

LAB MANUAL
PROGRAMMING LAB. IN C
(CS-1091)

FOR B.TECH/DUAL-M.TECH/DUAL-MBA FIRST SEMESTER STUDENTS



School of Computer Engineering
KIIT UNIVERSITY, BHUBANESWAR

INTRODUCTION

- Programming cannot be learned by watching others do it. Students must spend numerous hours working on programs themselves.
- This laboratory manual is a tool that will allow students to experiment with computer science & this is the beginning. As students progress through each laboratory, they may wonder how or why something works. The best way to discover the answer is to try things out.
- The purpose of this lab. manual is to acquaint the students to know the programming language as well as developing programming skills using C language.

STRUCTURE OF THIS LAB. MANUAL

- This lab. manual provides study aids from programming assignments to scheduled exercises using prepared materials.
- This lab manual is divided into 10 laboratory classes. Each laboratory class consists of the following:
 - a) **Sample Answers (SA):** These are the complete program samples that students will go through in detail before coming to the laboratory class, may refer during solving lab assignments.
 - b) **Lab Assignments (LA):** These are the assignments that ask each student to independently create small programs during the lab time.
 - c) **Home Assignments (HA):** These are the assignments to be done during lab time if lab. assignments are completed before lab. time or may be assigned as post-lab homework and submitted in the next lab class.

The approach of each Lab: **SA-LA-HA**

INSTRUCTIONS FOR STUDENTS

To make laboratory experiments effective, each student must obey the following rules:

1. **General instructions**

- Once you create a directory named as your rollno_section under the home directory of UBUNTU OS system using command-line or by GUI.
- In Each lab, store programs within appropriate folders named as LAB01, LAB02, LAB03...etc. which are the sub folders under your rollno_section folder.
- Always save programs files with the meaningful name preceded by lab assignment no within specified folders. If you want solve a lab assignment no. HA3.5 (3.5 means 5th assignment of 3rd lab) which is to find roots of a quadratic equation, then name the program as SA35_quadratic.c or SP35_quadeq.c etc.

2. **Attendance:** Attendance is required at all labs without exception. There are no make-up labs in this course. Performance will be judged based on the experiments conducted, quality and punctual submission of the labs reports for each experiment. Faculty/Instructor will take attendance. Failure to be present for an experiment will result in losing entire marks for the corresponding lab. However, genuine cases may be considered for repeat lab. If a student misses a lab session due to unavoidable circumstances can provide a legitimate proof as soon as possible, he/she may be then be allowed by the lab instructor, to make-it-up.
3. **Laboratory Report:** At the end of every lab student will be assigned to write-up one of the experiment's problem. Your report must present a clear and accurate account, results you obtained. Student should develop habit to submit the laboratory report/assignments continuously and progressively on the scheduled dates and should get the assessment done.
4. Read the write up of each experiment to be performed, a day in advance. Understand the purpose of experiment and its practical implications.
5. Student should not hesitate to ask any difficulty faced during conduct of practical / exercise.
6. The student shall study all the questions given in the laboratory manual and practice to write the answers to these questions.
7. Student shall develop the habit of evolving more ideas, innovations, skills etc. those included in the scope of the manual.
8. Student should develop the habit of not to depend totally on teachers but to develop self learning techniques.
9. While entering into the LAB students should wear their ID cards.
10. Shut down your system after you have finished with your experiment.

LAB - 1

Linux/Unix Commands

Compilation, Execution of a program in GCC Compiler

CONTENTS

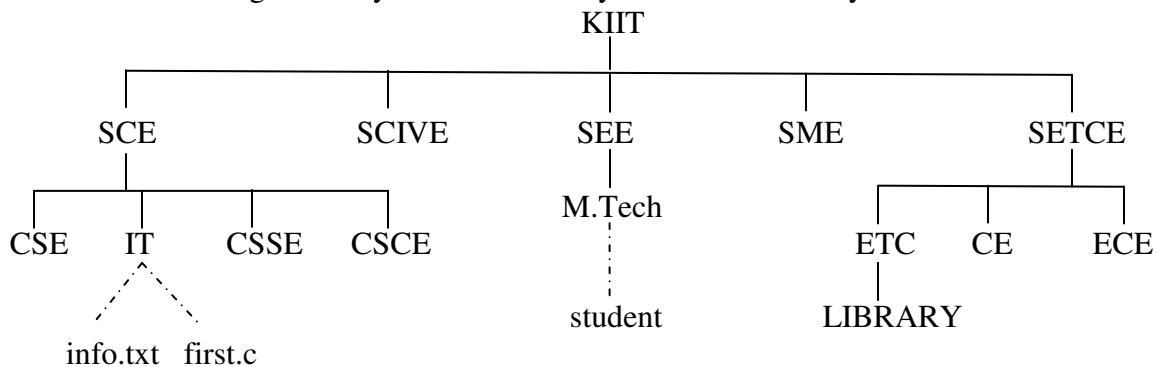
Experiment No-1/Lab-01

Sample Answers

- SA1.1** To get familiar with LINUX/UNIX (UBUNTU) Operating System and practice some frequently used commands on terminal (Command Prompt).
- SA1.2** To get familiar with **gedit** editor to create a new file, read the contents of a file, write into a file or modify the contents of a file.
- SA1.3** To learn how to compile and execute a C file that displays “Welcome to C Programming Laboratory” in **gcc** compiler on terminal (Command Prompt).
- SA1.4** WAP to display “IIT” using the character ‘*’.
- SA1.5** WAP to display the following message by using multiple printf statement.
 A Good End
 Can Only Be Achieved
 Only By Good Means.
- SA1.6** WAP to display the above message by using a single printf statement.

Lab. Assignments

- LA1.1** First create a sub-directory named as your roll number under your home directory. Then create the following directory structure under your rollno directory.



N.B. The names under solid lines are assumed as directories and dotted lines as file names.

Do the following operations

- Create the file names under the directories as mentioned in the figure and write some relevant data into the files.
- Rename the file info.txt as itstudentsdata.txt.
- Copy the file first.c into the directory CE with the same name.
- Copy the file first.c into the directory SME with a new name as hello.c.
- Transfer the file student into the directory SCIVE and check whether transferred or not.

- LA1.2** WAP to display “KIMS” using the character ‘#’.

LA1.3 WAP to display the following message by using multiple printf statement.

If The End Is Good,
Then It Is Good,
Whatever Be The Means.

LA1.4 WAP to display the message of LA 1.3 by using single printf statement.

Home Assignments

HA1.1 WAP to print your BIO-DATA (Name, Regd.no”, Branch, JEE Rank, Gender, Phone no., Address etc.) using printf statement.

