

Programming in C Laboratory Manual

Table of Contents

1. Getting started with C.....	6
2. Input Output.....	9
3. Operator.....	10
4. Conditional statements.....	12
5. Control Statements	14
6. Function.....	16
7. Array & Strings.....	17
8. Pointer,Storage class and dynamic memory allocation.....	19
9. Structure , Union and Enums.....	20
10. FILE, I/O and Command Line Argument	21

Rules:

1. The above mentioned 10 laboratories must be completed in a semester.
2. Each laboratory must be completed in two labs (One practice lab followed by an evaluation lab). In practice lab the lab assistants, faculty will assist the student whereas in evaluation lab no assistance from any of the staffs will be entertained.
3. Each laboratory (Excluding Laboratory-0) carries 100 marks.
 - a. Attending practice lab.....20 Marks
 - b. Record.....20 Marks
 - c. Evaluation Lab.....30 Marks
 - d. Lab Test.....30 Marks
 - i. Programming Ability Test.....20 Marks
 - ii. Viva/Quiz Test.....10 Marks

}

Continuous Evaluation

}

One-time Evaluation at the end of 10 laboratories
4. All practice lab and evaluation lab questions will be asked from the lab manual.
5. Absenteeism in any lab means corresponding marks will be deducted.
6. Discipline has the highest priority.

Laboratory 0

Basic Linux Commands

Login to your system using the user name and password given to you. Open Terminal.
Press ESC key to go to the ESCAPE mode else you are working in the command mode.
Check outputs of all commands listed in the following.

Sl.No	Command	Explanation	Syntax	Option
1	ls	Shows short listing of files presents in a directory.	ls [option] [file]	- l - long listing - d - directory - a - for hidden files
2	mkdir	Creates a directory	mkdir [name]	
3	cd	change to a named directory	cd [dir name]	cd . - stays at current directory cd .. - go to parent directory cd - changes to home directory
4	pwd	Display the path of the current directory	Pwd	
5	vi	Create/Edit a file	vi < filename > or vim	After editing go to escape mode and press :w - to save the content , :q - to quit w q - to quit after saving the content
6	cp	Copies the content of one file to another file.	cp <sourceFileName> <destinationFileName>	- i - interactively , -r - recursively for directory etc.
7	mv	Move or rename one file to another	mv <oldname> <newname>	
8	rm	Remove a file from the current directory	rm <filename>	- l interactively, - f forcefully , r - recursively
9	rmdir	Removes a directory	rmdir<dirname>	
10	cat	Displays the content of a file	cat <filename >	
11	grep	Searches a file for a keyword	grep “ keyword” <file>	
12	wc	counts no of word/characters/ line in a file	wc < filename >	- w - for word , - l for line
13	echo	Echos a particular string or content of a shell variable	echo Hello World will display Hello World in the prompt	
14	>	Redirect standard output to a file	<command> <filename>	e.g. echo Hello World > myfile stores the output of the command in the left hand side to a file named myfile

15	date	Displays the current date	Date	Options d - the day of the month , y the last two digits of the year , D date in mm /dd /yy format , H , M ,S hour , minute and second respectively
16	cal	Displays the calendar for a specific month and year.	cal , cal april 2009	Cal [[month] year]
17	who	Tells the users currently logged into the system	Who	
18	man	Displays documentation of a command	man < command >	
19	tac	Used to display the content of file in reverse order	tac <filename>	
20	More,less	These command are used to see content of a large file , one page at a time.	more < filename > less <filename >	less is the standard pager for linux and in general more powerful than more.
21	cmp	Is used to see whether two files are identical or not	cmp < file1 >< file 2>	The two files are compared byte by byte and the location of the first mismatch is printed on the screen If two files are identical, then it does not print anything on the screen .
22	comm	displays what is common between both the files	comm < file1 ><file2>	
23	head	displays the top of the file, when used without any option it will display first 10 lines of the file	head - n [numberofline] <filename>	When - n option is used followed by a number, it displays that many numbers from the top of the file.
24	tail	displays the end of the file. By default it will display last 10 lines of the file.	tail - n [numberof lines] < filename >	When - n option is used followed by a number, it displays that many numbers from the bottom of the file.
25	<	Redirect standard input. from a file < filename		
26	Clear	Clears the terminal		
27	chmod	Changes the permission of a file	chmod < mode > filename	
28	ssh	ssh option host	Makes a secure shell connection for remote access SUIT.	
29	whoami	Returns current username.		Needs to see who logged into the system
30	finger	Searches for some user.	finger xyz	
31	sort	Sorts the content of a files	sort <filename>	

Do the following tasks after logging into your system. You are directed to note down the outputs in your rough lab notebook.

