



P.G.K. Mandal's

HARIBHAI V. DESAI COLLEGE

(Affiliated to Savitribai Phule Pune University)

F.Y.B.C.A.(Science) Semester-I

Lab Course –I & II

Work Book

Name:_____

Roll No:_____

Academic Year:_____



Savitribai Phule Pune University

F.Y.B.C.A.(Science)

Semester-I

Section -I

BCA – 105

Lab Course –I

C Programming

INTRODUCTION

1. About the Workbook:

This workbook is intended to be used by FYBCA (Science) students for the C language and computer fundamentals Assignments in Semester-I. This workbook is designed by considering all the practical concepts / topics mentioned in syllabus.

2. The objectives of this Workbook are:

- 1) Defining the scope of the course.
- 2) To bring the uniformity in the practical conduction and implementation in all colleges affiliated to SPPU.
- 3) To have continuous assessment of the course and students.
- 4) Providing ready reference for the students during practical implementation.
- 5) Provide more options to students so that they can have good practice before facing the examination.
- 6) Catering to the demand of slow and fast learners and accordingly providing the practice assignments to them.

3. How to use this Workbook:

The workbook is divided into two sections. Section-I is related to Basic C programming assignments and Section II is related to Computer Fundamentals. The Section-I (C-programming) is divided into 10 assignments. Each assignment has three SET's. It is mandatory for students to complete all the SET's in given slot. The Section-II (Computer Fundamentals) is divided into 12 assignments. Each assignment has several SET's . It is mandatory for students to complete all the SETs in given slot.

4. Instructions to the students:

Please read the following instructions carefully and follow them.

- 1) Students are expected to carry this workbook every time they come to the lab for practical.
- 2) Students should prepare for the assignment by reading the relevant material.
- 3) Instructor will specify which problems to solve in the lab during the allotted slot and student should complete them and get verified by the instructor. However, student should spend additional hours in Lab and at home to cover all workbook assignments if needed.
- 4) Students will be assessed for each assignment on a scale from 0 to 5

Not done	0
Incomplete	1
Late Complete	2
Needs improvement	3
Complete	4
Well Done	5

5. Instruction to the Instructors:

- 1) Make sure that students should follow above instructions.
- 2) Explain the assignment and related concepts using white board if required or by demonstrating the software.
- 3) Evaluate each assignment carried out by a student on a scale of 5 as specified above by ticking appropriate box.
- 4) The value should also be entered on assignment completion page of the respective Lab course.

6. Instructions to the Lab administrator:

- 1) You have to ensure appropriate hardware and software is made available to each student.
- 2) The operating system and software requirements on server side and also client side areas given below:
 - a. Operating System: Linux and Windows
 - b. Microsoft Office

TABLE OF CONTENTS

Sr.No.	Assignment
1	Assignment on use of data types, simple operators (expressions)
2	Assignment on decision making statements (if and if-else, nested structures)
3	Assignment on decision making statements (switch case)
4	Assignment on use of Simple loops
5	Assignment on nested loops
6	Assignment on Standard Library Function and menu driven programs.
7	Assignment on user defined functions.
8	Assignment on recursive functions
9	Assignment on use of arrays (1-D array) and functions
10	Assignment on use of multidimensional array (2-D array) and functions

ASSIGNMENT COMPLETION SHEET

Lab Course I- Basic C Assignments

Sr.No.	Assignment	Marks (out of 5)	Teachers Sign
1	Assignment on use of data types, simple operators (expressions)		
2	Assignment on decision making statements (if and if-else, nested structures)		
3	Assignment on decision making statements (switch case)		
4	Assignment on use of Simple loops		
5	Assignment on nested loops		
6	Assignment on Standard Library Function and menu driven programs.		
7	Assignment on user defined functions.		
8	Assignment on recursive functions		
9	Assignment on use of arrays (1-D array) and functions		
10	Assignment on use of multidimensional array (2-D array) and functions		
Total (Out of 50)			
Total (Out of 20)			
Internal Viva(10)			
Total (Assignment(20)+Internal Viva(10))Out of 30			

This is to certify that Mr./Ms. _____
has successfully completed the basic programming in C language for
Lab course I and has scored ____ marks out of 30.

Lab In-charge

H.O.D

Internal Examiner

External Examiner

Assignment 1

Study of data types and simple operators

You should read following topics before starting this exercise.