PROGRAM NO. SA1.1

To get familiar with LINUX/UNIX (UBUNTU) Operating System and practice some frequently used commands on terminal (command prompt).

LINUX/UNIX COMMANDS

\$ _ Command Prompt

Anything written within [] is optional.

| Sl. No. | Command | Description | Example |
|---------|---------|---|---|
| 1. | man | <p><u>Manual Syntax</u></p> <p>man commandName</p> <p>It displays an on-line manual page for a command that it gives detailed information of a command how to use it.</p> | <p>\$ man ls It gives the manual page of ls command</p> <p>\$ man pwd You will see the manual for the pwd command.</p> |
| 2. | ls | <p><u>List Syntax</u></p> <p>ls [option(s)] [file(s)]</p> <p>It lists the contents of a directory, and can be used to obtain information on the files and directories within it.</p> | <p>\$ ls It lists the files & subdirectories available in the current directory.</p> <p>\$ ls -l Same as above except it lists the files 'long format', which contains lots of useful information, e.g. the exact size of the file, who owns the file and who has the right to look at it, and when it was last modified.</p> <p>\$ ls dir1 It lists the files & subdirectories available in dir1.</p> |
| 3. | pwd | <p><u>Print Working Directory Syntax</u></p> | <p>\$ pwd It tells you where you currently are, in which directory.</p> |

| | | | |
|----|--------------|---|---|
| | | <p>pwd</p> <p>It Shows the current location in the directory tree. In other words, the command gives the full pathname of your current directory.</p> | |
| 4. | cd | <p><u>Change Directory</u> Syntax</p> <p>cd [options(s)] [directory]</p> <p>It changes the current directory to other directory depending on the options and/or name of the directory.</p> | <p>\$ cd It changes to the user's home directory.</p> <p>\$ cd ~ Same as above.</p> <p>\$ cd dir1 It changes to the directory dir1 if dir1 is a sub directory of your current working directory.</p> <p>\$ cd /home/user1/kiit/csit It changes to csit directory as mentioned in the full path from your current working directory.</p> <p>\$ cd .. It simply move up one directory. For example, if you are in /home/user1/kiit/csit and you type “cd ..”, you will end up in /home/user1/kiit. After applying cd .. you can verify with pwd command.</p> |
| 5. | mkdir | <p><u>Make Directory</u> Syntax</p> <p>mkdir [option(s)] directoryName</p> <p>It creates a new directory.</p> | <p>\$ mkdir sce It creates a new directory named as sce under your current directory. After executing this command, check through ls whether sce directory is created or not.</p> <p>If you want to create a new directory under other than your current directory, then mention the full path name before the new directory name. As for example, if your current working directory is /home/user1/kiit, but you want to create a new directory named as ece under /home/user1/kiit/setce, then execute the following command: \$ mkdir /home/user1/kiit/setce/ece</p> |
| 6. | cp | <p><u>Copy</u> Syntax</p> | <p>\$ cp file1 file2 It copies the contents of the file file1 into</p> |

| | | | |
|----|----|---|---|
| | | <p>cp [option(s)] sourcefile targetfile</p> <p>Copies sourcefile to targetfile. Both file will be present.</p> | <p>a new file called file2. If you apply ls command, it will show you both the files.</p> <p>\$ cp ak.txt bk.txt dir1 It creates copies of files ak.txt and bk.txt (with the same names), within the directory dir1. dir1 must already exist for the copying to succeed.</p> <p>\$ cp file1 /home/user1/kiit/scive It copies the contents of the file file1 into the directory scive with the same name.</p> <p>\$ cp -r dir1 dir2 It recursively copies the directory dir1, together with its contents and subdirectories, to the directory dir2.</p> <p>\$ cp -i quard.c quradeq.c It waits for confirmation, if necessary, before an existing targetfile quardeq.c is overwritten. quard.c is copied as quradeq.c. Now If you apply ls command Then it will show you both the files quard.c and quradeq.c.</p> |
| 7. | mv | <p>Move Syntax</p> <p>mv [option(s)] sourcefile targetfile</p> <p>It moves a file to a new location, or renames it. Source file name will be deleted.</p> | <p>\$ mv info.txt itstudentsdata.txt It simply renames the file info.txt as itstudentsdata.txt . info.txt is deleted and only file available in current directory is itstudentsdta.txt.</p> <p>\$ mv -b info.txt itstudentsdata.txt It ceates a backup copy of the sourcefile info.txt before moving it to itstudentsdata.txt. It is similar to cp command.</p> <p>\$ mv /home/user1/kiit/sce/it/first.c home/user1/kiit/sme It simply move or transfer the file first.c into sme directory, no matter where is your current directory as both source and destination paths are mentioned.</p> <p>\$ mv /home/user1/kiit/sce/it/first.c home/user1/kiit/sme/firstcprog.c</p> |

| | | | |
|-----|---------|---|--|
| | | | <p>It simply move or transfer the file first.c into sme directory with a new name firstcprog.c</p> <p>\$ mv -i quard.c quradeq.c It waits for confirmation, if necessary, before an existing targetfile quardeq.c is overwritten. quard.c is renamed as quradeq.c. Now If you apply ls command Then it will show you only quradeq.c.</p> |
| 8. | rm | <p><u>Remove Syntax</u></p> <p>rm [option(s)] file(s)</p> <p>It removes the specified files from the file system. Directories are not removed by rm unless the option -r is used.</p> | <p>\$ rm quard.c It deleted the file quard.c available in the current directory.</p> <p>\$ rm -i quard.c It waits for confirmation before deleting quard.c</p> |
| 9. | rmdir | <p><u>Remove Directory Syntax</u></p> <p>rmdir [option(s)] directoryName</p> <p>It deletes the specified directory, provided it is already empty.</p> | <p>\$ rmdir dir1 If dir1 is empty, then it deletes the directory dir1 present under current directory.</p> |
| 10. | whereis | <p>Syntax whereis file It shows possible locations of file.</p> | <p>\$ whereis quard.c It shows you the location of the quard.c file.</p> |

Other LINUX/UNIX Commands (to know the detail about the following command use man)

date, cat, tail, which, locate, find, ps, id, du, clear, echo, grep, sot, su, ln, kill, chmod, ssh, tar, gzip, ping etc.

PROGRAM NO. SA1.2

To get familiar with **gedit** editor to create a new file, read the contents of a file, write into a file or modify the contents of a file.

gedit:

Text Editor (gedit) is the default GUI text editor in the Ubuntu operating system

1. To create a new file in c (first.c), run the following in command prompt.

\$ gedit first.c

It will open the gedit editor window with the name first.c where you can write anything (program code for first.c).

