GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

COURSE FILE

SUBJECT	**Programming for Problem Solving C**	**PROGRAMME**	**B. Tech.**
SUBJECT CODE	**BCS0IT1003**	**SEMESTER**	**I**
CREDITS	**3**	**DURATION OF SEMESTER**	**12 Week**
PREREQUISITE SUBJECTS	**--**	**SESSION DURATION**	**5 Horus /Week**

Prepared **Approved**

1. VISION AND MISSION OF GALGOTIAS UNIVERSITY

VISION

- To be known globally for value-based education, research, creativity and innovation

MISSION

- Establish state-of-the-art facilities for world class education and research.
- Collaborate with industry and society to align the curriculum.
- Involve in societal outreach programs to identify concerns and provide sustainable ethical solutions.
- Encourage life-long learning and team-based problem solving through an enabling environment.

2. VISION AND MISSION OF DEPARTMENT

VISION

- To be known globally as a premier department of computer science and engineering for value-based education, multi-disciplinary research and innovation.

MISSION

- Create a strong foundation on fundamentals of computer science and engineering through outcome-based teaching- learning process.
- Establish state-of-art facilities for analysis design and implementation to develop sustainable ethical solution.
- Conduct multi-disciplinary research for developing innovative solution
- Involve the students in group activity including that of professional bodies to develop leadership and communication skills.

3. PROGRAM EDUCATIONAL OBJECTIVES

PEO1	Graduates of Computer Science and Engineering will be globally competent and provide sustainable solutions for interdisciplinary problems as team players
PEO2	Graduates of Computer Science and Engineering will engage in professional activities with ethical practices in the field of Computer Science and Engineering to enhance their own stature to contribute towards society
PEO3	Graduates of Computer Science and Engineering will acquire specialized knowledge in emerging technologies for research, innovation and product development.

4. PROGRAMME OUTCOMES

PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOME (PSO)

PSO1	Able to analyze, design and implement sustainable and ethical solutions in the field of computer science.
PSO2	Able to use problem solving skills to develop efficient algorithmic solutions.

6. COURSE OUTCOMES (COs)

Course Outcomes (Cos): At the end of the course, the student will be able to			
CO No.	**Course Outcomes (Action verb should be in italics)**	**Bloom's taxonomy**	**Bloom's Level**
CO1	The student would learn the basic concepts of Computer and acquire various problem-solving techniques such as algorithms and flowchart.	Recall, Understand	**K1, K2, K3**
CO2	To understand the basic terminology used in programming and able to write, compile and debug programs in 'C' programming language and to develop program logics using decision structures and loop structures.	Recall Apply	**K1, K3**
CO3	To develop program logic using the concept of arrays and to understand the modular techniques such as functions.	Understand, Apply	**K2, K3**
CO4	Implement and develop small projects using the concept Structures and to understand and implement Pointers in C programming language.	Apply	**K3, K4**
CO5	Understand and apply basic file handling and string Operations	Apply	**K3**
CO6	Understanding of latest advances and its applications in Computer Programming and Problem Solving.	Apply, Analysing	**K4**

7. CO-PO-MAPPING

CO-PO-PSO Mapping (S/M/W indicates strength of correlation) S-Strong, M-Medium, L-Low														
CO's	Program Outcomes (Pos)													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2
CO1	2	2	1		2							2	2	
CO2	2	2	1		2							2	2	
CO3	2	3	2	3	2							1	1	
CO4	3	1	1	3	2								1	3
CO5	2	1	2	2	2								1	2
CO6	3	2	1	2	2							1	2	2

8. RELATIONSHIP BETWEEN THE COs AND PROGRAM OUTCOME Pos

S. No.	Course Outcomes	Mapped Program Outcomes	Mapping of Course Outcome with Assessment
1	The student would learn the basic concepts of Computer and acquire various problem-solving techniques such as algorithms and flowchart.	PO1, PO2, PO3, PO5, PO12, PSO1	Quiz-1, Quiz-2, VPL1, VPL2, VPL3
2	To understand the basic terminology used in programming and able to write, compile and debug programs in 'C' programming language and to develop program logics using decision structures and loop structures.	PO1, PO2, PO3, PO5, PO12, PSO1	Quiz-2, VPL1, VPL2, VPL3
3	To develop program logic using the concept of arrays and to understand the modular techniques such as functions.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1	Quiz-3, Quiz-4, VPL4, VPL5, VPL6
4	Implement and develop small projects using the concept Structures and to understand and implement Pointers in C programming language.	PO1, PO2, PO3, PO4, PO5, PSO2	Quiz-5, Quiz-6 VPL7,VPL8,VPL9,
5	Understand and apply basic file handling and string Operations	PO1, PO2, PO3, PO4, PO5, PSO2	Quiz-7, Quiz-8 VPL10, VPl1, VPL12
6	Understanding of latest advances and its applications in Computer Programming and Problem Solving.	PO1, PO2, PO3, PO4, PO5, P12	Research Article