1. type pwd to see the current directory .
2. Check the current username.
3. type mkdir <your name> to create a directory of your own
4. cd <your name> to go to your own directory
5. type again pwd to see the current directory
6. create a file named test and write 10 sentences about your first day at SUIIT
7. save the file and quit to prompt
8. type grep first test to see how many times the word first occurs in the file test
9. using wc find out total no of characters, no of lines and no of words in the file test
10. type cat test to see the content of the file test
11. type cp test testl
12. cmp test testl
13. see the content of the file testl using cat
14. type ls and ls-a to see the listing of files
15. type rm testl
16. type ls and ls-a and see the changes in the file listing in step 12 and 14.
17. Store the output of ls, and ls -a into files ls.txt, lsa.txt
18. Sort the above two files using sort.
19. type echo Hello How are you ?
20. type echo Hello World > testl
21. cat testl to see the content of testl file
22. See the content of testl in reverse order.
23. type echo sunshine > test2
24. cat test2 to see the content of the file test2
25. cat testl test2 > test3
26. cat test3 to see the content of test3
27. cp test3 test4
28. cat test4 to see the content of file test4
29. gedit test3 and see the file test3. Add a third line “ First year students” to the file, save and quit.
30. cat test3 to see the content of the file test3 again and see the changes made.
31. cmp test3 test4 to see the first mismatch among these two files.
32. Check the common content between test3 and test4
33. type cp test3 test4
34. count the no of lines, word and characters in both the files test3 and test4 and see whether they are same or not.
35. type ls , ls -a and check the difference in the result.
36. type rm -f test4
37. Display the calendar of the current month.
38. type echo * and see the result. Explain the output.
39. type mv test3 test4

Laboratory-1

Getting started with C

In Linux environment, **cc** and **gcc** are two well known C compilers.

All C programs (Source code) are stored in a file ending with extension **.c** (e.g. **MyProgram.c**)

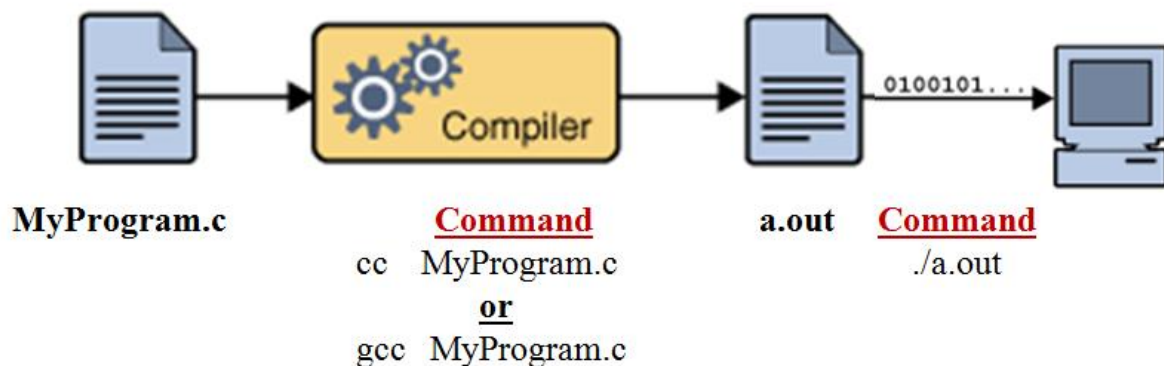
After writing the source code in C the files has to be compiled. To compile a source file **test.c**, use the command **cc MyProgram.c** (or **gcc MyProgram.c**).

If there is no problem in the source code, it generates the executable **a.out** (default name) and then you may run the same by typing **a.out** or **./a.out** (depending upon the path).

If you want to give a different name of the executable you need to use **cc (or gcc) MyProgram -o test** . This generates executable file **test** that you can run.

If you don't want to create executable, rather you want to create only object file (.o file) you may use the command **cc -c test** that creates **test.o**.

For inclusion of certain types of header libraries we need to give options while compiling e.g. **cc myprog.c -lm** (when **math.h** is included)



Type the following sample programs, save it giving a file name, compile and run it. See the result and match with the expected outputs.

Do the following tasks after logging to your system. You are directed to note down the outputs in your rough lab notebook.