Then save the contents of this file by choosing the appropriate options from gedit menu as follows:

File → Save

Now quit from gedit window and return to command prompt, do the following:

File → Exit

2. To open an existing file (say first.c) for editing do the following:

\$ gedit first.c

After editing will be over, save the file and quit from gedit window.

PROGRAM NO. SA1.3

To learn how to compile and execute a C file that displays “Welcome to C Programming Laboratory” in gcc compiler on terminal (Command Prompt).

PROCEDURE

Step-1: Create a file named as **sa13_first.c** in gedit editor and write the following program code in it, then save the file and quit from gedit window.

Step-2: Compile the C Program file named as sa13_first.c

\$ gcc sa13_first.c

It compiles the file sa13_first.c, if it is error free, then go for execution to get output. Else open the file again in gedit to correct the errors, again compile it till it does not show any errors.

Step-3: To get the output do the following

\$./a.out

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    printf("\n Welcome to C Programming Laboratory \n");
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Welcome to C Programming Laboratory

PROGRAM NO. SA1.4

WAP to display “IIT” using the character ‘*’.

PROGRAM CODE

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

INPUT/OUTPUT**RUN-1**

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

PROGRAM NO. SA 1.5

WAP to display the following message by using multiple printf statement.

A Good End
Can Only Be Achieved
Only By Good Means.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    printf("\n A Good End ");
    printf("\n Can Only Be Achieved ");
    printf("\n Only By Good Means.");
    return 0;
}
```

INPUT/OUTPUT

RUN-1

A Good End
Can Only Be Achieved
Only By Good Means.

PROGRAM NO. SA 1.6

WAP to display the following message by using multiple printf statement.

A Good End
Can Only Be Achieved
Only By Good Means.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    printf("\n A Good End\nCan Only Be Achieved\nOnly By Good Means.");
    return 0;
}
```

INPUT/OUTPUT

RUN-1

A Good End
Can Only Be Achieved
Only By Good Means.

LAB - 2

Operators & Expressions, Simple Input/Output Statemets

CONTENTS

Experiment No-2

Sample Answers

- SA2.1** WAP to perform the addition of two integers and display the result.
- SA2.2** WAP to find centigrade for a given Fahrenheit temperature.
- SA2.3** WAP to calculate area of a circle.
- SA2.4** WAP to calculate area of a triangle whose base and height is given.
- SA2.5** WAP to swap two integer numbers using third variable.
- SA2.6** WAP to convert given paisa into its equivalent rupee and paisa as per the following format.
Example. 550 paisa = 5 Rupee and 50 paisa
- SA2.7** WAP to convert given second into its equivalent hour, minute and second as per the following format. Example. 7560 second = 2 Hour, 27 Minute and 40 Second

Lab. Assignments

- LA2.1** WAP to subtract a number from another number and display the result.
- LA2.2** WAP to convert temperature from centigrade to Fahrenheit scale.
- LA2.3** WAP to calculate perimeter of a circle.
- LA2.4** WAP to calculate area of a triangle whose three sides are given.
- LA2.5** WAP to swap two integer numbers without using third variable.
- LA2.6** WAP to convert a quantity in meter entered through keyboard into its equivalent kilometer and meter as per the following format. Example. 2430 meter = 2 Km and 430 meter.

Home Assignments

- HA2.1** WAP to find the average mark of 5 subjects of a student and find the percentage. Assume full mark of each subject is 100.
- HA2.2** WAP swap the contents of two variables by using a single statement for swap in C.
- HA2.3** WAP to add two times in hour, minute & second format entered through the keyboard in the format hh:mm:ss

PROGRAM NO. SA 2.1

WAP to perform the addition of two integers & display the result.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int a, b, c;
    printf("\nEnter two numbers to add :");
    scanf("%d%d",&a,&b);
    c = a + b;
    printf("\n%d + %d = %d", c);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter two numbers to add: 2 6

2 + 6 = 8

RUN-2

Enter two numbers to add: 2 8

2 + 8 = 10

PROGRAM NO. SA 2.2

WAP to find centigrade for a given Fahrenheit temperature.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    float f,c;
    clrscr();
    printf("\nEnter a temperature in Fahrenheit scale=>");
    scanf("%f",&f);
    c=(f-32.0)/1.8;
```

```
printf("\nThe equivalent temperature in celcius is %0.2f",c);  
return 0;  
}
```

INPUT/OUTPUT

RUN-1

Enter a temperature in Fahrenheit scale=>100
The equivalent temperature in celcius is 37.78

RUN-2

Enter a temperature in Fahrenheit scale=>32
The equivalent temperature in celcius is 0.00

PROGRAM NO. SA 2.3

WAP to calculate area of a circle.

PROGRAM CODE