9. LAB EVALUATION SCHEME

Mode of Evaluation:

S. No.	Evaluation Component	Duration	Marks (100) (%)	Nature of Component
1.	Quizzes	20 Min per quiz	25	On Line
2.	VPL	30 Min VPL	25	On Line
3	ETE	2 hours	50	Offline

10. Rubrics for final assessment

S. No.	Rubrics – Parts	Marks
1	Able to understand the problem statement	**5**
2	Able to convert the problem statement into program	**10**
3	VPL	35
Total		**50**

11. Assessment Format

GALGOTIAS UNIVERSITY

Department of Computer Science and Engineering

Internal Assessment

Subject Code : BCSE3073 Subject Name :Industry oriented Python Programming -III

Session : 2020-2021 **Class :**

Date : 18-02-2020 **Max. Marks :**

GALGOTIAS UNIVERSITY

Department of Computer Science and Engineering

Final Assessment

Subject Code : **Subject Name:**

Session : **Class :**

Date : **Max. Marks :**

12. External Lab Assessment (End Semester)

S. No.	Enrollment. No.	Name of the Student	Internal Assessment Test (50)	External Assessment Test (50)	Total (100)	Marks (in words)
1.						
2.						
3.						
4.						
5.						

13. Syllabus

Syllabus Problem for Problem Solving C				
Name of The Course	Programming for Problem Solving C			
Course Code	**BCSE0IT1003**			
Prerequisite	NA			
Corequisite	NA			
Antirequisite	NA			
	L	**T**	**P**	**C**
	1	0	4	3
Course Objectives:				
• **Provide an overview of computers and problem-solving methods using 'C' language**				
• **Serve as a foundation for the study of programming languages.**				
• **Learn to develop program using 'C' language.**				
• **To develop the software using the concept of 'C' Language.**				
CO1	The student would learn the basic concepts of Computer and acquire various problem solving techniques such as algorithms and flowchart.			
CO2	To understand the basic terminology used in programming and able to write, compile and debug programs in 'C' programming language and to develop program logics using decision structures and loop structures.			
CO3	To develop program logic using the concept of arrays and to understand the modular techniques such as functions.			
CO4	Implement and develop small projects using the concept Structures and to understand and implement Pointers in C programming language.			
CO5	Understand and apply basic file handling and string Operations			

CO6	Understanding of latest advances and its applications in Computer Programming and Problem Solving.

Text Book (s)

- **Alexis Leon and Mathews Leon (2001), Introduction to Information Technology, Tata McGraw-Hill.**
- **R.G. Dromey (2001), How to Solve it by Computer, Prentice Hall of India.**
- **Al Kelley and Ira Pohl (1998), A Book on C Programming in C, 4th Edition, Pearson Education.**

Reference Book (s)

- **E. Balagurusamy7th Edition, Programming ANSI C, McGraw-Hill**
- **Brian W. Kernighan and Dennis M. Ritchie, The C programming Language, Prentice-Hall in 1988**
- **Byron Gottfried, Programming with C, Schaum's Outline**

Unit-1 Introduction to Computers and Algorithms **6 hours**

Parts of a computer, Overview of operating systems, assembler, compilers, interpreters and programming languages,

Flowchart: Elements, Identifying and understanding input/ output, Branching and iteration in flowchart,

Algorithm design: Problem solving approach (top down/bottom up approach), Pseudo Code: Representation of different construct, writing pseudo-code from algorithm and flowchart

Unit-2 Constructs of C **8 hours**

Introduction to C programming language, Data types, Variables, Constants, Identifiers and keywords, Storage classes, Operators and expressions, Types of Statements: Assignment, Control, jumping, Control statements: Decisions(if-else), Loops (while, for, do while), break, continue, case control structure, go to, exit statement

Unit-3 Arrays and Functions **8 hours**

Array handling in C – declaration – single dimensional arrays, two – dimensional arrays, multi-dimensional arrays, sorting and searching on single- and two-dimensional arrays.
Function– declaration - arguments (formal and actual) – return types – types of functions difference between built-in and user-defined functions, Call by Value and call by reference.