Program 1

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

```
main()
```

```
{
```

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

```
    return 0;
```

```
}
```

This **header file** contains declarations of widely used input and output

Every C program execution begins from **main()**

printf()→is one widely used function that prints a string passed or converts a value type into a string and prints in the standard output device. **\n**→newline character.

return is used to return certain value to the caller. In this case returns 0 helps returning the control to the operating system.

Program 2: Write a program to print the University Name and Course in two separate lines

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

```
    printf(" Sambalpur University Institute of Information Technology \n" );
    printf("Bachelor of Technology");
    return 0;
```

```
}
```

Sample output:

Sambalpur University Institute of Information Technology
Bachelor of Technology

Program 3: Write a program to add, subtract and multiply two numbers

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

```
    int no1=5, no2=10;
    int mul;
    printf("Addition=%d",no1+no2 );
    printf("\n Subtraction=%d",no1 - no2);
    mul=no1 * no2;
    printf("Multiplication=%d", mul);
    return 0;
```

```
}
```

Sample output:

Addition=15
Subtraction=-5 Multiplication=50

Program 4: Write a program to swap two numbers

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

```
    int no1, no2,temp;
    printf("Enter two numbers:");
    scanf("%d%d",&no1,&no2);
    temp=no1;
    no1=no2;
    no2=temp;
    printf("After Swapping no1=%d no2=%d", no1,no2);
    return 0;
```

```
}
```

Sample output:

Enter two numbers: 10 20
After Swapping no1=20 no2=10

Program 5: Write a program to convert temperature in Celsius to Fahrenheit

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

```
    int c;    float f;
    printf("Enter Celsius temp:");
    scanf("%d", &c);
    f = (9.0*c)/5.0 + 32.0;
    printf("Converted Temp = %f F\n",f);
    return 0;
```

```
}
```

Sample output:

Enter Celsius temp: 100
Converted Temp = 212 F

6. Write a program to scan an integer and prints its value in octal, hexadecimal, decimal and default specification (which is again decimal value).

```
#include<stdio.h>
main()
{
    int var;
    printf("Enter a number:");
    scanf("%d",&var);
    printf("Octal value=%o\n",var);
    printf("Hexadecimal value = %x\n",var);
    printf("Decimal value = %d\n",var);
    printf("Integer value=%i\n",var);
    return 0;
}
```

Sample output:

Enter a number: 100
Octal value=144
Hexadecimal value = 64
Decimal value = 100
Integer value= 100

Note:

%o→octal value
%x→hexadecimal value
%d→decimal value
%i→integer value

7. Write a program to find the distance between two 2-D points

```
#include<stdio.h>
#include<math.h>
main()
{
    float x1,y1,x2,y2,d;
    printf("Enter a Point:");
    scanf("%f%f", &x1, &y1);
    printf("Enter a 2nd Point:");
    scanf("%f%f", &x2, &y2);
    d=sqrt(pow(x1-x2, 2) + (y1-y2)*(y1-y2))
    printf("Distance between two points=%f", d);
    return 0;
}
```

Note:

1. While compilation use
cc FileName.c -lm
2. sqrt(X) function finds
square root of X.
3. pow(m,n) finds m^n

**8. Write a program to calculate volume and surface area of a sphere.[area = $4 * PI * R * R$,
vol= area * R / 3]**

9. Write a program to print the integer part and fractional part for the given number.

10. Write a program that accepts length in inches (integer value) and convert them into feet and yards.

Laboratory-2

Input Output

1. Write a program that will accept student's marks in five different subjects through keyboard and that will calculate total marks and percentage marks obtain by the student. Assume that the maximum mark of each subject is 100.
2. Write a program that accepts a floating point number and round it off to its nearest integral value.
3. Write a program that takes a double number and find its ceiling and floor. [e.g. $\text{ceil}(2.4)=3$, $\text{floor}(2.4)=2$]
4. Given an input (x), write a program to print it's reciprocal value ($1/x$), negated value ($-x$) and square (x^2) of it.
5. Write a program to accept the distance in KM through the keyboard. and convert this distance in meters and feet.
6. Write a Program to calculate effective relativistic mass by taking rest mass & velocity, [effective mass= $\text{restmass}/(\text{sqrt}(1.0-(v*v)/(C*C)))$], where v =velocity, C =speed of light, i.e $3.1E8$]
7. Write a program that compute the compound interest of given P, r, t, n as principle amount, interest rate, time period & compounding per year.
8. Program to calculate the salary of sales man where quantity and price are given from user and fixed basic of 15000, bonus of 200 and commission as 0.02 on both quantity and selling price.
9. Write a program that accept a character from user, and print its next and previous character.
10. Write a small calculator program that does addition, subtraction, multiplication, division and modulo operation of two input numbers.
11. Write a program to calculate range, height archived by a projectile where initial velocity and angle of projection will be user input.
12. Write a program to evaluate the fuel consumption of a car. The mileage at the start and end of the journey should be read, and also the fuel level in the tank at the start and end of the journey. Calculate fuel used, miles travelled, and hence the overall fuel consumption in miles travelled per Ltr. of fuel.
13. A, B and C can do a piece of work in $N1$, $N2$ and $N3$ days respectively. All three can finish the work in $N4$ days. WAP that will take $N1$, $N2$, $N3$ and compute and display $N4$.
14. Write a program to display week day (like Monday, Tuesday etc) when day of the month is given as input. Assuming day for first date of the month is known.
15. Write a program to find all roots of a quadratic equation.
16. Two trains are running opposite to each other with speed $V1$ and $V2$ Kmph respectively and distance between both the train is D . Write a program that will accept $V1$, $V2$ & D and calculate the time required by trains to meet each other.
17. Interest Calculation: Your C-program will accept Rate of Interest (RI in Percentage), Principal Amount (PA), Years (Y) from keyboard, and calculate the, Simple Interest (SI) and Compound Interest (CI). Finally, it will display SI, CI after Y-years.
$$SI=PA*RI*Y \quad CI=PA*(1+RI)^Y$$

Laboratory-3

Operator

1. Write a program to swap two numbers using bitwise operator.
2. Write a program to find the size of each primitive data types in your compiler. Also see the range of each data type defined in limits.h and float.h
3. Write a program to store a value greater than the allowed range for each data type and check the value actually stored.
4. Write a program to check the Divide by zero exception, using floating point operand for modulus (%)operator. By writing a constant on the left side (LVALUE) of an assignment operator.
5. Write a program to multiply two numbers, where the second number is power of two, using shift operator.
6. Write a program that converts an amount (in integer, upto 999999) into words.[i/p-431104, output-4 lakh 31 thousand 1 hundred and four only]
7. Write a program that takes one positive integer and shifts the whole bit pattern 4 bits to the left. Check that this operation is same as multiplying the no by 16 if no overflow occurs.
8. Write a program to show the logical value of the following logical expressions for values a=15 and b=20.
 - a. `Z1 = (a == b) || (a <= 15) && (b >= a)`
 - b. `Z2 = !(a) || (b-a)`
 - c. `Z3 = 0 || 1 && 5`
9. Given a number in seconds, Represent it in hours, minutes and seconds in the format: XX Hours : YY Minutes : ZZ Seconds
10. `int x=0, y= 10, w=20, z, T= 1 , F=0;`for the above variables Evaluate the value of Z after each statement
 - a. `z = x == y == z`
 - b. `z = x > y > z`
 - c. `z = x < y < z`
 - d. `z = (x == 5 && y < 15);`
 - e. `z = (x == 0 && y > 5 && w == 10)`
 - f. `z = (x == 0 || y > 5 && w == 20);`
 - g. `z = (F || ++x || w - 20 || x);`
11. Write program to do the following bitwise operation on the following data A = 0x 67A4, B = 0x7
 - a. `A & B`
 - b. `~ A`
 - c. `A || B`
 - d. `~0`
12. Given a number find out it' s 1 ' s complement and 2' s complement, (represent the numbers in hexadecimal format)
13. Write a program to check the value of the following expressions. Given a = 10, b = 20
 - a. `X = a++ + ++ a + ++b - a + ++a;`

- b. $Y = (a++, a+=5, a-1)$
 - c. $Z = a > b ? a++ : a-- ? b+1 : b+5$
14. Do the left and right shift of the following values and check the result, char c1 = 50, c2 = 122; Unsigned char c3 = 122;
- a. $2250 >> 2$
 - b. $-50 << 3$
 - c. $c1 >> 3$
 - d. $c2 << 2$
 - e. $c3 << 4$
15. Write a program to set desired bits in a given number. The data in which bits will be set and the bits need to be set will be given through user input.
- Example Number: 10110101
Bits to be set: 00110110
Output: 10110111
16. Write a program to toggle desired bits in a given number. The data in which bits will be toggled and the bits need to be toggled will be given through user input.
- Example Number: 10110101
Bits to be set: 00110110
Output: 10000011
17. Write a program to clear desired bits in a given number. The data in which bits will be cleared and the bits need to be cleared will be given through user input.
- Example Number: 10110101
Bits to be cleared: 00110110
Output: 10000001