```
#include <stdio.h>  
int main()  
{  
    float radius,area;  
    printf("\nEnter the radius of a circle : ");  
    scanf("%f",&radius);  
    area = 3.14 * radius * radius;  
    printf("\nArea of Circle : %f",area);  
    return 0;  
}
```

INPUT/OUTPUT

RUN-1

Enter the radius of circle: 2
Area of circle: 12.560000

RUN-2

Enter the radius of circle: 4
Area of circle: 50.24000

PROGRAM NO. SA 2.4

WAP to calculate area of a triangle whose base and height is given.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    float base,height,area;
    printf("\nEnter the values of base and height : ");
    scanf("%f %f",&base,&height);
    area = 0.5 * base * height;
    printf("Area of Triangle : %f",area);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter the values of base and height: 2 3
Area of Triangle : 3.000000

RUN-2

Enter the values of base and height: 3 4
Area of Triangle : 6.000000

PROGRAM NO. SA2.5

WAP to swap two integer numbers using third variable.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int a,b,temp;
    printf("\nEnter two integers a and b : ");
    scanf("%d%d", &a,&b);
    temp=a;
    a=b;
    b=temp;
    printf("\nAfter swapping a=%d and b=%d",a,b);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter two integers a and b : 2 3

After swapping a=3 and b=2

RUN-2

Enter two integers a and b : 10 20

After swapping a=20 and b=10

PROGRAM NO. SA 2.6

WAP to convert given paisa into its equivalent rupee and paisa as per the following format.

Ex. 550 paisa = 5 Rupee and 50 paisa

PROGRAM CODE

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

{
    int p,p1,r;
    printf("\nEnter paisa = ");
    scanf("%d",&p);
    r=p/100;
    p1=p%100;
    printf("\n%d paisa = %d rupees and %d paisa",p,r,p1);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter paisa = 2550

2550 paisa = 25 rupees and 50 paisa

RUN-1

Enter paisa = 245

2550 paisa = 2 rupees and 45 paisa

PROGRAM NO. SA 2.7

WAP to convert given second into its equivalent hour, minute and second as per the following format.

Ex. 7560 second = 2 Hour, 27 Minute and 40 Second

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    long sec1, sec2, hr, min, t;
    printf("\nEnter time in seconds: ");
    scanf("%ld", &sec1);
    hr = sec1/3600;
    t = sec1%3600;
    min = t/60;
    sec2 = t%60;
    printf("\n\n %ld second= %ld Hour %ld Minute and %ld Second",sec1, hr, min,sec2);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter time in seconds: 3750

3750 second=1 Hour 2 Minute 30 Second

RUN-2

Enter time in seconds: 4000

4000 second=1 Hour 6 Minute 40 Second

LAB - 3

Branching Statements (if..else, switch..case)

CONTENTS

Experiment No-3

Sample Answers

- SA3.1** WAP to find the largest between two numbers.
- SA3.2** WAP to read an alphabet from the user and convert it into uppercase if the entered alphabet is in lowercase, otherwise display an appropriate message.
- SA3.3** WAP to read a character from the user and test it whether it a vowel or consonant or not an alphabet.
- SA3.4** WAP to determine whether a year entered through the keyboard is a leap year or not.
- SA3.5** WAP to find the roots of a quadratic equation $ax^2+bx+c=0$ using if-else statement.
- SA3.6** WAP to display the grade system of KIIT University based on total marks secured by a student in a semester. Use switch-case statement.

Lab. Assignments

- LA3.1** WAP to input any two integers distinct and display the greater of two integers.
- LA3.2** WAP to input any three integers distinct and display the greater of three integers.
- LA3.3** WAP to test whether a number entered through keyboard is ODD or EVEN.
- LA3.4** WAP to read an alphabet from from the user and convert it into lowercase if the entered alphabet is in uppercase, otherwise display an appropriate message.
- LA3.5** WAP to input any two integers, and provide a menu to the user to select any of the options as add, subtract, multiply, divide and display the result accordingly.
- LA3.6** WAP to display the grade system of KIIT University based on total marks secured by a student in a semester. Use else..if ladder statement.

Home Assignments

- HA3.1** WAP to check whether a character entered through keyboard is a digit, letter, special character etc or not.
- HA3.2** WAP which takes two integer operands and one operator form the user, performs the operation and then prints the result. (Consider the operators +,-,*,/, % etc). Use switch cse.
- HA3.3** WAP to find the roots of a quadratic equation $ax^2+bx+c=0$ using switch-case statement.

PROGRAM NO. SA 3.1

WAP to find the largest between two numbers.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int a,b;
    printf("\nEnter two numbers: ");
    scanf("%d %d",&a,&b);
    if(a>b)
        printf("\nLargest number is %d",a);
    else
        printf("\nLargest number is %d",b);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter two numbers: 5 3

Largest number is 5

RUN-2

Enter two numbers: 10 30

Largest number is 30

PROGRAM NO. SA 3.2

WAP to read an alphabet from the user and convert it into uppercase if the entered alphabet is in lowercase, otherwise display an appropriate message.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    char ch;
    printf("\n Enter an alphabet: ");
    scanf("%d", &ch);
    if (ch>='a' && ch<='z')
    {
        ch=ch-32;
        printf("\nThe uppercase of the entered alphabet is %c", ch);
    }
}
```

```
}  
else  
    printf("\nThe entered character is not a lower case alphabet");  
  
return 0;  
}
```

INPUT/OUTPUT

RUN-1

Enter an alphabet: g

The uppercase of the entered alphabet is G

RUN-2

Enter an alphabet: G

The entered character is not a lower case alphabet

PROGRAM NO. SA 3.3

WAP to read a character from the user and test it whether it a vowel or consonant or not an alphabet.

PROGRAM CODE

```
#include<stdio.h>  
int main()  
{  
    char ch;  
    printf("\n Enter an alphabet: ");  
    scanf("%d", &ch);  
    if ((ch>='a' && ch<='z') || (ch>='A' && ch<='Z'))  
    {  
        if (ch=='a' || ch=='A' || ch=='e' || ch=='E' || ch=='i' || ch=='I' || ch=='o' || ch=='O' || ch=='u' ||  
ch=='U');  
            printf("\nThe entered character %c is a vowel", ch);  
        else  
            printf("\nThe entered character %c is a consonat", ch);  
    }  
    else  
        printf("\nThe entered character %c is not an alphabet",ch);  
    return 0;  
}
```

INPUT/OUTPUT**RUN-1**

Enter an alphabet: B

The entered character B is a consonant

RUN-2

Enter an alphabet: i

The entered character i is a vowel

RUN-3

Enter an alphabet: %

The entered character % is not an alphabet

PROGRAM NO. SA 3.4

WAP to determine whether a year entered through the keyboard is a leap year or not.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int year;
    clrscr();
    printf("\nEnter the year:");
    scanf("%d",&year);
    if((year%4==0 && year%100!=0)|| (year %400==0))
        printf("\n%d is a leap year.",year);
    else
        printf("\n%d is not a leap year.",year);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter the year:2005

2005 is not a leap year.

RUN-2

Enter the year:1996

1996 is a leap year.

PROGRAM NO. SA 3.5

WAP to find the roots of a quadratic equation $ax^2+bx+c=0$ using if-else statement.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    float a,b,c,real,imag, dis,i1,i2,r1,r2;
    clrscr();
    printf("\nInput values for a, b and c=>");
    scanf("%f%f%f",&a,&b,&c);
    dis=b*b-4*a*c;
    if(dis==0)
    {
        printf("\nThe Roots are Equal");
        r1=r2=-b/(2.0*a);
        printf("\nRoots are %.2f and %.2f",r1,r2);
    }
    else if(dis>0)
    {
        printf("\nThe Roots are real & unequal.");
        r1=(-b+sqrt(dis))/(2.0*a);
        r2=(-b-sqrt(dis))/(2.0*a);
        printf("\nRoots are %.2f and %.2f",r1,r2);
    }
    else
    {
        printf("\nThe Roots are inaginary");
        real=-b/(2.0*a);
        dis=-dis;
        imag=sqrt(dis)/(2.0*a);
        printf("\nRoot1=%.2f+i%.2f",real, imag);
        printf("\nRoot2=%.2f-i%.2f",real, imag);
    }
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Input values for a, b and c=>1

2

1

The Roots are Equal

Roots are -1.00 and -1.00

RUN-2

Input values for a, b and c=>1 8 3

The Roots are real & unequal.