Unit-4 Structures, Union and Pointers	**8 hours**
Structure Introduction, Declaration, Difference, Application, Nested structure, self-referential structure, Array of structures, Passing structure in function, unions- difference between structure and union. Pointer: Introduction, declaration of pointer variables, Operations on pointers: Pointer arithmetic, Arrays and pointers, Dynamic memory allocation, passing pointer variables into function.	
Unit-5 String and File Handling	**8 hours**
String: Introduction, predefined string functions, Manipulation of text data, Command Line Arguments. Files: Introduction, concept of record, I/O Streaming and Buffering, Types of Files: Indexed file, sequential file and random file, Creating a data file, Opening and closing a data file, Various I/O operations on data files: Storing data or records in file, adding records, Retrieving, and updating Sequential file/random file.	
Unit-6 Advances in C Programming	**7 hours**
The advances and the latest trends in the course as well as the latest applications of the areas covered in the course. The latest research conducted in the areas covered in the course. Discussion of some latest papers published in IEEE transactions and ACM transactions, Web of Science and SCOPUS indexed journals as well as high impact factor conferences as well as symposiums. Discussion on some of the latest products available in the market based on the areas covered in the course and patents filed in the areas covered in the course.	

14. Salient Feature of C

Some of C language characteristics that define the language and also have lead to its popularity as a programming language.

- Small size.
- Extensive use of function calls.
- Structured language.
- Low level (Bitwise) programming readily available.
- Pointer implementation - extensive use of pointers for memory, array, structures and functions.
- It has high-level constructs.
- It can handle low-level activities.
- It produces efficient programs.
- It can be compiled on a variety of computers

Objective(s):

- To be familiar with syntax and structure of C-programming.
- To learn problem solving techniques using C

Title

Write a Program to calculate and display the volume of a CUBE having its height (h=10cm), width (w=12cm) and depth (8cm).

Problem Analysis:

The problem is to calculate the volume of a CUBE having its inputs parameters identified as: Height (integer type), width (integer type) and depth (integer type). The output of the program is to display the volume; hence the output parameter is identified as vol (integer type). During the processing or calculation phase, we don't need any extra parameters (variables) for this problem.

The volume of the cube is the multiplication of its height, width and depth, hence the mathematical formula to calculate volume is:

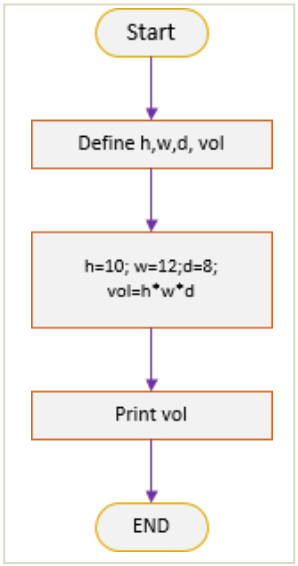
vol = height* width* depth. (vol = h*w*d)

Input variables	Processing variables/calculations	Output variables	Necessary header files/functions/macros
h(int) w(int) d(int)	vol = h*w*d	vol (int)	stdio.h

Algorithm:

1. Start
2. Define variables: h(int), w(int), d(int), vol(int)
3. Assign value to variables: h = 10, w=12, d=8
4. Calculate the volume as: vol = h*w*d
5. Display the volume (vol)
6. Stop

Flowchart:



Code:

```
//Following code is written and compiled in
Code::Blocks        IDE
#include<stdio.h> int
main(void)
{
//start the program
inth,w,d,vol;
//variables declaration
h=10;w=12;d=8;
//assign value to variables
vol=h*w*d;
//calculation using mathematical formula
printf("The Volume of the cube is:
%d",vol);
//display the volume return
0;
//end the main program
}
```

Output (Compilation, Debugging & Testing)

The Volume of the cube is: 960

Discussion and Conclusion

This is the first code written in C program. The program is focused on the calculation of volume of a cube for the given height, width and depth. From this lab, I understood the basic structure of C programming including the meaning of header files & steps of problem solving. Hence, volume of a cube is calculated and displayed.

