a = 9;
float y =2.0;
a = b% a;
printf(“%d”, a);
6. Write the output of the code →
int main()
{
int a = 2,b =3, c = 4;
a=b=c;
printf(“a=%d”,a);
return 0;
}
7. Evaluate the expression – (x > y) + ++a || !c
8. Write a program to count the number of vowel in a given text.
9. Why do we include <stdio.h> in our program?
10. Write a program to calculate simple and compound interest .

Unit 4 :-

1. Change the following for loop into do-while loop
int i;
for(i=10 ; i>0 ; i--)
printf(“%d”, i);
2. WAP to accept any number and print the number of digits in that program.
3. WAP to print the sum of all odd numbers from 1 to 100.
4. WAP that displays all the numbers from 1 to 100 that are not divisible by 2 as well as by 3.
5. Write down the output
#include<stdio.h>
main()
{ int num = 10;
for(;;--num)

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Title of Course: Introduction to Programming

Course Code: BCA104

Year: 1st Year

Semester: 1st

```
printf("%d", num);  
}
```

6. Find errors

```
#include<stdio.h>  
main() {  
int i,j;  
for (i=1,j=0 ; i+j <=10 ; i++)  
printf("%d", i);  
j+=2;  
}
```

7. WAP to print the pattern →

```
a> $ * * * *  
*$ * * *  
* * $ * *  
* * * $ *  
* * * * $
```

```
b> @  
@ @  
@ @ @  
@ @ @ @  
@ @ @ @ @  
@ @ @ @ @  
@ @ @  
@ @ @  
@ @  
@
```

8. WAP to print the sum of the following series →

a> $1 + (1+2) - (1+2+3) + (1+2+3+4) - \dots$

b> $-x + x^2 - x^3 + x^4 \dots$

Unit 5 :-

1. How are multi dimensional arrays useful?
2. What happens when an array is initialized with (a) fewer initializers as compared to its size?
(b) more initializers as compared to its size?
3. For an array declared as `int arr[50]`, calculate the address of `arr[35]`, if `base(arr) = 1000` and `w = 2`.
4. Write a program that reads a square matrix of size `n x n`. Write a function `int isUpperTraingular(int[][],int n)` that returns 1 if the matrix is upper triangular.
5. WAP to read two floating point arrays. Merge these arrays and display the resultant array.
6. WAP to display the word HELLO in the following format

```
H  
H E  
H E L  
H E L L  
H E L L O
```

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: **Basic computation and principles of computer programming**

Subject Code: **CS101**

Year: **1st Year**

Semester: **1st**

7. Wap to count the number of charecters, words and lines in the given text.
8. WAP to find the last instance of occurrence of a given string.
9. Write a program to display a list of candidates. Prompt 100 users to cast their vote. Finally display the winner in the election.
10. In a class there are 20 students. Each student is supposed to appear in three tests and two quizzes throughout the year. Make an array that stores the names of all these 20 students. Make five arrays that stores marks of three subjects as well as scores of two quizzes for all the students. Calculate the average and total marks of each student. Display the result.

Unit 6 :-

1. How many types of storage classes does the C language support? Why do we need different types of such classes?
2. What is the difference between formal and actual parameters?
3. Write a function to reverse a string using recursion.
4. What will happen when the actual parameters are less than the formal parameter in a function?
5. WAP to compute $F(x, y)$ where $F(x, y) = F(x - y, y) + 1$ if $y \leq x$ and $F(x, y) = 0$ if $x < y$.
6. Write a function to draw the following pattern on the screen

```
* * * * *
!           !
!           !
!           !
* * * * *
```

7. Write a function to print a table of bionomial coefficients which is given by the formula

$$B(m, x) = m! / (x! (m - x)!) \text{ where } m > x,$$

$$\text{hint : } B(m, 0) = 1, B(0, 0) = 1, \text{ and } B(m, x) = B(m, x-1) * [(m - x + 1) / x]$$

8. Write a program to swap two integers using Call by Value method of passing arguments to a function.
9. Find the output –

```
#include<stdio.h>
int prod (int x, int y)
```

```
{
return (x * y);
}
main()
{
```

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Title of Course: Introduction to Programming

Course Code: BCA104

Year: 1st Year

Semester: 1st

```
int x = 2, y = 3, z;  
z = prod (x, prod(x,y) );  
printf(“%d”,z);  
return 0;  
}
```

10. Find the output

```
#include <stdio.h>  
int a;  
static int func()  
{  
    return a++;  
}  
main()  
{  
    a = 10;  
    printf(“%d”, func() );  
    a*= 10;  
    printf(“%d”, func() );  
    return 0;  
}
```

Unit 7 :-

1. Explain the result of the following code

```
int num1 = 2, num2 =3;  
int *p = &num1, *q = &num2;  
*p++ = *q++;
```

2. What do you understand by null pointer? Discuss with example.

3. Differentiate between ptr++ and *ptr++.

4. Can we subtract two pointer variables?

5. Can array names appear on the left side of the assignment operator? Why ?

6. WAP to print the lowercase letters into uppercase characters and the vice versa in the given string – “HappY NeW YeaR.”.

7. What is a dangling pointer?

8. What do you mean by wild pointer?

9. With the help of an example, explain how pointers can be used to dynamically allocate space for 2 D and 3D arrays.

10. State true or false—

- Only one call to free() is enough to release the entire array allocated using calloc().
- Ragged arrays consumes less memory space.
- *ptr++ will add 1 to the value pointed by ptr.
- Pointer constants cannot be changed.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: **Basic computation and principles of computer programming**

Subject Code: **CS101**

Year: **1st Year**

Semester: **1st**

- e. Adding 1 to a pointer variable will make it point 1 byte ahead of the memory location to which it is currently pointing.

Unit 8 :-

1. What do you mean by nested structure? Discuss.
2. Differentiate between structure and union.
3. WAP to create a structure with information given below. Then read and print the data.
Employee[10]
 - (a) Emp_id
 - (b) Name
 - (i) First name
 - (ii) Middle Name
 - (iii) Last Name
 - (c) Address
 - (i) Area
 - (ii) City
 - (iii) State
 - (d) Age
 - (e) Salary
 - (f) Designation.
4. WAP to read data from the keyboard and write it to a file. Read the contents stored in the file and display it on the screen.
5. WAP to copy a file using feof().
6. Can we have a C program that does not use any pre-processor directive?
7. What happens when the argument passed to the macro has multiple white space characters?