Laboratory-4

Conditional statements

1. Write a program to check a given number is odd or even.
2. Write a program to test whether a character is in lowercase, upper case, digit or special character.
3. Write a program to print maximum and minimum of from three given numbers.
4. Write a program to read weekday number and print weekday name using switch case.
5. Write a program will read two integer numbers and perform basic operations like +, -, *, / and % using switch case statement in C language.
6. Write a program to find that entered year is leap year or not.
7. Write a program to find whether an input integer is divisible by both 2 and 3, divisible by 2 but not by 3, divisible by 3 but not by 2 or not divisible by 2 and 3.
8. WAP that will accept two numbers say X and Y and perform followings –
 - a. Add both the number if X and Y have same value
 - b. Subtract X from Y if $X > Y$
 - c. Otherwise Subtract Y from X
9. Write a program that accept student roll no and average mark from user and print the result and grade of that student as pass if average is greater than 30 and grade as ('A' for ≥ 80 , $80 > B \geq 60$, $60 > C \geq 50$ else 'D').
10. Write a program that accept two character (alphabet) from user, and print message that "Both character are same" if both alphabet are same. If both are same case convert the 2nd alphabet as the opposite case of the 1st alphabet.
11. Write a program that accept a character from user, and print its next and previous character. NOTE: if the entered character is 'a' then 'b' & 'z' is next and previous character of character 'a'.
12. WAP that will accept Basic Pay (BP) of an employee then calculate the Gross Pay (GP) and Net Pay (NP) as follows –

$$GP = BP + HRA + DA ; \quad NP = GP - Tax$$

BP	HRA	DA	Tax
<12000	10% of BP	50% of BP	Nil
≥ 12000 and <30000	15% of BP	55% of BP	12% of GP
≥ 30000	20% of BP	60% of BP	15% of GP

13. Write a program that accepts a character and changes its case(upper to lower and lower to upper)
14. You have four colors to paint 100 houses in a lane, those are RED, YELLOW, WHITE, BLUE. You have made a decision that the colors will be given in sequence, i.e House No 1 : RED, House No 2: Yellow, House No 3: White, House No 4: blue and House No 5 will be again paint with RED and this sequence will continue for all houses. Now write a program to print the color if user gives house no as input.

15. A company gives festival discount on purchase of their products(minimum of 3) in the following percentages:
If purchase amount < 1000 then 5% in total amount(except mincost product) and 7% in minimum cost product.
If purchase amount >1000 and <=10000 then 7% in total amount(except mincost product) and 9% in minimum cost product.
If purchase amount > 10000 and <=100,000 then 9% in total amount(except mincost product) and 11% in minimum cost product.
If purchase amount >100,000 then 15% discount in total price. Write a C program to compute the amount to be paid by the customer after discount.
16. WAP that will accept the aggregate mark and total marks and finds the percentage of marks and display the grade of a student. The grading system follows following criteria –
% of Marks between 90 to 100 is: Grade-A+
% of Marks between 80 to 90 is: Grade-A
% of Marks between 70 to 80 is: Grade- B+
% of Marks between 60 to 70 is: Grade-B
% of Marks between 50 to 60 is: Grade-C
% of Marks less than 50is: Grade-F
17. Use ternary operator to check whether a given number is odd or even.
18. Use ternary operator to check whether a given character is consonant or vowel.
19. Use ternary operator to find out maximum of three given integers.
20. Use ternary operator to check whether a character is in upper case or lower case.