LAB SHEET #1

Lab exercises (please code yourself and show the output to instructor):

1. Write a program to display “hello world” in C.
2. Write a program to add two numbers (5&7) and display its sum.
3. Write a program to multiply two numbers (10&8) and display its product.
4. Write a program to calculate area of a circle having its radius (r=5).
5. Write a program to calculate area of an ellipse having its axes (minor=4cm, major=6cm).
6. Write a program to calculate simple interest for a given P=4000, T=2, R=5.5. (I = P*T*R/100)

Objective(s):

To be familiar with different data types, Operators and Expressions in C.

Title:

Write a program to take input of name, rollno and marks obtained by a student in 5 subjects each have its 100 full marks and display the name, rollno with percentage score secured.

Problem Analysis:

Based on the problem, it is required to get the input of name, roll number and marks in 5 subjects of a student. The program should display the name; roll number and percentage of marks secured by that student as output. The input variables shall be: name, rollno, msub1, msub2, msub3, msub4, msub5. We need to calculate percentage of marks obtained. So the variable 'score' holds the percentage to be displayed.

$$\text{Percentage of marks obtained} = \frac{\text{total marks on 5 subjects}}{\text{total full marks}} \times 100$$

Hence, msum = msub1 + msub2 + msub3 + msub4 + msub5;

$$\text{Score} = \frac{\text{msum}}{500} \times 100$$

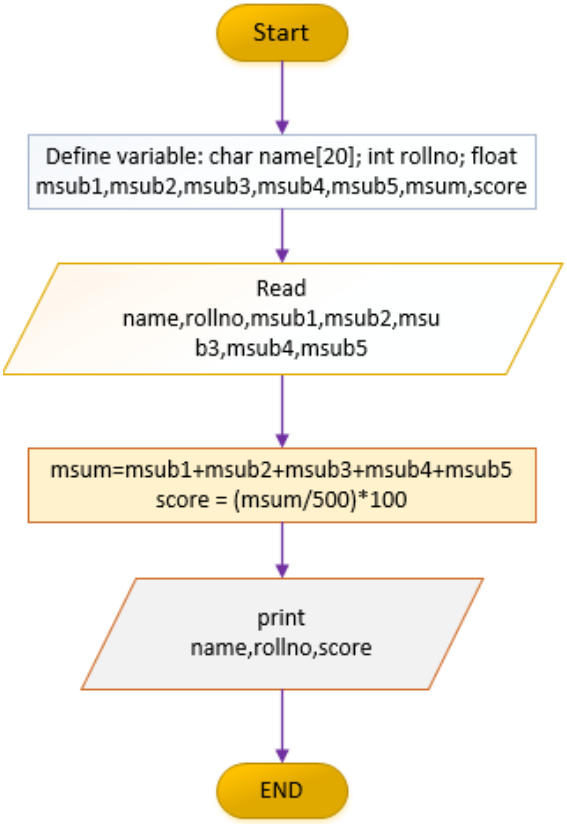
Input variables	Processing variables/calculations	Output variables	Necessary header files/functions/macros
Name (char type) rollno (int) msub1, msub2, msub3, msub4, msub5 (float)	msum (float)	name (char type) rollno (int) score(float)	stdio.h conio.h scanf() &printf() for formatted i/o.

Algorithm:

1. Start
2. Define variables: name, rollno, msub1, msub2, msub3, msub4, msub5, msum, score
3. Take input from keyboard for all the input variables
4. Calculate the sum of marks of 5 subjects and also calculate the percentage score as: Msum = msub1 + msub2 + msub3 + msub4 + msub5; Score = $\frac{\text{msum}}{500} \times 100$
5. Display the name, roll number and percentage score.

6. Stop

Flowchart:



Code:

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

int main(void)
{
char name[20];
introllno;
float msub1, msub2, msub3, msub4, msub5, msum, score;

printf("Enter Name of Student: ");
scanf("%[^\n]", name); /*can use scanf("%s",name) but it reads single word only.*/
printf ("\nRoll Number: "); scanf("%d",
&rollno);
printf ("\nEnter Marks in 5 Subjects:\n"); scanf("%f%f%f%f%f", &msub1,
&msub2, &msub3, &msub4, &msub5);

msum=msub1+msub2+msub3+msub4+msub5; score
= msum/500*100;
```

```
printf("\nName of Student: %s", name);

printf("\nRoll Number: %d", rollno);
printf ("\nPercentage Score Secured: %2.2f%c", score,'%');

return 0;
}
```

Output (Compilation, Debugging & Testing):