Roots are -0.39 and -7.61

RUN-3

Input values for a, b and c=>3 5 7

The Roots are imaginary

Root1=-0.17+i1.28

Root2=-0.17-i1.28

PROGRAM NO. SA 3.6

WAP to display the grade system of KIIT University based on total marks secured by a student in a semester. Use switch-case statement.

```
#include<stdio.h>
int main()
{
    int total_mark,tm;
    printf("\nEnter total mark secured by a student: ");
    scanf("%d",&total_mark);
    tm=total_mark/10;
    switch(tm)
    {
        case 9: printf("\nSecured grade is O");
                break;
        case 8: printf("\nSecured grade is E");
                break;
        case 7: printf("\nSecured grade is A");
                break;
        case 6: printf("\nSecured grade is B");
                break;
        case 5: printf("\nSecured grade is C");
                break;
        case 4: printf("\nSecured grade is D");
                break;
        default: printf("FAIL");
    }
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Enter total mark secured by a student: 55
Secured grade is C

RUN-2

Enter total mark secured by a student: 95
Secured grade is O

LAB - 4

Looping (while, do..while and for)

CONTENTS

Experiment No-4

Sample Answers

- SA4.1** WAP to print the natural numbers from 1 to 20.
- SA4.2** WAP to print all numbers within a given range. The range is given by user.
- SA4.3** WAP to calculate the factorial of a given number.
- SA4.4** WAP to calculate the sum of digits of a given number.
- SA4.5** WAP to find the GCD/HCF of two numbers .
- SA4.6** WAP to check whether a number n is prime number or not.
- SA4.7** WAP to check whether an input integer is perfect number or not.
- SA4.8** WAP to find the first n numbers of a Fibonacci sequence.
- SA4.9** WAP to evaluate the equation $y=x^n$ where n is a non-negative integer.
- SA4.10** WAP to sum the following series $S=1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+n)$
- SA4.11** WAP to print the following pattern for n rows. Ex. for n=5 rows

```

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

- SA4.12** WAP to print the following pattern for n rows. Ex. for n=5 rows

```

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

- SA4.13** WAP to print the following pattern for n rows. Ex. for n=5 rows

```

A
A B
A B C
A B C D
A B C D E
```

- SA4.14** WAP to form a pyramid of numbers for a given number. Ex. for number 4

```

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

- SA4.15** WAP to print the following pattern for n rows. Ex. for n=6 rows

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
```

Lab. Assignments

- LA4.1** WAP to print your name 5 times using while loop.
- LA4.2** WAP to input any integer and print your name that many times.
- LA4.3** WAP to print the series as 1 2 3 4 5 6 7100.
- LA4.4** WAP to print the series as 1 2 7 15 31n, where n is given by user.
- LA4.5** WAP to print the series as 1 1 2 3 5 8 13n, where n is given by user.
- LA4.6** WAP to print the series as 3 5 7 11 13 17.....n, where n is given by user.
- LA4.7** WAP to print all odd and even numbers separately within a given range. The range is input through user.
- LA4.8** WAP to display the reverse of a number entered through keyboard.
- LA4.9** WAP to check whether an integer number is a Armstrong number or not!.
- LA4.10** WAP to print the following pattern for n rows. Ex. for n=5 rows

```
A
B A
C B A
D C B A
E D C B A
```

- LA4.11** WAP to print the following pattern for n rows. Ex. for n=5 rows

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

Home Assignments

- HA4.1** WAP to check whether an input integer is strong number or not.
(Hint: If the sum of factorials of all digits of a number are equal to the number are equal to the number, it is called a strong number)
- HA4.2** WAP to find out the prime factors of a number entered through keyboard (distinct).
/*Hints: A prime number is any number with no divisors other than itself and 1, such as 2 and 5. Any number can be written as a product of prime numbers in a unique way (except for the order). These are called prime factors of a number. In other words, In number theory, the prime factors of a positive integer are the prime numbers that divide that integer exactly, without leaving a remainder. The process of finding these numbers is called integer factorization, or prime factorization.

- Enter a number : 100
- The prime factors of 100 are 2(2) and 5(2)
- That is, $100 = 2 \times 2 \times 5 \times 5$, and those numbers are primes. */

HA4.3 WAP to form reverse pyramid of numbers for a given number. Ex. for number 4

```

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

```

HA4.4 WAP to print the following pattern for n rows. Ex. for n=6 rows

```

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

```

HA4.5 WAP to generate the pascal triangle pyramid of numbers for a given number.

Ex. for number 4

```

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

```

HA4.6 WAP to display the following style o/p for a given string input through keyboard.(Ex.for a string “KIITCSIT”)

```

KIITCSITTISCTIHK
KIITCSI  ISCTIHK
KIITCS   SCTIHK
KIITC     CTIHK
KIIT      TIIK
KII       IIK
KI        IK
K         K
KI        IK
KII       IIK
KIIT      TIIK
KIITC     CTIHK
KIITCS    SCTIHK
KIITCSI   ISCTIHK
KIITCSITTISCTIHK

```

HA4.7 WAP to convert a decimal number into its equivalent binary number.

PROGRAM NO. SA 4.1

WAP to print the natural numbers from 1 to 20.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int i;
    printf("\nThe natural number's are: ");
    for(i=1;i<=10;i++)
        printf("%d  ",i);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

The natural number's are: 1 2 3 4 5 6 7 8 9 10

PROGRAM NO. SA 4.2

WAP to print all numbers within a given range. The range is given by user.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int min,max,i;
    printf("\nEnter minimum and maximum range:");
    scanf("%d%d",&min,&max);
    printf("\nThe number's are::\n");
    for(i=min;i<=max;i++)
    {
        printf("%d\t",i);
    }
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter minimum and maximum range: 20 45

The number's are::

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | | |

RUN-2

Enter minimum and maximum range: 2 14

The number's are::

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|----|----|----|----|----|
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|----|----|----|----|----|

PROGRAM NO. SA 4.3

WAP to calculate the factorial of a given number.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int n,i;
    long int fact=1;
    printf("\nEnter A Number To Find Out Its Factorial=>");
    scanf("%d",&n);
    for (i=n;i>0;i--)
        fact=fact*i;
    printf("\n%d!=%ld\n",n,fact);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter A Number To Find Out Its Factorial=>6

6!=720

RUN-2

Enter A Number To Find Out Its Factorial=>12

12!=479001600

PROGRAM NO. SA 4.4

WAP to calculate the sum of digits of a given number.

PROGRAM CODE

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

```
int main()
{
    int n,i,sum=0,d,num;
    printf("\nEnter a number: ");
    scanf("%d",&n);
    num=n;
    while(n!=0)
    {
        d=n%10;
        sum=sum+d;
        n=n/10;
    }
    printf("\nSum of digits of the number %d is = %d",num,sum);
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Enter a number : 234

Sum of digits of the number 234 is = 9

RUN-2

Enter a number : 389

Sum of digits of the number 389 is = 20

PROGRAM NO. SA 4.5

WAP to find the GCD/HCF of two numbers .