- Basic datatypes in C & variable declaration rules
- Operators and operator precedence in C

Basic Datatypes

Data format	Data types	Format String	Declaration	Example
Numeric	int Short int Long int	%d %d %ld	int age; short qty; long mbno;	printf(“%d”,age); printf(“%d”,qty); printf(“%ld”,mbno);
Real	float double	%f	float area; double price;	scanf(“%f”,&area); printf(“%f”,area);
Character	char	%c	char ch;	scanf(“%c”,&ch); printf(“%c”,ch);

Few Operators:

Type of operator	Operators	Description
Arithmetic	+ - / * %	Addition Subtraction Division Multiplication Modulus/Reminder
Relational	> < >= <= == !=	Greater than Less than Greater than equal to Less than equal to Equal to Not equal to
Logical	&& !	Logical AND Logical OR Logical NOT

Expression examples:

Expression	C Expression
$a \times b - c \times d$	$a * b - c * d$
$(i + j) (k + m)$	$(i + j) * (k + m)$
$3x^2 + 2x + 5$	$3*x*x + 2*x + 5$
Increment a by 5	$a=a+5$

Sample Program: Write a program to input principal, number of years, and rate of interest and calculate Simple interest.

Algorithm	Flowchart	Program
1. Start 2. Input principal amt p, interest rate r, no of years n 3. Simple Interest = $(p*n*r)/100$ 4. Output Simple Interest 5. Stop	<pre> graph TD Start([Start]) --> Input[/Input p,n,r/] Input --> Process[SI=p*n*r/100] Process --> Output[/Print SI/] Output --> Stop([Stop]) </pre>	<pre> #include<stdio.h> void main() { float p,n,r,si; printf("Enter Principal amt: "); scanf("%f",&p); printf("Enter rate of interest: "); scanf("%f",&r); printf("Enter no of years: "); scanf("%f",&n); si=p*n*r/100; printf("Simple Interest : %f",si); } </pre>

SET A:

1. Accept 2 numbers and print arithmetic and harmonic mean of two numbers.
2. Accept dimensions of a cylinder. Calculate and print surface area and volume of cylinder.
3. Accept initial velocity u, acceleration a, and time t. Calculate and print Final velocity v and distance s travelled.
4. Accept temperature in Fahrenheit(F). Convert and print in celcius(C) (Hint: $C = 5/9(F-32)$)
5. Accept 2 numbers and display their squares on one line, cube on another and square root on third line.
6. Accept a character and display its ASCII value. Also display the ASCII value of space, tab, enter.
7. Accept a character from keyboard and display its previous and next character in order.

Signature of Instructor

Date:

SET B:

1. Accept 2 integers from user and interchange them. Display interchanged numbers.
2. Accept 2 integers and display its division, reminder, and exact division.
3. Accept 4 integers a,b,c,d. Add numbers a & b. Subtract d from c. Then divide sum a & b by result of subtraction. Display exact answer.

Signature of Instructor

Date:

SET C:

1. Accept length l, breadth b, and height h of cuboid . Calculate and print surface area and volume
2. Accept radius from user and display area and circumference of a circle.
3. Accept an integer from user and display it along with its octal and hexadecimal equivalent separately.

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 2

Decision Making statements(if, if-else, and nested structures)

You should read following topics before starting this exercise.

- Basics and types of decision making statements
- Syntax of decisions making statements

Normally, program flows sequentially in the order in which it appears in a source code. sometimes, it is required to execute a particular portion of code depending on result of some condition(True or false) This is known as branching or decision making statements. Some of them are;

1. If statement
2. If-else statement
3. Nested if

Type	Syntax	Flowchart	Example
if statement: statement enables you to test for a condition and branch to different parts of your code, depending on the result. An expression may consist of logical or relational operators	if(condition) statement; or if(condition) { statements; }		<pre>void main() { int var; printf("Enter any number;"); scanf("%d",&var); if(var==10) printf("The user entered number is Ten"); }</pre>
if-else statement: Statement1 will be executed if condition/expression results true otherwise statement2 will be executed	if(condition) { Statements1; } else { Statements2; }		<pre>void main() { int a,b; printf("enter 2 numbers") ; scanf("%d%d",&a,&b); if(a>b) printf("%d is greater",a); else printf("%d is greater",b); }</pre>
Nested if: if statement can be nested in if or else.	if(condition) { if(condition) { Statements1;} else { Statements2;} } else { if(condition) { Statements1;} else { Statements2;} }		<pre>void main() { int a,b,c; printf("enter 3 numbers") ; scanf("%d%d%d",&a,&b,&c); if(a>b && a>c) { printf("%d is greater",a); } else { if(b>c) printf("%d is greater",b); else printf("%d is greater",c); } }</pre>

Conditional (Ternary) Operator

The conditional operator (?:) is ternary operator; it is the only operator to take three terms.