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: Mathematics
Year: 1st Year

Subject Code-BCA105
Semester: First

| Module Number | Topics | Number of Lectures |
|---------------|--|--------------------|
| 1 | Algebra | 9L |
| | Sets, Union, intersection, complement, mapping | 2 |
| | Polynomials, division algorithm, fundamental theorem of classical algebra (without proof), Descartes rule of signs and their application | 3 |
| | Relation between roots and coefficients, symmetric function of roots, transformation of polynomial equations, Cardan's solution of cubic equation | 4 |
| 2 | Matrices | 4L |
| | Matrices, addition and multiplication of matrices, inverse matrix | 2 |
| | Solution of linear equations in three variables by Cramer's rule, solution of three linear equations by matrix inversion method. | 2 |
| 3 | Differential calculus | 11L |
| | Limits of function and continuity, fundamental properties of continuous functions (without proof) | 2 |
| | Geometric meaning of derivative and differential, rules of differentiation | 1 |
| | successive differentiation | 1 |
| | Rolle's theorem, mean value theorem, Taylor's and Maclaurin's theorems with Cauchy's and Lagrange's forms of remainder, Taylor's series | 3 |
| | Function of several variables, partial derivatives, total differential | 2 |
| | Euler's theorem on homogeneous functions of two variables. | 2 |
| 4 | Dimensional geometry | 5L |
| | Transformation of rectangular axes, invariants, general equation of second degree – reduction to standard forms and classification, plane polar equation of a straight line, circle and conic. | 5 |

Assignment:

Module-1:

- Find the quadratic equation whose roots are $\alpha + \frac{5}{\beta}$ & $\beta + \frac{5}{\alpha}$ where α and β are the roots of the quadratic equation $2x^2 + 7x - 3 = 0$.
- Consider the polynomials $f(x), d(x)$ given by

$$f(x) = x^4 + 2x^3 + 4x^2 + x + 3$$

$$d(x) = 2x^2 + x + 3$$
 Find the quotient $q(x)$ and remainder $r(x)$ when $f(x)$ is divided by $d(x)$.
- What are symmetric functions of root?
- Prove that two sets X and Y are equal if and only if

$$X \cup Y = X \cap Y$$
- Find a quadratic equation with roots $2\alpha - 1$ and $2\beta - 1$, where α and β are the roots of the equation $4x^2 + 7x - 5 = 0$.
- Let $f: U \rightarrow V, g: V \rightarrow X, h: X \rightarrow Y$ be the three mappings then prove that

$$ho(gof) = (hog)of.$$

7. State fundamental Theorem of classical algebra.
8. Use Descartes' Rule of Signs to determine the number of possible positive and negative real solutions of the equation
 $P(x) = 2x^7 + 15x^6 + 31x^5 - x^4 - 49x^3 - 52x^2 - 78x - 36 = 0$.
9. Let $f(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_n = 0$. What will be the new equations formed when root of $f(x)$ is transformed to
 - i) $\frac{1}{x}$
 - ii) $-x$
10. State division algorithm for polynomials. Divide $f(x)$ by $g(x)$ using division algorithm where, $f(x) = 2x^4 + x^2 - x + 1$, $g(x) = 2x - 1$.

Module-2:

1. Whether the matrix $A = \begin{bmatrix} 2 & -3 & 4 \\ 1 & 0 & 1 \\ 0 & -1 & 4 \end{bmatrix}$ is invertible or not?
2. If $A = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}$ and $B = \begin{bmatrix} -3 & 2 \\ 5 & -3 \end{bmatrix}$ then calculate $\text{tr}(3A-B)$.
3. Find $\text{adj}(A)$ if $A = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix}$
4. Define symmetric and skew-symmetric matrix.
5. Prove $\begin{vmatrix} 5 & 2 & 3 \\ 7 & 3 & 4 \\ 9 & 4 & 5 \end{vmatrix} = 0$
6. Find the inverse of $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$.
7. If $A = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}$, then prove that $A^2 - 4A - 5I = 0$.
8. If $A = \begin{bmatrix} 3 & 1 & -1 \\ 0 & 1 & 2 \end{bmatrix}$, then prove AA^{-1} is symmetric.
9. If $A = \begin{bmatrix} 2 & 1 & 3 \\ 3 & 1 & 2 \\ 1 & 2 & 3 \end{bmatrix}$, verify that $A \cdot (\text{adj } A) = |A| \cdot I$, where I is the identity matrix of the same size.
10. Find the value of t for which the matrix $\begin{bmatrix} 2 & 0 & 1 \\ 5 & t & 3 \\ 0 & 3 & 1 \end{bmatrix}$ is singular.
11. Find the matrix A if $A^{-1} = \begin{bmatrix} 3 & -1 & 1 \\ 1 & -2 & 3 \\ 3 & -3 & 4 \end{bmatrix}$.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lecture-wise Plan

12. If $A = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}$, $B = \begin{bmatrix} -1 & 0 \\ 5 & 1 \end{bmatrix}$ verify $(AB)^{-1} = B^{-1}A^{-1}$.

13. Check whether the matrix $\begin{bmatrix} 2 & -3 & 4 \\ 1 & 0 & 1 \\ 0 & -1 & 4 \end{bmatrix}$ is invertible or not. Find its inverse if possible.

14. Solve by matrix Inversion method, the following system of equations:

$$2x - 3y + 4z = -4$$

$$x + z = 0$$

$$-y + 4z = 2$$

Module-3:

1. Given $u = e^{xy}$, find $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$

2. If $\sqrt{x^2 + y^2 + z^2}$, prove that $\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial z}\right)^2 = 1$

3. Write statement for Euler's Theorem

4. If $u = x^3 + y^3$, where $x = a \cos t$, $y = b \sin t$, find $\frac{du}{dt}$.

5. Write the formula for Matrix Inversion Method.

6. What is a Homogeneous function?

7. If $u = (ax + by)^2$ then find $\frac{\partial^2 u}{\partial x \partial y}$

8. Verify Euler's Theorem for $z = 3x^3 - 5x^2y + y^3$.

9. Prove that: $\vec{a} \times (\vec{b} \times \vec{c}) + \vec{b} \times (\vec{c} \times \vec{a}) + \vec{c} \times (\vec{a} \times \vec{b}) = 0$.