PROGRAM CODE

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

    printf("\nEnter two numbers=>");
    scanf("%d%d",&a,&b);
    m=a;n=b;
    while(a!=b)
    {
        if(a>b)
            a=a-b;
        else
            b=b-a;
    }
    gcd=b;
}
```

```
    printf("\nThe GCD of %d and %d is %d\n",m,n,gcd);
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Enter two numbers=>15 20

The GCD of 15 and 20 is 5

RUN-2

Enter two numbers=>34 88

The GCD of 34 and 88 is 2

PROGRAM NO. SA 4.6

WAP to check whether a number n is prime number or not.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int n,i,test=1;

    printf("\nEnter A Number :");
    scanf("%d",&n);
    for(i=2;i<=n/2;i++)
    {
        if(n%i==0)
        {
            test=0;
            break;
        }
    }
    if(test==1)
        printf("\nThe Number %d is a prime",n);
    else
        printf("\nThe Number %d is not a prime",n);
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Enter A Number :45

The Number 45 is not prime

RUN-2

Enter A Number :13

The Number 13 is prime

OR

PROGRAM NO. SA 4.6

WAP to check whether a number is prime number or not.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int a,c=0,i,n;
    printf("\nEnter a number to be checked: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
        {
            c=c+1;
        }
    }
    if (c==2)
        printf("\n%d is a prime number.");
    else
        printf("\n%d is not a prime number.");
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter a number to be checked: 12

12 is not a prime number.

RUN-2

Enter a number to be checked: 13

13 is a prime number.

PROGRAM NO. SA4.7

WAP to check whether an input integer is perfect number or not.

/*(Hints: In mathematics, a perfect number is a positive integer that is the sum of its proper positive divisors, that is, the sum of the positive divisors excluding the number itself. The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and $1 + 2 + 3 = 6$. The next perfect number is $28 = 1 + 2 + 4 + 7 + 14$. This is followed by the perfect numbers 496 and 8128)*/

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int n,i,sum=0;
    printf("\nEnter Any Number=>");
    scanf("%d",&n);
    for(i=1;i<n;i++)
    {
        if(n%i==0)
            sum+=i;
    }
    if(n==sum)
        printf("\n%d is a perfect number",n);
    else
        printf("\n%d is not a perfect number",n);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter Any Number=>371

371 is not a perfect number

RUN-2

Enter Any Number=>6

6 is a perfect number

PROGRAM NO. SA4.8

WAP to find the first n numbers of a Fibonacci sequence.

PROGRAM CODE

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

```
int main()
{
    int n,i,f1=0,f2=1,f;

    printf("\nEnter the number of terms :");
    scanf("%d",&n);
    printf("\nThe fibonacci series generated is :\n");
    printf("%d %d ",f1,f2);
    for(i=1;i<=n-2;i++)
    {
        f=f1+f2;
        printf("%d  ",f);
        f1=f2;
        f2=f;
    }
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Enter the number of terms :10

The fibonacci series generated is :

0 1 1 2 3 5 8 13 21 34

RUN-2

Enter the number of terms :15

The fibonacci series generated is :

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

PROGRAM NO. SA4.9

WAP to evaluate the equation $y=x^n$ where n is a non-negative integer.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int n,i;float x,y=1.0;
    printf("\nEnter values for x and n =>");
    scanf("%f%d",&x,&n);
    for(i=1;i<=n;i++)
    {
```

```
        y=y*x;
    }
    printf("\n%0.2f to the power %d is %0.2f",x,n,y);
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Enter values for x and n =>3.5 2

3.50 to the power 2 is 12.25

RUN-2

Enter values for x and n =>2 4

2.00 to the power 4 is 16.00

PROGRAM NO. SA4.10

WAP to sum the following series $S=1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+n)$

```
#include <stdio.h>
int main()
{
    int n,i,j,sum,totalsum=0;
    printf("\n Enter tbe value of n :");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        sum=0;
        for(j=1;j<=i;j++)
            sum=sum+j;
        totalsum=totalsum+sum;
    }
    printf("\nThe sum of the given series is %d",totalsum);
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Enter tbe value of n :7

The sum of the given series is 84

RUN-2

Enter the value of n :10

The sum of the given series is 220

PROGRAM NO. SA4.11

WAP to print the following pattern for n rows. Ex. for n=5 rows

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

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int n,i,j,k;

    printf("\nEnter how many rows =>");
    scanf("%d",&n);
    printf("\n");
    for(i=1;i<=n;i++)
    {
        for(k=1;k<=i;k++)
            printf("* ");
        printf("\n");
    }
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter how many rows =>4

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

RUN-2

Enter how many rows =>7

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

PROGRAM NO. SA4.12

WAP to print the following pattern for n rows. Ex. for n=5 rows

```

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

PROGRAM CODE

```

#include<stdio.h>
int main()
{
  int n,i,j,k;
  printf("\nEnter how many rows =>");
  scanf("%d",&n);
  printf("\n");
  for(i=1;i<=n;i++)
  {
    for(j=1;j<=n-i;j++)
      printf(" ");
    for(k=1;k<=i;k++)
      printf("* ");
    printf("\n");
  }
  return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter how many rows =>4

```

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

RUN-2

Enter how many rows =>7

```

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

PROGRAM NO. SA4.13

WAP to print the following pattern for n rows. Ex. for n=5 rows

```
A
A B
A B C
A B C D
A B C D E
```

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int i,j,n;
    char ch;

    printf("\nHow many rows=>");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        ch='A';
        for(j=1;j<=i;j++)
        {
            printf("%c ",ch);
            ch=ch+1;
        }
        printf("\n");
    }
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