Syntax: (expression1) ? (expression2) : (expression3);

This line is read as "If expression1 is true, return the value of expression2; otherwise, return the value of expression3." Typically, this value would be assigned to a variable.

For Example: void main(void)

```
{
    float per;
    printf("\n Enter your percentage;");
    scanf("%f",&per);
    printf("\n you are");
    printf("%s", per >= 60 ? "Passed": "Failed");
}
```

SET A

1. Accept an integer from user and checks whether the number is even or odd (Using if-else)
2. Accept 3 numbers from user and display maximum
3. Accept 3 numbers and check whether the first number is in between other 2 numbers.
4. Accept 3 numbers. Subtract middle number from others and then find greatest among 3 numbers.
5. Accept 4 numbers as input from user and display minimum
6. Accept an integer and check if it divisible by 5 and 7.
7. Accept a character and check whether it is a vowel or consonant.(vowels : a,e,i,o,u)
8. Accept a character. Check if it is uppercase then convert it into lowercase and vice versa.

Signature of Instructor

Date:

SET B

1. Write a program to check whether a given year is leap year or not.
2. Accept marks of 5 subjects and calculate total marks & percentage and print grade of a student.
3. Accept a character and check if it is a digit or lowercase or uppercase character(Hint : Use ASCII values)
4. Accept basic salary of employee. Calculate and display gross salary of employee.
If basic salary > 50,000 DA(dearness allowance) is 25% and HRA (house rent allowance) is 20% of basic; Otherwise DA is 10% and HRA is 15% of basic. (Hint: gs = basic + DA + HRA)
5. Accept 2 sides of a rectangle. Check whether it is a rectangle or square, Display its area.

Signature of Instructor

Date:

SET C : Using ternary operator

1. Accept an integer from user and checks whether the number is even or odd.
2. Accept an 2 integers from user and display their greatest.
3. Accept an integer and display message as “Positive Number”, ”Negative Number” or “Zero” depending on value of a number.

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 3

Decision Making Statements (Switch Case)

The switch statements allow you to branch on any of a number of different values. It is a multi-way decision making statement.

Syntax:	Sample Program
<pre>switch (expression) { case value1: statement; break; case value2: statement; break; : : default: default block; }</pre>	<pre>main() { char grade; printf("\n Enter your Grade: "); grade=getchar(); switch(grade) { case 'A': case 'a': printf("\n Your percentage is 80 or above 80 "); break; case 'B': case 'b': printf("\n Your percentage is in 70-80 "); break; default: printf("\n Your percentage is below 70 "); } }</pre>

Important Points about Switch Statement

- Switch case performs equality check of the value of expression/variable against the list of case values.
- The expression in switch case must evaluate to return an integer or character.
- You can use any number of case statements within a switch. The expression value is compared with the constant after case.
- The data type of the value of expression/variable must be same as the data type of case constants.
- The break statement is optional. If break statement is not used, all statements below that case statement are also executed until it found a break statement.
- The default code block gets executed when none of the case matches with expression. default case is optional and doesn't require a break statement.

SET A

1. Accept the week day as number from user and display Monday to Sunday.
2. Accept a single digit and display it in words. (Ex. If input is 5 then display Five)

Signature of Instructor

Date:

SET B

1. Accept 2 integers and a users choice to perform addition, Subtraction, Multiplication, Division. Perform the operation on 2 numbers as per users choice.
2. Write a program to input a number and choice from user to perform
 - a. Factorial(Calculate factorial of no)
 - b. Even/odd (Check whether the no is even or odd)
 - c. Perfect (Check whether the no is perfect or not)

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 4

Simple Loops (while, do-while, for)

A loop is used for executing a block of statements repeatedly until a given condition returns false. Types of Loops are ;