10. If z is a function of x & y , where $x = e^u + e^{-v}$ & $y = e^{-u} - e^v$, show that

$$\frac{\partial z}{\partial u} - \frac{\partial z}{\partial v} = x \frac{\partial z}{\partial x} - y \frac{\partial z}{\partial y}$$

11. If $\cos x = \frac{x+y}{\sqrt{x} + \sqrt{y}}$, prove that

$$x \frac{du}{dx} + y \frac{du}{dy} = -1/2 \cot x.$$

12. If $u = \log(x^2 + y^2 + z^2)$, prove that: $x \frac{\partial^2 u}{\partial y \partial z} = y \frac{\partial^2 u}{\partial z \partial x} = z \frac{\partial^2 u}{\partial x \partial y}$

13. If $u = (ax + by)^2 - (x^2 + y^2)$, where $a^2 + b^2 = 2$, prove that $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$.

14. $x = u + v$, $y = uv$ & z is a function of x, y ; show that

$$u \frac{\partial z}{\partial u} + v \frac{\partial z}{\partial v} = x \frac{\partial z}{\partial x} + 2y \frac{\partial z}{\partial y}.$$

15. Verify Euler's Theorem for the following function

$$z = \frac{xy}{x+y}$$

Module-4:

1. Determine the area of the inner loop of $r = 2 + 4\cos\theta$.
2. Determine the area that lies inside $r = 3 + 2\sin\theta$ and outside $r = 2$.
3. Determine the area that is inside both $r = 3 + 2\sin\theta$ and $r = 2$.
4. Determine the area of the region outside $r = 3 + 2\sin\theta$ and inside $r = 2$.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Title of Course: PC Software Lab

Course Code: BCA194

L-T-P scheme: 0-0-6

Course Credit: 4

Objectives:

The main objectives of PC Software Lab course are to familiarize with basic operations of:

1. Operating systems such as Windows and Linux.
2. Word Processor such as Open Office and MSWord.
3. Workbook, worksheet, graphics and Spreadsheets.
4. PowerPoint including animation and sounds.
5. Address book, Spam and Filtering in E-mail.
6. Browsing, Search, Discussion forum and Wiki's.

Learning Outcomes: The students will have a detailed knowledge of the concepts of process and memory, learn the basics behind file systems and input output systems and understand the fundamentals of network and operating systems. Upon the completion of Operating Systems practical course, the student will be able to:

- **Understand** and implement basic services and functionalities of the operating system using system calls.
- **Use** modern operating system calls and synchronization libraries in software/ hardware interfaces.
- **Understand** the benefits of thread over process and implement synchronized programs using multithreading concepts.

Course Contents:

Exercises that must be done in this course are listed below:

Section 1 : Operating System

Session 1: Familiarization (Keyboard, Memory, I/O Port),

Session 2: Windows (2 Session)

Session 3: Linux (2 Session)

Section 2: Word Processor (Open Office and MS Word)

Session 1: Basic Operations (Font selection, Justification, Spell check, Table, Indentation),

Session 2: Table of Contents, Track Changes and Commenting.

Session 3: Mail Merge, Printing, Practice session.

Section 3 : Spread sheet (Concept of Worksheet, Workbook and cell)

Session 1 : Data entry, Data editing and Formula

Session 2 : Functioning

Session 3 : Graphics and Practice session

Section 4 : PowerPoint

Session 1 : Basics operation

Session 2 : Animation and Sounds

Section 5 : E-mail

Session 1 : Basic Operation,

Session 2: Address Book, Spam and Filtering

Section 6 : Browsing and Discussion Forum

Session 1 : Browsing and Search (2 Sessions)

Session 2 : Discussion Forum, Wiki and Google Doc (3 Sessions)

Text Book:

1. Ramesh Bangia, "PC Software Made Easy (SIXTEEN-IN-ONE)", Khanna Publishing.
2. Sinha P K, Computer Fundamentals, Bpb Publications-New Delhi.

Recommended Systems/Software Requirements:

1. Microsoft Office 2007, 2010.
2. Windows XP or Linux Operating System.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Title of Course: Basic Programming Lab(C/Pascal)

Course Code: BCA193

L-T-P scheme: 0-0-6

Course Credit: 4

Introduction:

This course is designed to familiarize students with the basic components of a computer, so as to be able to operate it and be able to interact with it, and carry out simple tasks. In addition, it will initiate the students into the discipline of Programming. It aims to start off the development of problem solving ability using computer programming. This course teaches not only the mechanics of programming, but also how to create programs that are easy to read, maintain, and debug. Students are introduced to the design principles for writing good programs regardless of the hardware and the software platforms.

Objective:

Students will develop their ability to design, develop, test and document structured programs in C language.

Learning Outcomes: Students should be able to

1. Understand the basic terminology used in computer programming
2. Write, compile and debug programs in C language.
3. Use different data types in a computer program.
4. Design programs involving decision structures, loops and functions.
5. Explain the difference between call by value and call by reference
6. Understand the dynamics of memory by the use of pointers.
7. Enhance programming skills through problem solving and code development of small-size software applications.
8. Improve self-learning, teamwork and communication skills through project development practices.
9. Engage in continuing professional development under minimal guidance.

Course Contents:

Exercises that must be done in this course are listed below:

- 1 Introduction to C programming
- 2 Structured Program Development in C
- 3 Flowchart and Algorithm
- 4 C Program Control
- 5 C Functions
- 6 C Arrays
- 7 C Pointers
- 8 C Characters and Strings
- 9 C Structures, Unions, Bit Manipulations and Enumerations
- 10 C File Processing

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

CYCLE – I

1. Write a program to evaluate area of triangle using the formula $\sqrt{s(s-a)(s-b)(s-c)}$
2. Write a program to swap two numbers.
3. Write a program to find the greatest of three numbers and print the numbers in ascending order.
4. Write a program to perform the arithmetic expression using switch statement.
5. Write a program to find a factorial of given number using do while statement.
6. Write a program to print all prime numbers upto 'N' numbers.
7. Write a program to print sum of 'N' natural numbers.
8. Write a program to find the total number of even integers and odd integers of 'N' numbers.
9. Write a program to find the sum of odd numbers and even numbers upto 'N' numbers.
10. Write a program to print the product of two matrices of any order.
11. Write a program to read 'N' number of students with 5 subject marks.
12. Write a program to find greatest of 'n' numbers using functions.
13. Write a program to print Fibonacci series using recursion.
14. Write a program to convert all lower case to uppercase characters.
15. Write a program to sort 5 city names in alphabetical order.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

CYCLE II

16. Write a program to extract a string.
17. Write a program to implement the concept of call by value.
18. Write a program to implement the concept of call by reference.
19. Write a program to implement the concept of structure and union.
20. Write a program to access a variable using pointer.
21. Write a program to print the element of array using pointers.
22. Write a program to print the elements of a structure using pointers.
23. Write a program to display student information by initializing structures.
24. Write a program to pass structure as arguments to function and calculate total marks of 5 subjects.
25. Write a program to write integer data into file and read it from file.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 1

AREA OF TRIANGLE

AIM:

To write a program for evaluating the area of triangle using the formula $\sqrt{s(s-a)(s-b)(s-c)}$.