Laboratory-5

Control statements: Loops

1. Write a program that prints the number from 1 to 20.
2. Write a program to print all even numbers from 100 to 1.
3. Write a program to find the sum of all odd numbers between 1 to n. The value of n is entered by the user.
4. Write a program to print all the numbers divisible by either 2 or 3 but not by both.
5. Print Series: 1,4,9,16,25,36,.... 20 elements
6. Print Series: 0, 7, 26, 63,.....20 elements
7. Print Fibonacci series: 0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 ... 20 elements.
8. Print Lucas series: 1, 3, 4, 7, 11, 18, 29,..... 20 elements.
9. Print series : 1, 2, 2, 3, 4, 6, 9, 14, 22,..... 20 elements.
10. Print series: 1, 2, 4, 7, 11, 16, 22, 29, 37,..... 20 elements.
11. Print series: 0, 1, 2, 4, 6, 10, 12, 16, 18, 22 20 elements.
12. Write a program to compute, $\text{sum} = 1 - 1/2 + 1/3 + 1/4 + \dots + 1/n$.
13. Compute a series: $\text{Cos}(X) = 1 - X^2/2! + X^4/4! - \dots - X^n/n!$ for a given x in radians.
14. Compute a series: $\text{Sin}(X) = X - X^3/3! + X^5/5! - X^7/7! + \dots + X^n/n!$ for a given X in radians.
15. Write a program to find the multiplication table of a given number.
16. Write a program that prints the ASCII value of all caps letters using for loop.
17. Write a program that prints the ASCII value of all small letters using for loop.
18. Write a program to convert the decimal number to binary, hexadecimal and octal equivalent.
19. Write a program that takes a positive integer as input and finds out the minimum number of bits required to store the number.
20. Write a program that checks whether a number is a power of 2 or not.
21. Write a program to compute Factorial of a given number (n).
22. Write a program to compute, $\text{sum} = 1! + 2! + 3! + 4! + \dots + n!$.
23. Write a program to check the given number is Prime or not.
24. Write a program to print all prime number from 1 to N.
25. Write a program that accept a number from user and check whether the entered number is **Armstrong** number or not.
26. Write a program to check whether a number is strong or not.
27. Write a program that accept a number from user and print the sum of digits of the number (i.e. $123 = (3+ 2+1)$).
28. Write a program to check whether a number is perfect or not.
29. Write a program that accepts one integer from user and finds out largest digit in it

30. Write a program that accept a number from user and print that in reverse form i.e. (123 to 321). Check whether the number is palindrome or not.
31. Write a program that computes LCM & GCD of two given integers using while loop
32. Write a program that computes GCD and LCM of two given numbers using do-while loop.
33. generate the following pattern

*	*	*	a	1
**	**	***	ab	21
***	***	*****	abc	321
****	****	*****	abcd	4321
*****	*****	*****	abcde	54321

```

1
22
333
4444
55555

```

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1

```

```

      *
     ***
    *****
   *****
  *****
 ***
 *
```

```

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Laboratory-6

Functions and Recursive Function

1. Write a program that has a user defined function `sum()` which takes three integers as argument and returns the sum of three numbers.
2. Write a program to find summation, subtraction, multiplication and division of two integers based on users choice. Use four user defined functions to do the arithmetic operations.
3. Write functions for finding GCD and LCM of two positive integers using two user defined functions `GCD()`, `LCM()`.
4. Write a function that takes a number as parameter and returns its sum of digits.
5. (**Find the Minimum**) Write a function that returns the smallest of three floating-point numbers.
6. Write a function **`isPrime()`** that checks whether an integer is prime or not.
7. Write a Boolean function **`isPerfect()`** that checks whether a number is perfect or not.
8. Write a program to illustrate indirect assignment using pointer.
9. Write a function **`badSwap()`** that tries to swap two values passed as parameter. Write a swap function that swaps values of two integers passing pointers to them.
10. Write a function that finds factorial of a given integer. Write both non recursive and recursive versions.
11. Write a recursive function that finds GCD of two positive integers.
12. Write a function that takes an integer `n` and returns `fib(n)`, the `n` th Fibonacci no.
13. Write a Boolean function **`isPalindrome()`** that checks whether a string is palindrome or not.
14. Write a recursive function to findout factorial of a given integer.
15. Write a recursive function to multiply two numbers `m` and `n`.
16. Write a recursive function to find sum of an AP series with `n` numbers.
17. Write a recursive function to find sum of an GP series with `n` numbers.
18. Write a recursive function to find value of a^n where `a` is an integer and `n` is a positive integer.
19. (**Reversing Digits**) Write a function that takes an integer value and returns the number with its digits reversed. For example, given the number 7631, the function should return 1367.
20. Write a Boolean function **`isStrong()`** to check whether a number is strong or not. (A number is called strong number if sum of the factorial of its digit is equal to number itself. For example: 145 since $1! + 4! + 5! = 1 + 24 + 120 = 145$)
21. (**Recursive main**) Can `main` be called recursively? Write a program containing a function `main`. Include static local variable `count` initialized to 1. Postincrement and print the value of `count` each time `main` is called. Run your program. What happens?