Syntax	Flowchart	Example
While loop:- <pre>while(condition) { Statement 1; Statement 2; : : }</pre>	<pre> graph TD Entry(()) --> Test{Test expression} Test -- true --> Body[Body of while Loop] Body --> Test Test -- false --> Exit[Statement just below while] Exit --> Exit2(()) style Entry fill:none,stroke:none style Exit2 fill:none,stroke:none </pre>	<pre>//Print 1 to 5 numbers #include <stdio.h> void main() { int count=1; while (count <= 5) { printf("%d ", count); count++; } }</pre>
do-while loop: <pre>do { Statement 1; Statement 2; : : } while(condition);</pre>	<pre> graph TD Entry(()) --> Body[Body of Loop] Body --> Test{Test expression} Test -- true --> Body Test -- false --> Exit[Statement just below Loop] Exit --> Exit2(()) style Entry fill:none,stroke:none style Exit2 fill:none,stroke:none </pre>	<pre>//Print 0 to 5 numbers #include <stdio.h> void main() { int j=0; do { printf("Value of variable j is: %d\n", j); j++; } while (j<=5); }</pre>
for loop: <pre>for (initialization; condition test; increment or decrement) { Statement 1; Statement 2; : : }</pre>	<pre> graph TD Init([Initilization statement]) --> Test{Test expression} Test -- True --> Body[Body of for Loop] Body --> Update[Update statement] Update --> Test Test -- False --> Exit([Exit for Loop]) Exit --> Exit2[Statement just below for Loop] Exit2 --> Exit3(()) style Init fill:none,stroke:none style Exit3 fill:none,stroke:none </pre>	<pre>//Calculate & print sum of 1 to 10 #include <stdio.h> void main() { int i,sum=0; for(i=1; i<=10 ; i++) { sum=sum + i; } printf("Sum of 1 to 10 =%d", sum); }</pre>

SET A

1. Accept 2 numbers m and n. Display all numbers between m & n which are divisible by 3 and not divisible by 7.
2. Accept a number n and display sum and average of first n numbers
3. Write a program to accept n integers and count number of odds and evens.
4. Accept n integers and count numbers greater than 999 and numbers less than 99. Display both counts.
5. Accept n integers. Calculate and display sum of odds and evens.
6. Write a program to accept n and display n numbers of Fibonacci series.
7. Write a program to accept an integer n. count number of digits in it and display it.
8. Write a program to accept 2 integers and find & display their GCD.

Signature of Instructor

Date:

SET B

1. Write a program to accept an integer and display all its divisors.
2. Write a program to accept an integer and display the sum of all its divisors.
3. Write a program to check whether a given number is perfect number or not.
4. Write a program to calculate sum of digits of a given number.
5. Accept a number and find sum of its first and last digit.

Signature of Instructor

Date:

SET C

1. Write a program to accept a number and display reverse of a number.
2. Write a program to check whether a given number is Armstrong number or not(hint: sum of cube of digits= number then it is Armstrong number).
3. Write a program to check whether a given number is Palindrome number or not.
4. Write a program to check whether a given number is prime number or not.

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 5

Nested Loops

Nested loops means a loop that is contained within another loop. Nesting can be done upto any levels. The Inner loop has to be completely enclosed in outer loop. Overlapping of loops is not allowed

Format	Example program
Nested for: <pre>for (init; condition; increment) { for (init; condition; increment) { statement(s); } statement(s); }</pre>	<pre>//Print Floyd's triangle #include <stdio.h> void main () { int i, j; for(i = 1; i<=5; i++) { for(j = 1; j <= i; j++) printf("%d\t",i); printf("\n"); } }</pre>
While Loop: <pre>while(condition){ while(condition){ set of Statements } set of Statements }</pre>	<pre>/* program to calculate sum of digits till sum is a single digit #include<stdio.h> void main() { int n,sum; printf("Enter number : "); scanf("%d",&n); printf("%d → ",n); do{ sum=0; while(n>0) { sum=sum+n%10; n=n/10; } n=sum; }while(n>9); printf("%d",n); }</pre>
do-while loop: <pre>do { statement(s); do { statement(s); }while(condition); }while(condition);</pre>	

Note: It is possible to nest any loop within another type of loop. For example, a 'for' loop can be inside a 'while' loop/do-while loop or vice-versa.