ALGORITHM:

- Step1: Start the program.
- Step2: Get the inputs a, b, c and s.
- Step3: Calculate $s = (a+b+c) / 2$.
- Step4: Calculate $\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$.
- Step 5: Print the result 'area'.
- Step 6: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<math.h>
void main()
{
    int a,b,c;
    float s,area;
    clrscr();
    printf("Enter the values of a,b,c: ");
    scanf("%d%d%d",&a,&b,&c);
    s=(a+b+c)/2;
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("The area of a triangle is =%f",area);
    getch();
}
```

OUTPUT:

Enter the values of a,b,c: 10 20 30

The area of a triangle is = 0.000000

RESULT:

Thus the C program to find the area of triangle using the formula $\sqrt{s(s-a)(s-b)(s-c)}$ has been successfully executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 2

SWAP TWO NUMBERS

AIM:

To write a program for swapping of two numbers.

ALGORITHM:

Step1: Start the program.

Step2: Get the inputs a and b.

Step3: Find $a=a+b$.

Step4: Find $b=a-b$.

Step 5: Find $a=a-b$.

Step6: Print the result 'a' and 'b'.

Step7: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b;
    clrscr();
    printf("Enter the values of a and b: ");
    scanf("%d%d",&a,&b);
    a=a+b;
    b=a-b;
    a=a-b;
    printf("The values of a and b are: %d %d", a, b);
    getch();
}
```

OUTPUT:

Enter the values of a and b: 10 20

The values of a and b are: 20 10

RESULT:

Thus the C program to swap two numbers has been successfully executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 3 GREATEST OF THREE NUMBERS AND PRINT ASCENDING ORDER AIM:

To write a program for finding the greatest of three numbers and printing the numbers in ascending order.

ALGORITHM:

- Step1: Start the program.
- Step2: Get the inputs a, b and c.
- Step3: Check if((a>b) &&(a>c))
- Step4: Again check if(b>c)
- Step5: Then print the greatest number and display a, b, c.
- Step6: Else print the greatest number and display a, c, b.
- Step7: Check if((b<c) &&(b<a))
- Step8: Again check if(c<a)
- Step9: Then print the greatest number and display b, c, a.
- Step10: Else print the greatest number and display b, a, c.
- Step11: Check if((c<a) && (c<b))
- Step12: Again check if(a<b)
- Step13: Then print the greatest number and display c, a, b.
- Step14: Else print the greatest number and display c, b, a.
- Step15: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    clrscr();
    printf("Enter the values of a, b and c: ");
    scanf("%d%d%d", &a, &b, &c);
    if(a<b && a<c)
    {
        if(b<c)
```

Lab Manual

```
        {
            printf("The greatest number is: %d", a);
            printf("The ascending order: %d%d%d", a, b, c);
        }
    else
        if(b>c)
        {
            printf("The greatest number is: %d", a);
            printf("The ascending order: %d%d%d", a, c, b);
        }
    }
    else if(b<c && b<a)
    {
        if(c<a)
        {
            printf("The greatest number is: %d", b);
            printf("The ascending order: %d%d%d", b, c, a);
        }
        else
        {
            printf("The greatest number is: %d", b);
            printf("The ascending order: %d%d%d", b, a, c);
        }
    }
    else
        if(b<a)
        {
            printf("The greatest number is: %d", c);
            printf("The ascending order: %d%d%d", c, b, a);
        }
        else
        {
            printf("The greatest number is: %d", c);
            printf("The ascending order: %d%d%d", c, a, b);
        }
    }
}
```

OUTPUT: Enter the values of a, b and c: 6 4 5

The greatest number is: 6
The ascending order: 4 5 6

RESULT:

Thus the C program to find greatest of three and to print the numbers in ascending order has been successfully executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 4

ARITHMETIC EXPRESSION USING SWITCH STATEMENT

AIM:

To write a program for performing the arithmetic expression using switch statement.

ALGORITHM:

Step1: Start the program.
Step2: Display 1. Addition 2. Subtraction 3. Multiplication and 4. Division
Step3: Get the input a and b.
Step4: Get the choice.
Step5: Switch(result)
Step6: case '+': print the sum of a & b.
Step7: case '-': print the difference of a & b.
Step8: case '*': print the multiplication of a & b.
Step9: case '/': print the division of a & b.
Step10: default: invalid option.
Step11: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b;
    int op;
    clrscr();
    printf("Enter the values of a & b: ");
    scanf("%d%d", &a, &b);
    printf(" 1.Addition\n 2.Subtraction\n 3.Multiplication\n 4.Division\n");
    printf("Enter your choice: ");
    scanf("%d", &op);
    switch(op)
    {
        case 1 :printf("Sum of %d and %d=%d", a, b, a+b);
                break;
        case 2 :printf("Subtraction of %d and %d=%d", a, b, a-b);
                break;
        case 3 :printf("Multiplication of %d and %d=%d", a, b, a*b);
                break;
        case 4 :printf("Division of %d and %d=%d", a, b, a/b);
                break;
        default : printf(" Enter Your Correct Choice.");
                break;
    }
```

```
}  
getch();  
}
```

OUTPUT:

Enter the values of a & b: 10 20

1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter your choice: 1

Sum of 10 and 20 = 30

RESULT:

Thus the C program for arithmetic expression using switch statement has been successfully executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 5 FACTORIAL OF A NUMBER USING DO WHILE STATEMENT

AIM:

To write a program for finding the factorial of a given number using do while statement.