```
Enter Name of Student: Shree HariKoirala
Roll Number: 522
Enter Marks in 5 Subjects:
45.5
50
63
76
62.5
Name of Student: Shree HariKoirala
Roll Number: 522
Percentage Score Secured: 59.40%
```

Discussion & Conclusion:

In this second lab of C Programming, based on the focused objective(s) to understand about C data types with formatted input/output functions, the additional lab exercises made me more confident towards the fulfillment of the objectives.

LAB SHEET #2

Lab exercises (please code yourself and show the output to instructor):

1. Write a program to declare two integer and one float variables then initialize them to 10, 15, and 12.6. Also print the variable values in the screen.
2. Write a C program to prompt the user to input 3 integer values and print these values in forward and reversed order.
3. Write a program to calculate simple and compound interest.
4. Write a program to swap two variables' values with and without using third variables
5. Write a program to check odd or even number (a) using modulus operator (b) using bitwise operator (c) without using bitwise and modulus operator (d) using conditional operator.
6. Print the value of y for given x=2 & z=4 and analyze the output.
   a. y = x+++ ++x; b. y= ++x + ++x; c. y= ++x + ++x + ++x;
   d. y = x>z; e. y=x>z? x:z; f. y = x&z;
   g. y= x>>2 + z<<1;
7. Write a program to print the size of char, float, double and long double data types in C.

LAB SHEET #3

Objective(s): To be familiar with formatted and unformatted I/O in C with preprocessor directives

Lab Exercises (Please Code yourself and show the output to instructor):

1. Write a program to produce the output as shown below:

x	y	expressions	results
6	3	x=y+3	x=6
6	3	x=y-2	x=1
6	3	x=y*5	x=15
6	3	x=x/y	x=2
6	3	x=x%y	x=0

2. Given x=3.0, y=12.5, z= 523.3, A=300.0, B=1200.5, C=5300.3, Write a program to display the following:

```
X   y   z=      3.0|    12.5|    523.3|
A   B   C    300.0|  1200.5|   5300.3|
            =
-------------------------------------------------------
X   y   z=    |3.00       |12.50    |523.30
A   B   C=    |300.00      |1200.50  |52300.30
```

3. Given the three numbers a(=8), b(=4),c and constant value PI=3.1415, calculate and display the following result using macros (preprocessor directives)
   a. c = PI * mult(a,b) //the macro mult(a,b) perform the multiplication of a & b(a*b)
   b. c= PI* sum(a,b) //the macro mult(a,b) perform the sum of a & b (a+b)
   c. c= PI *sub(a,b) //the macro mult(a,b) perform the subtraction of a & b (a-b)
   d. c= PI*div(a,b) //the macro mult(a,b) perform the division of a & b (a/b)
4. Demonstrate the differences among getch(), getche(), getchar(). Demonstrate the difference between scanf() & gets(), printf() & puts().
5. Write a program to take a character input from keyboard and check if it is a number or alphabet or special character using ASCII CODE Again check if the character is using character functions below:
   a. Alphanumeric => isalnum()
   b. Blank character => isblank()
   c. Alphabetic => isalpha()
   d. Control character => iscntrl()
   e. Number-digit => isdigit()
   f. Upper case => isupper()
   g. Lower case => islower()

h. Hexadecimal digit => ixdigit()
i. Graphical character => isgraph()

LAB SHEET #4

Objective(s): To understand the programming knowledge using Decision Statements (if, if-else, if- else if ladder, switch and GOTO)

Lab Exercises (Please Code yourself and show the output to instructor):

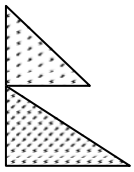
1. Write a program to find the largest and smallest among three entered numbers and also display whether the identified largest/smallest number is even or odd.
2. Write a program to check whether input alphabet is vowel or not using if-else and switch statement.
3. Write a program to get input of two or higher digit integer number and display in reverse order.
4. Write a program that asks a number and test the number whether it is multiple of 5 or not, divisible by 7 but not by eleven.
5. Write a program to check whether the entered year is leap year or not (a year is leap if it is divisible by 4 and divisible by 100 or 400.)
6. Write a program to read the values of coefficients a, b and c of a quadratic equation $ax^2+bx+c=0$ and find roots of the equation.

LAB SHEET #5

Objective(s): To understand the programming using Loop & nested loop Statements (for, while, do-while)

Lab Exercises (Please Code yourself and show the output to instructor):

1. Write a program to input two integer numbers and display the sum of even numbers between these two input numbers.
2. Write a program to find GCD (greates common divisor or HCF) and LCM (least common multiple) of two numbers.
3. Write a program to display Fibonacci series of last term up to 300.
4. Write a program to display the flag of Nepal using symbolic/HEX character in C.



5. Write a program to display the following.

a.

b.