SET A:

1. Write a program accept n1 & n2 and Display all prime numbers between n1 & n2
2. Write a program to display all Armstrong numbers between 1 to 1000.
3. Write a program to display all perfect numbers between 1 to 500.(Hint: Perfect number is a number such that sum of divisors is equal to the number itself)

4. Write a program to display multiplication tables having n multiple of each. Output should be displayed in tabular format.
5. Write a program to calculate sum of digits of a given number till the sum reaches to single digit.(hint: $n=3456 \rightarrow 3+4+5+6=18 \rightarrow 1+8=9$)
6. Write a program to find answer of following series.
 $1/2^2 + 2/3^2 + 3/4^2 + 4/5^2 + 5/6^2 \dots\dots\dots$

Signature of Instructor

Date:

SET B : Write a program to print the following patterns.

1.

```

21
22    23
24    25    26
27    28    29    30
31    32    33    34    35

```

2.

```

AA
BB    CC
DD    EE    FF
GG    HH    II    JJ
KK    LL    MM    NN    OO

```

3.

```

  1
0  1
1  0    1
0  1    0    1
1  0    1    0    1

```

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 6

Standard Library functions

The standard functions are built-in functions. In C programming language, the standard functions are declared in header files. The standard functions are also called as library functions or pre-defined functions. When we use standard functions, we must include the respective header file using #include statement.

For example, the function printf() is defined in header file stdio.h (Standard Input Output header file). Few examples of header files with standard functions provided by C Programming language are

math.h: This contains function prototypes for performing various mathematical operations on numeric data.

Function Name	Description	Example
exp(double x)	Exponential function, computes e^x	exp(x)
log	Natural log	C=log(x)
log ₁₀	Base 10 logarithm	A=log ₁₀ (x)
cos	Cosine	cos(angle)
sin	Sine	sin(angle)
sqrt	Square root	Y=sqrt(9)
pow(x,y)	Computes x^y	Y=pow(2,3)

The math library file need to be linked to your program. To link it compile the program as follows
\$cc filename -lm

ctype.h: Contains function prototypes for performing various operations on characters

Function Name	Description	Example
isalpha	Check whether a character is a alphabet	if(isalpha(ch))
isalnum	Check whether a character is a alphanumeric	if(isalnum(ch))
isdigit	Check whether a character is a digit	if(isdigit(ch))
isspace	Check whether a character is a space	if(isspace(ch))
ispunct	Check whether a character is a punctuation mark	if(ispunct(ch))
isupper	Check whether a character is a uppercase char	if(isupper(ch))
islower	Check whether a character is a lowercase char	if(islower(ch))
toupper	Converts character to uppercase	Ch=toupper(ch)
tolower	Converts character to lowercase	Ch=tolower(ch)

If any of the above function is used in a program the header file must be included as;
#include<math.h> or #include<ctype.h>

SET A

1. Write a program to check whether a given character is upper case. If it is upper case convert it to lower case and vice versa
2. Accept a character and check if it is an alphabet, digit, or punctuation mark. If it is an alphabet, check if it is uppercase or lowercase.

Signature of Instructor

Date:

SET B

1. Write a program to accept an angle value x and display its sin, cos, and tan.
2. Write a program to find Square root of a given number and power of x,y(input x & y)

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 7

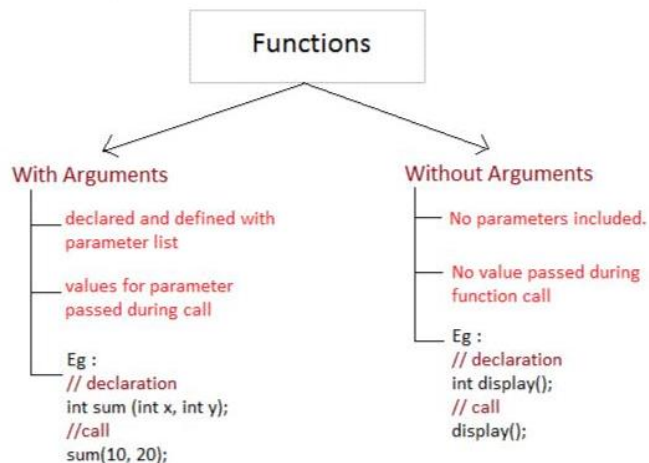
User Defined Functions

A function is a block of code that performs a particular task. The functions defined by the user are known as User-defined Functions

Actions involving functions	Syntax	Example
Function Declaration	returntype functionName(type1 parameter1, type2 parameter2,...);	void display(); OR int sum(int x,int y);
Function Definition	returntype functionName (type1 parameter1, type2 parameter2,...) { // function body goes here }	int sum(int x,int y) { int c=x+y; return c; }
Function Call	functionName(arguments); or var= functionName(arguments);	display(); ans=sum(6,7);

Passing Arguments to a function

Arguments are the values specified during the function call, for which the formal parameters are declared while defining the function.