ALGORITHM:

Step1: Start the program.
Step2: Assign $f=i=1$.
Step3: Get the input n.
Step4: do .. the following.
Step5: Find $f=f*i$
Step6: Increment $i=i+1$
Step7: Repeat from step5 to step6 till while($i \leq no$).
Step8: Then print f.
Step9: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,f;
    f=i=1;
    clrscr();
    printf("Enter a number: ");
    scanf("%d",&n);
    do
    {
        fact*=i;
        i++;
    }while(i<=no);
    printf("Factorial of %d=%d\n", no, fact);
}
```

OUTPUT:

Enter a number: 5
Factorial of 5 = 120

RESULT:

Thus the C program for finding the factorial of a given number using do while statement has been successfully executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 6

GENERATE PRIME NUMBERS UPTO N NUMBERS

AIM: To write a program for printing all prime numbers upto N numbers.

ALGORITHM:

- Step1: Start the program.
- Step2: Get the n value.
- Step3: for(i=1;i<=n;i++)
- Step4: Repeat a, b, c, d & e
 - a) Assign fact=0
 - b) for(j=1;j<=n;j++) repeat c & d
 - c) if i percentage j equal to zero
 - d) fact equal to fact added with one
 - e) if fact equal to 2 print i as prime number
- Step5: Display the prime number till nth number.
- Step6: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,fact,j;
    printf("Enter the range: ");
    scanf("%d",&n);
    printf("Prime numbers are: \n");
    for(i=1;i<=n;i++)
    {
        fact=0;
        for(j=1;j<=n;j++)
        {
            if(i%j==0)
            fact++;
            if(f==2)
            printf("%d ",i);
        }
        getch();
    }
}
```

OUTPUT:

Enter the range: 10

Prime numbers are: 3 5 7

RESULT:

Thus the C program for printing all prime numbers upto N numbers has been successfully executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 7

SUM OF N NATURAL NUMBERS

AIM:

To write a program for printing the sum of N natural numbers.

ALGORITHM:

Step1: Start the program.
Step2: Get the n value.
Step3: Initialize i=0 and sum=0.
Step4: Perform from step 5 to step 6 until i<=n
Step5: i++
Step6: sum+=i
Step7: Print the sum.
Step9: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i=0,sum=0;
    clrscr( );
    printf("Enter the Limit : ");
    scanf("%d",&n);
    while(i<=n)
    {
        i++;
        sum+=i;
    }
    printf("Sum of %d natural numbers = %d",n,sum);
    getch();
}
```

OUTPUT:

Enter the Limit : 10

Sum of 10 natural numbers = 55

RESULT:

Thus the C program for printing the sum of N natural numbers has been successfully executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 8 TOTAL NUMBER OF EVEN INTEGERS AND ODD INTEGERS OF 'N' NUMBERS

AIM: To write a program for finding the total number of even integers and odd integers of 'N' numbers.

ALGORITHM:

- Step1: Start the program.
- Step2: Declare int i, n, odd=0 and even=0;
- Step3: Get the n value
- Step4: for(i=0;i<=n;i++) do the following step.
 - a) Check if(i%2==0)
 - b) even=even+1;
 - c) Else odd=odd+1;
- Step5: Print the odd and even value.
- Step6: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,odd=0,even=0;
    clrscr();
    printf("Enter the n value: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        if(i%2==0)
            even=even+1;
        else
            odd=odd+1;
    }
    printf("The total number of odd integers =%d",odd);
    printf("The total number of even integers =%d",even);
    getch();
}
```

OUTPUT:

```
Enter the n value: 10
The total number of odd integers =5
The total number of even integers = 5
```

RESULT:

Thus the above C program for finding the total number of even integers and odd integers of 'N' numbers has been successfully executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 9 SUM OF EVEN INTEGERS AND ODD INTEGERS OF 'N' NUMBERS AIM:

To write a program for finding the sum of even integers and odd integers of 'N' numbers.

ALGORITHM:

- Step1: Start the program.
- Step2: Declare int i, n, odd=0 and even=0;
- Step3: Get the n value
- Step4: for(i=0;i<=n;i++) do the following step.
 - a) Check if(i%2==0)
 - b) even=even+i;
 - c) Else odd=odd+i;
- Step5: Print the odd and even value.
- Step6: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,n,sum,even=0,odd=0;
    clrscr();
    printf("Enter any number: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        if(i%2==0)
            even=even+i;
        else
            odd=odd+i;
    }
    printf("Sum of even integer is: %d",even);
    printf("Sum of odd integer is: %d",odd);
    getch();
}
```

OUTPUT:

```
Enter any value: 5
Sum of even integer is: 6
Sum of odd integer is: 9
```

RESULT:

Thus the C program for finding the sum of even integers and odd integers of 'N' numbers has been successfully executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 10

PRODUCT OF TWO MATRICES OF ANY ORDER

AIM:

To write a program for finding the product of two matrices of any order.

ALGORITHM:

Step1: Start the program.
Step2: Declare int Matrix A[9][9] , MatrixB[9][9] , Matrixsproduct [9][9].
Step3: Declare int n , i , j , k, Row1 , Row2 , Column1 , Column2.
Step4: Enter the order of Matrix A Row1, Column1.
Step4: Enter the order of Matrix B Row2, Column2.
Step5: Check if(Column1 == Row2)
Step6: Enter the elements of Matrix A and B using for loops.
Step7: Find Matrixproduct[i][j] = Matrixproduct[i][j] +(Matrix A[i][k] * Matrix B[k][j] using for loops.
Step7: Print the resultant matrix Matrixproduct[i][j] using for loop.
Step8: Else print invalid order so multiplication not possible.
Step9: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int Matrix A[9][9] , MatrixB[9][9] , Matrixsproduct [9][9] ;
    int n , i , j , k; /* 'i' used for rows and 'j' used for columns */ int Row1 , Row2 ,
    Column1 , Column2; clrscr();

    printf(" Enter the order of Matrix A\n");
    scanf("%d * %d " , &Row1 , &Column1);
    printf(" Enter the order of Matrix B\n");
    scanf("%d * %d " , &Row2 , &Column2);
    if(Column1 == Row2)
    {
        printf(" Enter the elements of Matrix A\n");
        for(i=0 ; i<Row1 ; i++)
        {
            for(j=0 ; j<Column1 ; j++)
            {
                scanf("%d" , &Matrix A[i][j] );
            }
        }
        printf(" Enter the elements of Matrix B\n");
```

```
for(i=0 ; i<Row2 ; i++)
{
    for(j=0 ; j<Column2 ; j++)
    {
        scanf("%d" , &Matrix B[i][j] );
    }
}
for(i=0 ; i<Row1 ; i++)
{
    for(j=0 ; j<Column2 ; j++)
    {
        Matrixproduct[i][j] = 0 ;
        for(k=0 ; k<Row2 ; k++)
        {
            Matrixproduct[i][j] = Matrixproduct[i][j] +(Matrix A[i][k] * Matrix B[k][j] );
        }
    }
}
printf(" Product Matrix\n");
for(i=0 ; i< Row1 ; i++)
{
    for(j=0 ;j< Column2;j++)
    {
        printf("%d" , Matrixproduct[i][j] );
    }
    printf("\n");
}
}
else
    printf(" Invalid order so Multiplication not possible\n");
}
```