Laboratory-7

Arrays and Strings

1. Write a program that accepts all elements of an integer array and finds out maximum and minimum element of the array.
2. Write a program that accepts all elements of an integer array and finds out sum and average among the values.
3. Write a program that generated the first 10 Fibonacci numbers and store them into an array of size 10.
4. Write a program that accepts all elements of an integer array and finds out mean and standard deviation of the array.
5. Write a program that accepts all elements of a double array and finds its maximum and next maximum elements without using sorting.
6. Write a program that accepts two integer arrays, add them and store them in a third array.
7. Write a program that accepts a string and finds its length without using any library function.
8. Write a program that sorts an array of integers using bubble sort.
9. Write a program that accepts a string and count the total no of capital letters, small letters and digits present in it.
10. Write a program to take two string inputs and find out whether they are same or different.
11. Write a program to reverse a string without using any extra space.
12. Write a program to take a string and print the first character of each word in the string.
13. Write a program to take a string and print it in title case, sentence case.
14. Write a program that concatenates two strings. The second string is concatenated after the first string and it becomes the resultant string.
15. Write a program to check whether a string is a palindrome or not.
16. Write a program that accepts a string and counts the no of vowels (both uppercase and lowercase) in it.
17. Write a program that accepts a sentence and counts the no of words in it and prints.
18. Write a program that inputs four strings that represent integers, converts the strings to integers, sums the values and prints the total of the four values.
19. Write a program to take two strings and find out whether the first string is present in the second or not if it is present program should return position of occurrence.
20. Write a program that adds two matrices and store the result in a third matrix.
21. Write a program that multiplies two matrices and store the result in a third matrix.
22. Write a program to read a matrix and check it for Identity Matrix.
23. Write a program to accept an array of integers. Allow user to insert a new element as desired position or delete an element from any position.
24. Write a program to take two 2-dimensional arrays to store country and capital names and should return a capital name for given country name.
25. Write a program that finds out the transpose of a square matrix and store it in the same array.

26. Write a C program to find the frequency of odd numbers and even numbers in the input of a matrix.
27. Write a C program to accept a matrix of order M x N and store its elements and interchange the main diagonal elements of the matrix with that of the secondary diagonal elements.
28. Write a C program to accept a string and find the sum of all digits present in the string.
29. Write a C program to accept a matrix of order M x N and find the sum of each row and each column of a matrix.
30. Write a C program to accept a matrix and determine whether it is a sparse matrix. A sparse matrix is matrix which has more zero elements than nonzero elements.
31. Write a function that takes an array as parameter and return the sum of the elements.
32. Write functions **myStrlen()** that finds length of a string without doing any modification on it.
33. Write functions **myStrcpy()** and **myStrcat()** that works like strcpy and strcat library functions.
34. Write a function myatoi that converts one string into integer.[“ 1234”→ 1234]
35. Write a function myitoa that converts one integer into its equivalent string.[1234→ “ 1234”]
36. Write a function **myStrcmp** that compares two strings lexicographically.
37. Declare one pointer to function that returns one integer and accepts two integer parameter. Define two functions GCD and FMAX that accepts two integers. Call the functions using the function pointer.
38. Write a program that generates all possible permutation with few characters.[if x,y,z are given then it would generate all 6 possible permutations]
39. Write a function that returns the index of the largest value stored in an array-of double. Test the function in a simple program.
40. Write a recursive function to display a string in reverse order.
41. Write a recursive function to find the minimum element of the array.
42. Write a recursive function printArray that takes an array and the size of the array as arguments, prints the array, and returns nothing. The function should stop processing and return when it receives an array of size zero.
43. Write a recursive function to sort an array.
44. Write a recursive function to perform linear search. The index of the element is returned if the element is found otherwise -1 is returned.
45. Write a recursive function to perform binary search.