Sample Program:

```
/* Program to calculate addition and subtraction of 2 given numbers
#include<stdio.h>
int sum(int a,int b){
    return(a+b);
}
int sub(int a,int b){
    return(a-b);
}
void main()
{
    int n1,n2,add,sub;
    printf("Enter 1st number:");
    scanf("%d",&n1);
```

```

printf("Enter 2nd number:");
scanf("%d",&n2);
add=sum(n1,n2);
sub=sub(n1,n2);
printf("%d+%d=%d",n1,n2,add);
printf("%d-%d=%d",n1,n2,sub);
}

```

SET A

1. Write a function, which accepts an integer as parameter and returns 1 if number is even and 0 otherwise. Use this function in main to accept n numbers and check if they are even or odd.
2. Write a function, which accepts a character and integer n as parameter and display the next n characters.
3. Write a function disp_word() which accept single digit number and display it in words. If number is not in valid range then display "out of bounds".
4. Write a function odd(int n) that will return the value true or false depending on given number is odd or even. Using this function display given number is odd or even.
5. Write a function getsum(int n) that will return the sum of digits of a given number. Using this function display sum of digits of a given number.
6. Write a function ispalindrome(int n) that will return 1 or 0 depending on given number is palindrome or not. Use this function in main() to check n is palindrome or not.
7. Write a function getoctal(int n) that will return the octal equivalent of given integer n. Using this function display the octal equivalent of given number.

Signature of Instructor

Date:

SET B

1. Write a function prime() which accepts an integer as a parameter and returns 1 if the number is prime and 0 otherwise. Use this function in main to display first 10 prime numbers
2. Write a function swap(int,int) that will interchange the values of given 2 integers. Print the values before and after swap() call in main
3. Write a function Welcome(int) to which hours value is passed as parameter. The function should display message "Good Morning", "Good Afternoon", "Good Evening", "Good Night" depending on given hours value.
4. Write a function with prototype int factorial(int); to display factorial of a number.
5. Write a program using function to calculate x raise to y. Accept x & y from user. The function prototype is int power(int,int)

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 8

Recursive Functions

Recursion: Recursion is a process by which a function calls itself either directly or indirectly.

The points to be remembered for recursive functions:

1. There must be some terminating condition which will stop recursion.
2. Each time the function is called recursively it must be closer to solution
3. Expressions having recursive definition can be easily converted into recursive functions.

Sample Program: A program that calculates the following series using recursion.

$$n + (n-1) + (n-2) + \dots + 3 + 2 + 1$$

```
#include<stdio.h>
int add(int);
void main(void)
{
    int num,ans;
    printf("Enter any number:");
    scanf("%d",&num);
    ans=add(num);
    printf("Answer=%d",ans);
}

int add(int n)
{
    int result;
    if(n==1)
        return 1;
    result=add(n-1) + n;
    return result;
}
```

SET A

1. Write a program to calculate the factorial of an Integer (using recursion)
2. Write a program to calculate the GCD of two numbers using Recursion

GCD can be calculated as :

$$\begin{aligned} \text{gcd}(a,b) &= a && \text{if } b=0 \\ &= \text{gcd}(b,a \bmod b) && \text{otherwise} \end{aligned}$$

Signature of Instructor

Date:

SET B

1. Write a recursive functions to calculate sum of digits of a number. Use this function in main to accept a number and print sum of digits.
2. Write a recursive functions to calculate x^y (Don't use standard library function)
3. Write a recursive functions to print Fibonacci series of n numbers. Use this function in main to display first n Fibonacci numbers. The recursive definition for nth number is as follows:

$$\begin{aligned} \text{Fib}(n) &= 1 && \text{if } n=1 \text{ or } n=2 \\ &= \text{fib}(n-2)+\text{fib}(n-1) && \text{if } n>2 \end{aligned}$$