OUTPUT:

```
Enter the order of Matrix A
2 * 2
Enter the order of MatrixB
2 * 2
Enter the elements of Matrix A
1
2
3
4
Enter the elements of Matrix B
5
6
7
8
Product Matrix
19    22
43    50
```

RESULT:

Thus the C program for finding the product of two matrices of any order has been successfully executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 11

READ 'N' NUMBER OF STUDENTS WITH 5 SUBJECT MARKS

AIM:

To write a program for reading 'N' number of students with 5 subject marks.

ALGORITHM:

Step1: Start the program.

Step2: Initialize a character array n and integer array r and s.

Step3: Initialize integer i, j and n.

Step3: Read the value of n.

Step4: for(i=0;i<n;i++)

a) Enter rollno,name,,,,,,

b) Read these and enter 5 subject marks using for loop and array.

Step5: Display n[i],r[i],s[i][j]

Step6: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char n[20][10];
    int i,j,r[20],s[20][6];
    printf("Enter n value: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter name,rollno,....");
        scanf("%s%d",&n[i],&r[i]);
        printf("Enter 5 subject marks:");
        s[i][5]=0;
        for(j=0;j<5;j++)
        {
            scanf("%d",s[i][j]);
            s[i][5]=s[i][5]+s[i][j];
        }
    }
    printf("The data entered is: \n");
    for(i=0;i<n;i++)
    {
        printf("%s\t%d\t",n[i],r[i]);
        for(j=0;j<5;j++)
            printf("%d\t",s[i][j]);
    }
    getch();
}
```

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

OUTPUT:

```
Enter n value: 1
Enter name,rollno,...Eswar 20
Enter 5 subject marks:
10 50 34 06 42
The data entered is:
Eswar 20          10      50      34      06      42
```

RESULT:

Thus the C program for reading 'N' number of students with 5 subject marks has been successfully executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 12 GREATEST OF 'N' NUMBERS USING FUNCTION

AIM: To write a program for finding greatest of 'n' numbers using function.

ALGORITHM:

- Step1: Start the program.
- Step2: Initialize integer a, b and c.
- Step3: Read the value of a,b and c.
- Step4: Call the function large().
 - a) Check if((a>b) && (a>c)) then print a is greater.
 - b) Check elseif (b>c) then print b is greater.
 - c) Check else print c is greater.
- Step5: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    printf(" Enter the value of a,b and c: ");
    scanf("%d, %d, %d", &a, &b, &c);
    large(a,b,c);
    getch();
}

large(int a, int b, int c)
{
    if((a>b) && (a>c))
        print("%d is greater than %d, %d", a, b, c);
    elseif (b>c)
        print("%d is greater than %d, %d", b, a, c);
    else
        print("%d is greater than %d, %d", c, a, b);
}
```

OUTPUT:

```
Enter the value of a,b and c: 10 30 20
30 is greater than 10, 20
```

RESULT:

Thus the C program for finding greatest of 'n' numbers using function has been successfully executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 13

FIBONACCI SERIES USING RECURSION

AIM:

To write a program for finding Fibonacci series using recursion.

ALGORITHM:

- Step1: Start the program.
- Step2: Initialize a function as int Fibonacci(int).
- Step3: Initialize integer i=0, c and n in main function.
- Step3: Read the value of n.
- Step4: Within for loop call the Fibonacci(int) recursively.
- Step5: In Fibonacci(int) function calculate (Fibonacci(n-1) + Fibonacci(n-2)) recursively and return the value.
- Step6: Print the result.
- Step7: Stop the program.

PROGRAM:

```
#include<stdio.h>
int Fibonacci(int);
int main()
{
    int n, i = 0, c;
    printf("Enter the n value: ");
    scanf("%d",&n);
    printf("Fibonacci series\n");
    for ( c = 1 ; c <= n ; c++ )
    {
        printf("%d\n", Fibonacci(i));
        i++;
    }
    return 0;
}

int Fibonacci(int n)
{
    if ( n == 0 )
        return 0;
    else if ( n == 1 )
        return 1;
    else
        return ( Fibonacci(n-1) + Fibonacci(n-2) );
}
```

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR
Lab Manual

OUTPUT:

Enter the n value: 9

Fibonacci series: 0 1 1 2 3 5 8 13 21

RESULT:

Thus the C program for finding Fibonacci series using recursion has been successfully executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 14

LOWER CASE TO UPPERCASE CHARACTERS

AIM: To write a program for converting all lower case to uppercase characters.

ALGORITHM:

Step1: Start the program.
Step2: Take a string a function of return value data type is void str upper.
Step3: Read a string.
Step4: While (s[i] != '\0') the do the following
 a) if((s[i] >= 'a') && (s[i] <= 'z'))
 b) s[i] = s[i] - 32;
 c) i++;
Step5: Display changed string.
Step6: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char str;
    printf("Enter a string: ");
    scanf("%s",str);
    to_str_upper(char[]);
    printf("Changed to: %s",str);
}
void to_str_upper(char[])
{
    int i=0;
    while(s[i]!='\0')
    {
        if((s[i]>='a') && (s[i]<='z'))
            s[i]=s[i]-32;
        i++;
    }
}
```

OUTPUT:

Enter a string : gnec
changed to: GNEC

RESULT:

Thus the C program for converting all lower case to uppercase characters has been successfully executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 15 **SORT 5 CITY NAMES IN ALPHABETICAL ORDER**

AIM:

To write a program for sorting 5 city names in alphabetical order.

ALGORITHM:

Step1: Start the program.

Step2: Using for loop and array get the city name.

Step3: Using loop for(i=65;i<122;i++) and for(j=0;j<5;j++)

a) Check if(city[j][0]==i)

b) Display the sorted list of cities.

Step4: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    ch city[5][20];
    int i,j;
    clrscr();
    printf("Enter the names of cities...\n\n");
    for(i=0;i<5;i++)
        scanf("%s",&city[i]);
    printf("Sorted list of cities...\n\n");
    for(i=65;i<122;i++)
    {
        for(j=0;j<5;j++)
        {
            if(city[j][0]==i)
                printf("\n%s",city[j]);
        }
    }
}
```