```
1
1   4
1   4   9
1   4   9   16
1   4   9   16   25
```

c.

d.

```
1   6   10   13   15
2   7   11   14
3   8   12
4   9
5
```

LAB SHEET #6

Objective(s): To understand function programming, its types and function-call

Lab Exercises (Please Code yourself and show the output to instructor):

1. Write a program to add, subtract, multiply and divide two integers using user defined type function with return type.
2. Write a program to calculate sum of first 50 natural numbers using recursive function.
3. Define a function named fact() to calculate factorial of a number n and then write a program that uses this function fact() to calculate combination and permutation.
4. Write a recursive function to generate Fibonacci series.
5. Write a program that illustrates use of local, global and static variables

LAB SHEET #7

Objective(s): To understand programming using different dimensions of Array.

Lab Exercises (Please Code yourself and show the output to instructor):

1. Write a program to enter 10 floating numbers in an array and display it.
2. Write a program to display largest and smallest element of an array defined in Q.No. 1.
3. Write a program to initialize one dimensional array of size 8 and display the sum and average of array elements
4. Write a program to read two matrices of order 3 * 2, add them and display the resultant matrix in matrix form.
5. Write a program to multiply two 3*3 matrix.
6. Write a program to read a string and check for palindrome without using string related function (a string is palindrome if its half is mirror by itself eg: abcdcba).

LAB SHEET #8

Objective(s): To understand programming with Pointer, String and Function call by reference.

Lab Exercises (Please Code yourself and show the output to instructor):

1. Write a program to find biggest among three numbers using pointer.
2. Write a program to find the sum of all the elements of an array using pointers.
3. Write a program to swap value of two variables using pointer.
4. Write a program to read a sentence and count the number of characters &words in that sentence.
5. Write a program to read a sentence & delete all the white spaces. Replace all "." by ":".
6. Write a program to copy one string to another string with and without using string handling function.
7. Write a program to concatenate two strings.
8. Write a program to compare two strings.
9. Write a program to sort 5 string words stored in an array of pointers.
10. Write a program to print the following pattern

```
U N
U N I V
U N I V E R
U N I V E R S I
U N I V E R S I T Y
U N I V E R S I
U N I V E R
U N I V
U N
```

LAB SHEET #9

Objective(s): To understand programming with Structure

Lab Exercises (Please Code yourself and show the output to instructor):

1. Create a structure named company which has name, address, phone and noOfEmployee as member variables. Read name of company, its address, phone and noOfEmployee. Finally display these members' value.
2. Write a program to enter to Cartesian coordinate points and display the distance between them.
3. Write a function which accepts structure as argument and returns structure to the calling program.
4. Pass the structures defined in Question 1 into a function and read the structure member and display the values from the function (use structure pointer).
5. Define a structure "complex" (typedef) to read two complex numbers and perform addition, subtraction of these two complex numbers and display the result.
6. Write a program to read RollNo, Name, Address, Age & average-marks of 12 students in the BCT class and display the details from function.
7. Write a program to show programming examples with union and enumerations.

LAB SHEET #10

Objective(s): To understand data files and file handling in C.

Lab Exercises (Please Code yourself and show the output to instructor):

1. Write characters into a file “filec.txt”. The set of characters are read form the keyboard until an enterkey is pressed (use putc() and getc() function).
2. Read characters form file “filec.txt” created in question 1. Also count the number of characters in the file (use fputs() and fgets() function).
3. Write set of strings each of length 40 into a file “stringc.txt” and display it (use fputs() and fgets() function).
4. Write name, age and height of a person into a data file “person.txt” and read it (use fprintf() and fscanf() function)
5. Write a program to replace DOS command “type” by “watch”. The “watch” command is to be created by C program “watch.c” and read the file “filec.txt” written in question no 1. (In DOS, we use the command like #type filec.txt which is to be replaced like #watch filec.txt)