4. Write a recursive function to calculate the sum of digits of a number till you get a single digit number(Do not use a loop).(Example : 2345->2+3+4+5=14->1+4->5)
5. Write a recursive function to print the digits of a number in reverse order. Use this function in main to accept a number and print the digits in reverse order separated by tab

Example: 2345 output: 2 3 4 5

(Hint: recursive print(n)= print n if n is a single digit no
=print(n%10)+tab + recursive print(n/10))

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 9

One dimensional Array

You should read following topics before starting this exercise.

- Concept of array and how to declare an array?
- Entering and accessing array elements
- Passing array to functions

An array is a collection of data items of same data type by a common name. Each storage location is called an element of the array. Each element can be accessed by its index called as its subscript. It is surrounded by square brackets.

Actions involving array	Syntax	Example
Declaration of array	Datatype arrayName[size];	int marks[5];
Initialization of array	Datatype arrayName[]={ele1,ele2...elen} Datatype arrayName[size]={ele1,ele2...ele-size}	int marks[]={45,56,76,78,56}; int ct[3]={1,4,5}; so ct[0]=1, ct[1]=4, ct[2]=5
Accessing elements of array	The array index begins from 0. To access an array element, we need to refer to it as array_name[index]	Value=marks[3] This refers to 4 th element in array
Entering data into array		for(i=0;i<=4;i++) scanf("%d",&marks[i]);
Printing data from array		for(i=0;i<=4;i++) printf("%d",marks[i]);
Array and functions	We can pass an array to a function using 2 methods Pass an array element by element Pass entire array to function	/*Passing whole array*/ void print(int ct[3]) { for(i=0;i<=4;i++) printf("%d",ct[i]); }

Sample Program: Find largest element of an array

<pre>#include<stdio.h> int find_largest(int [],int); void main() { int arr[30],size,largest,i; printf("Enter the size of the array maximum upto30: "); scanf("%d",&size); printf("Enter the %d integer numbers: ",size); for(i=0;i<size;i++) scanf("%d",&arr[i]); largest = find_largest(arr,size); printf("\nThe largest element is: %d\n",largest); }</pre>	<pre>int find_largest(int arr1[],int size1) { int temp_larg,i; temp_larg=arr1[0]; for(i=1;i<size1;i++) { if(arr1[i]>temp_larg) temp_larg=arr1[i]; } return(temp_larg); }</pre>
--	--

SET A

1. Write a program to accept n numbers in range 1 to 25 and count number of occurrences of each number.
2. Write a program to accept n numbers and store all prime numbers in an array called prime. Display this array.
3. Write a function for linear search, which accepts an array of n elements and a key as a parameter and returns the position of key in the array and -1 if the key is not searched. Search it using functions and display proper message.
4. Write a function to accept an integer array and a number as a parameter. Count the occurrences of the number in array

Signature of Instructor

Date:

SET B

1. Write a program to accept n numbers and store them in an array in sorted order. Display the array. Write separate functions to accepts and display the array.
2. Write a function to sort an array of n integers using bubble sort method. Accept n numbers from user store them in array and sort them using this function. Display the sorted array.
3. Write a program to find the union and intersection of 2 sets of integers. Store them in 2 different arrays.
4. Write a program to accept a decimal number and convert it to binary, octal and hexadecimal. Write separate function.
5. Write a program to remove all duplicate elements from an array

Signature of Instructor

Date:

SET C

1. Write a program to read 5 integers in two arrays. Find the sum of these 2 arrays and store in third array and display it.
2. Write a program to read n integers in an array. Display the given array in reverse order.
3. Write a program to read n integers in an array. Display alternate elements of that array.
4. Write a program to initialize array of 10 integers to some numbers. Display array elements and find their sum and display it.
5. Write a function getsum(). The function should have 2 parameters as one is integer array and another is size of an array. The function should return sum of the numbers in given array

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Assignment 10

Multidimensional Array

You should read following topics before starting this exercise.