OUTPUT:

```
Enter the names of cities: Hyderabad Chennai Bombay Goa Vizag
Sorted list of cities:
Bombay
Chennai
Goa
Hyderabad
Vizag
```

RESULT: Thus the C program for sorting 5 city names in alphabetical order has been successfully executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 16

EXTRACTS THE PART OF A STRING

AIM:

To write a program for extracting the part of a string.

ALGORITHM:

- Step1: Start the program.
- Step2: Declare the character array s[30] and r[30].
- Step3: Declare the integer variables i, j, m & n.
- Step4: Get the input string using gets().
- Step5: Get the value of m and n for extracting from the input string.
- Step6: Initialize j=0.
- Step7: Using a loop for(i=n-1;i<m+n-1;i++)
 - a) Assign r[j]=s[i];
 - b) Increment J by 1.
- Step8: Print the extracted part of the string.
- Step9: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void main()
{
    char s[30],r[30];
    int i,j,m,n;
    clrscr();
    printf("Enter a string: ");
    gets(s);
    printf("Enter the values of m & n: ");
    scanf("%d%d",&m,&n);
    j=0;
    for(i=n-1;i<m+n-1;i++)
    {
        r[j]=s[i];
        j++;
    }
    printf("The extracted part of string %s: ",r);
    getch();
}
```

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR
Lab Manual

OUTPUT:

Enter a string: Gurunanak
Enter the values of m & n: 3 5
The extracted part of string: run

RESULT:

Thus the C program for extracting a part from the given string was executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 17

CALL BY VALUE

AIM:

To write a program to increment the value of an argument using call by value.

ALGORITHM:

- Step1: Start the program.
- Step2: Declare the integer variable x and a integer function incr()
- Step3: Initialize x=7.
- Step4: Pass the x value to the function incr(x).
 - a) Within the function increment the x value by 1.
 - b) Return the value.
- Step5: Print the original value and incremented value of x.
- Step6: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<string.h>
main()
{   int x;
    int incr(int n);
    printf("***Call by Value***\n");
    x = 7;
    printf("Original value of x is: %d/n: ", x);
    printf("Value of incr(x) is: %d/n ", incr(x));
    printf("The value of x is: %d/n: ", x);
}

/* Function increments n */
int incr(int n)
{
    n = n + 1;
    return n;
}
```

OUTPUT:

```
Original value of x is: 7
Value of incr(x) is : 8
The value of x is: 7
```

RESULT:

Thus the C program to increment the value of an argument using call by value was executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 18

CALL BY REFERENCE

AIM:

To write a program for swapping two values using call by reference method.

ALGORITHM:

Step1: Start the program.
Step2: Assign the integer variable a=10 and b=20.
Step3: Call the swap() function.
Step4: Swap the values using pointer.
Step5: Print the original value and swapped value of a & b.
Step6: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void swap( int *x, int *y )
{
    int t ;
    t = *x ;
    *x = *y ;
    *y = t ;
    printf( "\nx = %d y = %d", *x,*y);
}

int main( )
{
    int a = 10, b = 20 ;
    swap ( &a, &b ) ;
    printf ( "\na = %d b = %d", a, b ) ;
    getch();
}
```

OUTPUT:

```
a=10 b=20
x=20 y=10
```

RESULT:

Thus the C program to swap two values using call by reference method was executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 19(a)

STRUCTURE

AIM:

To write a program for displaying student information by initializing structures.

ALGORITHM:

Step1: Start the program.

Step2: Initialize a structure student with name as character array and roll number and age as integer.

Step3: In the main program create a object s1 for the structure student.

Step4: Using the object s1 print the student name, roll number and age.

Step6: Stop the program.

PROGRAM:

```
#include<stdio.h>
struct student
{
    char name[10];
    int rollno;
    int age;
};
main()
{
    static struct student s1;
    clrscr();
    printf("Enter the name, rollno & age");
    scanf("%s%d%d\n",&s1.name,&s1.rollno,&s1.age);
    printf("%s %d %d",s1.name,s1.rollno,s1.age); getch();
}
```

OUTPUT:

```
Enter name, rollno & age
Ravi 11 25
Ravi 11 25
```

RESULT:

Thus the C program to display student information by initializing structures was executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 19(b)

UNION

AIM:

To write a program for implementing the concept of union data type.

ALGORITHM:

Step1: Start the program.

Step2: Initialize a union Data with Str as character array, i as integer and f as float.

Step3: In the main program create a variable name data for the union Data.

Step4: Using the variable and member access operator print all the members of the union Data.

Step5: Stop the program.

PROGRAM:

```
#include <stdio.h>
#include <string.h>
union Data
{
    int i;
    float f;
    char str[20];
};

int main( )
{
    union Data data;
    data.i = 10;
    printf( "data.i : %d\n", data.i);
    data.f = 220.5;
    printf( "data.f : %f\n", data.f);
    strcpy( data.str, "C Programming");
    printf( "data.str : %s\n", data.str);
    return 0;
}
```

OUTPUT:

```
data.i : 10
data.f : 220.500000
data.str : C Programming
```

RESULT:

Thus the C program to implement the concept of union data type was executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 20

ACCESS THE VALUE OF VARIABLES USING POINTER

AIM:

To write a program for accessing the value of variables using pointer.

ALGORITHM:

Step1: Start the program.
Step2: Declare integer as a, b, c and two pointer variables *p1 & *p2.
Step3: Initialize a=12 and b=4.
Step4: Assign the a & b values to the pointer variables p1 & p2.
Step5: Perform arithmetic operations.
Step6: Print the address of a & b and print the a, b, c, x & y values.
Step7: Stop the program.

PROGRAM:

```
#include<stdio.h>
main()
{
    int a,b,*p1,*p2,x,y,z;
    clrscr();
    a=12,b=4;
    p1=&a; p2=&b;
    x=*p1**p2-6;
    y=(4-*p2)**p1+10;
    printf("Address of a=%d\n",p1);
    printf("Address of b=%d\n",p2);
    printf("a=%d,b=%d\n",a,b);
    printf("x=%d,y=%d\n",x,y);
    *p2=*p2+3; *p1=*p2-5;
    z=*p1**p2-6;
    printf("a=%d,b=%d\n",a,b);
    printf("z=%d\n",z);
    getch();
}
```

OUTPUT: Address of a = 65543

Address of b = 64455

a = 12 b = 4

x = y =

z=42

RESULT:

Thus the C program to access the value of variables using pointer was executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 21 **PRINT THE ELEMENT OF ARRAY USING POINTERS**

AIM:

To write a program for printing the element of array using pointers.

ALGORITHM:

Step1: Start the program.
Step2: Declare integer array a[5] and a pointer variable *p=&a[0]
Step3: Initialize i as integer.
Step4: Using the for loop for(i=0;i<5;i++)
Step5: Print the value of *(p+i).
Step6: Then using the for loop for(i=0;i<5;i++)
Step7: Print the value of (p+i).
Step8: Stop the program.

PROGRAM:

```
#include<stdio.h>
main()
{
    int a[5]={5,4,6,8,9};
    int *p=&a[0];
    int i;
    clrscr();
    for(i=0;i<5;i++)
        printf("%d",*(p+i));
    for(i=0;i<5;i++)
        printf(" %u\n",p+i);
    getch();
}
```

OUTPUT:

```
1 2 3 4 5
1 2 3 4 5
```

RESULT:

Thus the C program to print the element of array using pointers was executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 22

PRINT THE ELEMENTS OF A STRUCTURE USING POINTERS

AIM:

To write a program printing the elements of a structure using pointers.

ALGORITHM:

Step1: Start the program.
step2: Take a character array name, a number and price in structure
step3: In main take a struct variable product and a pointer
Step4: Using a loop for(*ptr=product;ptr<product+3;ptr++)
Step5: Read the value by using array operator
ptr->name,ptr->no,ptr->price
step6: Display name,no,price.
Step7: Stop the program.

PROGRAM:

```
#include<stdio.h>
struct invest
{
    char name[20];
    int number;
    float price;
};
main()
{
    struct invest product[3],*ptr;
    clrscr();
    printf("input\n\n");
    for(*ptr=product[3];ptr<product+3;ptr++)
        scanf("%s%d%f",&ptr->name,&ptr->number,&ptr->price);
    printf("\nResult: \n\n");
    ptr=product;
    while(ptr<product+3)
    {
        printf("%20s%5d%10.2f\n",ptr->name,ptr->number,ptr->price);
        ptr++;
    }
    getch();
}
```

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR
Lab Manual

OUTPUT:

Raja

11

120

Result:

Raja

11

120

RESULT:

Thus the C program to print the elements of a structure using pointers was executed and verified.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 23 DISPLAY COLLEGE ADDRESS USING STRUCTURES AND POINTERS AIM:

To write a program for displaying college address using structures and pointers.

ALGORITHM:

- Step1: Start the program.
- Step2: Take name, location and city inside the collegeaddress structure.
- Step3: Enter the required data.
- Step4: Print the result.
- Step5: Stop the program.

PROGRAM:

```
#include<stdio.h>
struct collegeaddress
{
    char name[20],location[20],city[20];
};
main()
{
    struct collegeaddress add,*ptr;
    p=&add;
    p->name={"Annamalai University"};
    p->location={"Annamalainagar"};
    p->city={"Chidambaram"};
    printf("%s%s%s",p->name,p->location,p->city);
}
```

OUTPUT:

Annamalai University Annamalainagar Chidambaram

RESULT:

Thus the C program to display college address using structures and pointers was executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 24 **PASS STRUCTURE AS ARGUMENT TO FUNCTION**

AIM:

To write a program for passing structure as argument to function and calculate total marks of 5 subjects.

ALGORITHM:

- Step1: Start the program.
- Step2: Inside the structure ex2 declare 6 integers.
- Step3: Declare structure ex2 as s1.
- Step4: Declare structure ex2 as s2, ex2 as fun().
- Step5: Display the message as enter the marks.
- Step6: Take value of the subjects from the user.
- Step7: Store the return value in s2.total.
- Step8: Print the value of s2.total.
- Step9: Stop the program.

PROGRAM:

```
#include<stdio.h>
struct ex2
{
    int m1,m2,m3,m4,m5,total;
};
main()
{
    struct ex2 s1;
    struct ex2 s2;
    struct ex2 fun();
    printf("enter the marks");
    scanf("%d%d%d%d%d",&s1.m1,&s1.m2,&s1.m3,&s1.m4,&s1.m5);
    s2=fun(s1);
    printf("%d",s2.total);
}
struct ex2 fun(s1)
struct ex2 s2;
{
    s2.total=s1.m1+s1.m2+s1.m3+s1.m4+s1.m5;
    return(s2);
}
```

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Lab Manual

OUTPUT:

Enter the marks
10 20 30 40 50
150

RESULT:

Thus the C program to pass structure as argument to function and calculate total marks of 5 subjects was executed and verified.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Ex. No. 25 WRITE INTEGER DATA INTO FILE AND READ IT FROM FILE

AIM:

To write a program for writing integer data into file and read it from file.

ALGORITHM:

Step1: Start the program.
Step2: Initialize integer num.
Step3: Declare FILE *f2.
Step4: Open the file f2 using fopen() in write mode.
Step5: Get the integer from user and write it into the file using putw().
Step6: Close the file.
Step7: Open the file f2 using fopen() in read mode.
Step8: Read the integer using getw().
Step9: Print the integer.
Step10: Close the file.
Step11: Stop the program.

PROGRAM:

```
#include<stdio.h>
main()
{
    int num;
    FILE *f2;
    f2=fopen("data.int","w");
    scanf("%d",&num);
    putw(num,f2);
    fclose(f2);
    f2=fopen("data.int","r");
    num=getw(f2);
    printf("%d",num);
    fclose(f2);
}
```

OUTPUT:

12
12

RESULT:

Thus the C program to write integer data into file and read it from file was executed and verified.