Laboratory-8

Pointers, Storage classes & Dynamic Memory Allocation

1. Write a program to sort an array using insertion sort. Use a user defined function insertionSort() to sort the dynamically allocated array and another user defined function display() to display the array.
2. Write a program to print the value and address of a variable.
3. Write a program to read values for an array and find the second minimum of the array using different array and pointer syntax, (like *(ptr+i), ptr[i], i[ptr], arr[i], * (arr+i)
4. Take an array of floating point numbers and test pointer arithmetic like pointer + int, pointer-int, pointer -pointer etc.
5. Write a swap function to swap the values of the argument passed to it.
6. Write a program to implement a 2D array to store multiple strings and sort them.
7. Write a program to test dangling pointers by returning the address of a local variable from a function.
8. Write a C program to read N integers and store them in an array A, and so find the sum of all these elements using pointer. Output the given array and the computed sum with suitable heading.
9. Write a program to count how many times a function is called.
10. Write a program to perform add, subtract, multiplication or division of two numbers based on user choice without using any control structure or ternary operator. (Note: Using Function Pointers)
11. Write a program to dynamically allocate an array and reverse the array using a user defined function reverseArray() without using any extra space. Note: Allocate the array within the user defined function, reverse it and return the array to the calling function.
12. Write a program to demonstrate the use of Extern variable.
13. Write a program to add two matrices using a user defined function addMatrix() which returns the dynamically allocated resultant matrix.
14. Write a program to demonstrate different way to access a 2D array with pointer notation and array notations. Like (arr[i][j], *((ptr+i)+j), *(ptr[i]+j) etc.
15. Write a function to take two metrics as argument (using pointers) and return the result of their multiplication as pointer to the resultant array.

Laboratory-9

Structure , Union and Enums

1. Write a program to declare a structure DATE containing day, month and year. Find the age of a person if the date of birth and current date is given.
2. Write a program to declare a structure for TIME. Define a user defined function subtractTime() that accepts two times and returns the differenced time.
3. Write a program to declare a structure for a complex number. Define user defined functions add(), subtract(), multiply() and display() to do their respective function.
4. Write a program to declare a STUDENT structure having members Name,Roll, Branch. Define n students dynamically initialize them using a user defined function and display them using another user defined function.
5. Create a structure that stores a point in 2D. Accept 2 points and find out the distance between the two points.
6. Create a structure that stores name, rollno, and branch of a student. Accept data of 20 students. Write a user defined function search() that searches a student record by roll no and displays the student information.
7. Write a function that accepts two points in space and finds out the Euclidean distance between them. Call this function to find out the area of a triangle formed by 3 points.
8. Create a structure for a 2D point. Create also a structure of a line segment whose two end points are given. Write a boolean function isIntersect that returns true when the line segments intersect else returns false.
9. Either registration no or roll no is a must for student info. Apart from this, name, roll no and branch is must for a student. Create a suitable structure/union for a student. Enter 10 students information and display them.
10. One array of numbers to be sorted. The no of element of the array is a user input. Create the array dynamically, accept its members and sort the array using selection sort.
11. Write two small programs to illustrate memory leak/garbage and dangling pointer/reference.
12. Dynamically create 3 square matrices of same sizes (a user input), scan elements of the first two matrices, add them and store the result in the third matrix. Use only malloc function to allocate memory dynamically.
13. Dynamically create 3 square matrices of same sizes (a user input), scan elements of the first two matrices, add them and store the result in the third matrix. Use only calloc function to allocate memory dynamically.
14. Declare an enum for all weekdays. Enter a day of the week (0 -6 , 0-Sunday, 6 - Saturday) and print the day name.

Laboratory-10

FILE, I/O and Command Line Argument

1. Write a program to merge two files i.e. append the contents of one file after another file.
2. Write a program that reads 10 integers from an existing ASCII file, square the numbers and write into another ASCII file sqr.txt. Cube the numbers and write them into a file cube.txt.
3. Write a program to read a C file and display the comment lines of that C file.
4. Write a program that reads few integers from an ASCII file data.txt. Writes all even integers into even.txt and all odd integers into odd.txt.
5. Student data [roll no, name,branch] is available in an ASCII file student.txt. Read all the data and write them on different files depending on their branch [All CSE student data in cse.txt etc]
6. Write a program to display the contents of a file in reverse order.
7. Write a program to read a file and count the number of characters, words and lines in that file.
8. Write a program myCmp.c that will accept two file names from the command line and will compare the files.
9. Write a program myFileCopy.c that copies one file into another. Both the file names should be taken as input from user.
10. Write a program myEcho.c using command line argument that will echo a string. Compile the program by gcc myEcho.c -o myEcho. Run the program by myEcho hello world and it is expected that it would print hello world.
11. Write a program myCalc.c using command line argument that works like a mini command line calculator. Compile the program by gcc myCalc.c -o myCalc. Run it like mycalc 35 + 22 and the expected output would be 57.
12. Write a program myFileCopy.c that copies one file to another using command line argument.
13. Write a program myGrep.c that works like grep command using command line argument.
14. Write a program myCat.c that works like cat command using command line argument.