- Concept of 2-D array and how to declare 2-D array?
- Entering and accessing array elements

Actions involving 2-D Array	Syntax	Example
Declaration of 2-D array	Datatype arrayName[row][col];	int a[3][3];
Initialization of 2-D array	Datatype arrayName[][]={ { elements of row0},{ elements of row1 }, ...} Datatype arrayName[row][col]={ {elements of row0},{ elements of row1 }, ..., {elements of rowN} }	int a[3][2]={ { 1,2},{ 3,5},{ 6,9} }; int mat[3][2]={ 1,2,3,4,5,6};
Accessing elements of 2-D array	The array index begins from 0. To access an array element, we need to refer to it as array_name[index]	int a[3][2]; a is declared as a 2-D array having 3 rows (0,1,2)and 2 cols(0,1). So a[0][0] be the first element
Entering data into 2-D array		int a[3][4]; for(i=0;i<3;i++)//Outer loop(rows) for(j=0;j<=4;j++)//inner loop(col) scanf("%d",&a[i][j]);
Printing data from 2-D array		for(i=0;i<3;i++)//Outer loop(rows) { for(j=0;j<=4;j++)//inner loop(col) printf("%d\t",a[i][j]); printf("\n"); }

Sample Program: Program to calculate sum of all elements of matrix.

```
#include<stdio.h>
void display(int a[5][5],int m,int n);
void accept(int a[5][5],int m,int n);
void sum(int a[5][5],int m,int n);
void main()
{
    int ar[5][5],m,n;
    printf("Enter number of rows and columns\n");
    scanf("%d%d",&m,&n);
    printf("Enter Matrix elements\n");
    accept(ar,m,n);
    printf(" Matrix elements\n");
    display(ar,m,n);
}
```

```

        sum(ar,m,n);
    }
    void accept(int a[5][5],int m,int n) {
        int i,j;
        for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                scanf("%d",&a[i][j]);
    }
    void display(int a[5][5],int m,int n) {
        int i,j;
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
                printf("%d\t",a[i][j]);
            printf("\n");
        }
    }
    void sum(int a[5][5],int m,int n) {
        int i,j,s=0;
        for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                s=s+a[i][j];

        printf("Sum=%d\n",s);
    }
}

```

SET A

1. Write a program to add and multiply 2 matrices. Write separate functions to accept and display , add and multiply the matrices. Display addition and multiplication matrix.
2. Write a program to accept a matrix A of size m*n. Negate Numbers in matrix which are between 0 to 9.

Signature of Instructor

Date:

SET B

3. Write a program to accept a matrix A of size m*n and display its transpose. Write separate functions.
4. Write a program to accept a matrix A of size n*n and display sum of its diagonal elements and display it.
5. Write a program to accept a matrix A of size m*n and display an m+1 Xn+1 matrix such that the m+1th row contains sum of all elements of corresponding row and n+1th column contains the sum of elements of corresponding column.

Example:

A=

2	3
5	6

B=

2	3	5
5	6	11
7	9	16

Signature of Instructor

Date:

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []



Savitribai Phule Pune University

F.Y.B.C.A.(Science)

Semester-I

Section -II

BCA – 106

Lab Course –II

Fundamentals of Computers

TABLE OF CONTENTS

Sr.No.	Assignment
1	DOS Commands
2	Microsoft Word I
3	Microsoft Word II
4	Microsoft Excel I
5	Microsoft Excel II
6	Microsoft Powerpoint I
7	Microsoft Powerpoint II
8	Introduction to HTML
9	Working with Lists and Hyperlink
10	Working with Table
11	Working with Forms
12	Working with Frames

ASSIGNMENT COMPLETION SHEET

Lab Course II –Fundamentals of Computers

Sr.No.	Assignment	Marks (out of 5)	Teachers Sign
1	DOS Commands		
2	Microsoft Word I		
3	Microsoft Word II		
4	Microsoft Excel I		
5	Microsoft Excel II		
6	Microsoft Powerpoint I		
7	Microsoft Powerpoint II		
8	Introduction to HTML		
9	Working with Lists and Hyperlink		
10	Working with Table		
11	Working with Frames		
12	Working with Forms		
Total (Out of 60)			
Total (Out of 20)			
Internal Viva(10)			
Total (Assignment(20)+Internal Viva(10))Out of 30			

This is to certify that Mr./Ms. _____
has successfully completed the practical for Fundamentals of Computer Lab course II
and has scored ____ marks out of 30.

Lab In-Charges

H.O.D

Internal Examiner

External Examiner