How many rows=>4

```
A
A B
A B C
A B C D
```

RUN-2

How many rows=>6

```
A
A B
A B C
A B C D
A B C D E
A B C D E F
```

PROGRAM NO. SA4.14

WAP to form a pyramid of numbers for a given number. Ex. for number 4

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

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int n,i,j,k;
    printf("\nEnter a number to form a pyramid=>");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n-i;j++)
            printf(" ");
        for(k=1;k<=i;k++)
            printf("%d ",k);
        for(k=i-1;k>0;k--)
            printf("%d ",k);
        printf("\n");
    }
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter a number to form a pyramid=>5

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

RUN-2

Enter a number to form a pyramid=>7

```
  1
 1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1
```

PROGRAM NO. SA4.15

WAP to print the following pattern for n rows. Ex. for n=6 rows

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
```

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int i,j,n;
    printf("\nEnter The Number Of Rows =>");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=i;j++)
            printf("%4d",(i+j+1)%2);
        printf("\n");
    }
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter The Number Of Rows =>4

```
1
0 1
1 0 1
0 1 0 1
```

RUN-2

Enter The Number Of Rows =>7

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
1 0 1 0 1 0 1
```


LAB - 5

1-D Array & Matrix

CONTENTS

Experiment No-5

Sample Answers

- SA5.1** WAP to create an array that can store max. 50 integers and display the contents of the array
- SA5.2** WAP to find out the sum of the numbers stored in an array of integers.
- SA5.3** WAP to find largest element stored in an array.
- SA5.4** WAP to display the array elements in ascending order.
- SA5.5** WAP to add two matrices and display it.
- SA5.6** WAP to multiply two matrices and display it.
- SA5.7** WAP to find the Trace(sum of the diagonal element) of a given mxn matrix .

Lab. Assignments

- LA5.1** WAP to input 10 integers into an array of size 10. Print all elements.
- LA5.2** WAP to store max. 100 numbers into an array. Print all the elements that are three digit even integers.
- LA5.3** WAP to find out the largest even integer stored in the array of n integers. n is the user input.
- LA5.4** WAP to swap the pair of elements starting from beginning.
- LA5.5** WAP to arrange the numbers stored in the array so that it will display first all odd numbers, then even numbers.
- LA5.6** WAP to display the array elements in reverse order.
- LA5.7** WAP to find out the sum of the elements stored in a matrix.
- LA5.8** WAP to find out the transpose of a given matrix.

Home Assignments

- HA5.1** WAP to swap first element with last, second element with second last and so on, stored in an array.
- HA5.2** WAP to find out the second largest element stored in an array of 20 integers.
- HA5.3** WAP to find the median of a list of numbers.
- HA5.4** WAP to find the standard deviation of a list of numbers.
- HA5.5** WAP to sort 3 elements stored in an array without using any sorting algorithm.
- HA5.6** WAP to find out the sum of the diagonal elements of a matrix.
- HA5.7** WAP to check whether a given matrix is symmetric or not.
- HA5.8** WAP to check whether a given matrix is orthogonal or not.

PROGRAM NO. SA 5.1

WAP to create an array that can store max. 50 integers and display the contents of the array.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int a[50],i,n;
    printf("\nEnter how many numbers :");
    scanf("%d",&n);

    printf("\nEnter values for the array: ");
    for(i=0; i<n; i++) /* Stores number entered by user. */
        scanf("%d",&a[i]);

    printf("\nThe elements are:");
    for(i=1; i<n; i++) /* Loop to print the numbers */
        printf("%d\t",a[i]);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter how many numbers: 6

Enter values for the array:

23

34

45

56

67

78

The elements are

23 34 45 56 67 78

RUN-2

Enter how many numbers : 5

Enter values for the array:

11

22

83

44

55

The elements are

11 22 83 44 55

PROGRAM NO. SA5.2

WAP to find out the sum of the numbers stored in an array of integers.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int a[50],i,n, sum=0;
    printf("\nEnter how many numbers :");
    scanf("%d",&n);

    printf("\nEnter values for the array: ");
    for(i=0; i<n; i++) /* Stores number entered by user. */
        scanf("%d",&a[i]);

    /*code to find out sum of element in an array*/
    for(i=1; i<n; i++)
    {
        Sum=sum+a[i];
    }
    printf("\nThe sum of the elements stored in array a is %d", sum);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter how many numbers: 6

Enter values for the array:

23

34

45

56

67

78

The sum of the elements stored in array a is 303

RUN-2

Enter how many numbers : 5

Enter values for the array:

11

22

83

44

55

The sum of the elements stored in array a is 215

PROGRAM NO. SA5.3

Write a program to find largest element stored in an array.

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int a[50],i,n, large;
    printf("\nEnter how many numbers :");
    scanf("%d",&n);

    printf("\nEnter values for the array: ");
    for(i=0; i<n; i++) /* Stores number entered by user. */
        scanf("%d",&a[i]);

    /*code to find largest element in an array*/
    large=a[0];
    for(i=1; i<n; i++)
    {
        if(a[i]>large)
            large=a[i];
    }

    printf("\nThe largest element stored in array a is %d", large);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter how many numbers: 6

Enter values for the array:

23

34

45

56

67

78

The largest element stored in array a is 78

RUN-2

Enter how many numbers : 5

Enter values for the array:

11

22

83

44

55

The largest element stored in array a is 83

PROGRAM NO. SA5.4

Write a program to display the array elements in ascending order.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int a[100],n,i,temp;
    printf("\nEnter how many numbers :");
    scanf("%d",&n);
    printf("\nEnter the value of %d Numbers",n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    /*BUBBLE SORT technique*/
    for(i=1;i<=n-1;i++)
    {
        for(j=0;j<n-i;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    printf("\nThe Numbers in ascending order are\n");
    for(i=0;i<n;i++)
        printf("%d  ",a[i]);
    return 0;
}
```

INPUT/OUTPUT**RUN-1**

Enter how many numbers :7

Enter the value of 7 Numbers:-

7 6 4 5 2 4 8

The Numbers in ascending order are

2 4 4 5 6 7 8

RUN-2

Enter how many numbers :10

Enter the value of 10 Numbers:-

12 34 56 78 90 34 56 799 122 56

The Numbers in ascending order are

12 34 34 56 56 56 78 90 122 799

PROGRAM NO. SA5.5

Write a program to add two matrices and display it.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int a[2][3],b[2][3],c[2][3];
    int i,j;
    printf("\nEnter value of matrix a\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }

    printf("\nEnter value of matrix b\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    printf("\nValue of matrix a\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    printf("\nValue of matrix b\n");
    for(i=0;i<2;i++)
```

```
{
    for(j=0;j<3;j++)
    {
        printf("%d\t",b[i][j]);
    }
    printf("\n");
}
for(i=0;i<2;i++)
{
    for(j=0;j<3;j++)
    {
        c[i][j]=a[i][j]+b[i][j];
    }

}
printf("\nAfter addition resultant matrix is\n");
for(i=0;i<2;i++)
{
    for(j=0;j<3;j++)
    {
        printf("%d\t",c[i][j]);
    }
    printf("\n");
}
Return 0;
}
```

INPUT/OUTPUT

RUN-1:

Enter value of matrix a

1
2
3
4
5
6

Enter value of matrix b

2
1
3
2
4
2

Value of matrix a

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |

Value of matrix b

| | | |
|---|---|---|
| 2 | 1 | 3 |
| 2 | 4 | 2 |

After addition resultant matrix is

| | | |
|---|---|---|
| 3 | 3 | 6 |
| 6 | 9 | 8 |

RUN-2:

Enter value of matrix a

2
2
5
2
4
2

Enter value of matrix b

1
4
2
5
3
6

Value of matrix a

| | | |
|---|---|---|
| 2 | 2 | 5 |
| 2 | 4 | 2 |

Value of matrix b

| | | |
|---|---|---|
| 1 | 4 | 2 |
| 5 | 3 | 6 |

After addition resultant matrix is

| | | |
|---|---|---|
| 3 | 6 | 7 |
| 7 | 7 | 8 |

PROGRAM NO. SA5.6

Write a program to multiply two matrices and display it.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int a[3][3],b[3][3],c[3][3];
    int i,j,k;
    printf("\nEnter value of matrix a\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }

    printf("\nEnter value of matrix b\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    printf("\nValue of matrix a\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    printf("\nValue of matrix b\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n");
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
```

```
        c[i][j]=0;
        for(k=0;k<3;k++)
        {
            c[i][j]=c[i][j]+a[i][k]*b[k][j];
        }
    }

    printf("\nAfter Multiplication resultant matrix is\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
return 0;
}
```

INPUT/OUTPUT

RUN-1:

Enter value of matrix a

1
1
1
1
1
1
1
1
1
1

Enter value of matrix b

2
2
2
2
2
2
2
2
2
2

Value of matrix a

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

Value of matrix b

| | | |
|---|---|---|
| 2 | 2 | 2 |
| 2 | 2 | 2 |
| 2 | 2 | 2 |

After Multiplication resultant matrix is

| | | |
|---|---|---|
| 6 | 6 | 6 |
| 6 | 6 | 6 |
| 6 | 6 | 6 |

RUN-2:

Enter value of matrix a

2
2
2
2
2
2
2
2
2
2

Enter value of matrix b

3
3
3
3
3
3
3
3
3
3
3

Value of matrix a

| | | |
|---|---|---|
| 2 | 2 | 2 |
| 2 | 2 | 2 |
| 2 | 2 | 2 |

Value of matrix b

| | | |
|---|---|---|
| 3 | 3 | 3 |
| 3 | 3 | 3 |
| 3 | 3 | 3 |

After Multiplication resultant matrix is

| | | |
|----|----|----|
| 18 | 18 | 18 |
| 18 | 18 | 18 |
| 18 | 18 | 18 |

OR

PROGRAM NO. SA5.6

Write a program to multiply two matrices and display it.

PROGRAM CODE

```
#include<stdio.h>
int main()
{
    int a[10][10],b[10][10],c[10][10];
    int i,j,k,m,n,p,q;
    clrscr();
    printf("\nThe row & column of Matrix A :");
    scanf("%d%d",&m,&n);
    fflush(stdin);
    printf("\nThe row & column of Matrix B :");
    scanf("%d%d",&p,&q);
    if (n==p)
    {
        printf("\nFor Matrix A:-\n");
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                printf("\nEnter values for A[%d][%d]=> ",i,j);
                scanf("%d",&a[i][j]);
            }
        }
        printf("\nFor Matrix B:-\n");
        for(i=0;i<p;i++)
        {
            for(j=0;j<q;j++)
            {
                printf("\nEnter values for B[%d][%d]=> ",i,j);
                scanf("%d",&b[i][j]);
            }
        }
        //Matrix Multiplication Logic
        for(i=0;i<m;i++)
        {
            for(j=0;j<q;j++)
            {
                c[i][j]=0;
                for(k=0;k<n;k++)
                {
                    c[i][j]+=a[i][k]*b[k][j];
                }
            }
        }
    }
}
```

```
        }
    }
}

printf("\nMatrix A is\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
        printf("%d\t",a[i][j]);
    printf("\n");
}
printf("\n");

printf("\nMatrix B is\n");
for(i=0;i<p;i++)
{
    for(j=0;j<q;j++)
        printf("%d\t",b[i][j]);
    printf("\n");
}
printf("\n");
printf("\nMultiplication Matrix C is\n");
for(i=0;i<m;i++)
{
    for(j=0;j<q;j++)
        printf("%d\t",c[i][j]);
    printf("\n");
}
}
else
    printf("\nMultiplication is not possible.\n");
return 0;
}
```

INPUT/OUTPUT

RUN-1

The row & column of Matrix A :2 3

The row & column of Matrix B :3 1

For Matrix A:-

Enter values for A[0][0]=> 1

Enter values for A[0][1]=> 2

Enter values for A[0][2]=> 3

Enter values for A[1][0]=> 4

Enter values for A[1][1]=> 5

Enter values for A[1][2]=> 6

For Matrix B:-

Enter values for B[0][0]=> 7

Enter values for B[1][0]=> 6

Run the above code and check the output.

RUN-2

The row & column of Matrix A :3 4

The row & column of Matrix B :3 4

Multiplication is not possible.

PROGRAM NO. SA5.7

WAP to find the Trace(sum of the diagonal element) of a given mxn matrix .

PROGRAM CODE

```
#include <stdio.h>
int main()
{
    int a[10][10],i,j,n,sum=0;
    printf("\nEnter the order of the square matrix A :-");
    scanf("%d",&n);
    printf("\n Enter the order of the square matrix A\n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    for(i=0;i<n;i++)
        for(j=0;j<=n;j++)
            if(i==j)
                sum=sum+a[i][j];
    printf("\nsum of digonal values=%d",sum);
    return 0;
}
```

INPUT/OUTPUT

RUN-1

Enter the order of the square matrix A :- 3

Enter the values of matrix A

3 4 5

6 8 9

1 3 4

sum of digonal values=15

RUN-2

Enter the order of the square matrix A :- 6

Enter the values of matrix A

1 2 3 4 5 6

7 8 9 3 5 3

4 5 6 7 8 1

1 2 3 4 1 2

1 1 1 1 1 1

3 4 5 6 6 7

sum of digonal values=22