DRONACHARYA
GROUP OF INSTITUTIONS

PROGRAMMING FOR PROBLEM SOLVING LAB

SUBJECT CODE: BCS- 251

B.TECH.(First Year Applied Science)
SEMESTER - I / II

Academic Session: 2024-25

Student Name:	
Roll. No.:	
Branch/Section:	

Dronacharya Group of Institutions

Plot No. 27, Knowledge Park-3, Greater Noida, Uttar Pradesh 201308

Affiliated to

Dr. A P J Abdul Kalam Technical University

Lucknow, Uttar Pradesh 226031

Table of Contents

1. Vision and Mission of the Institute.
2. Vision and Mission of the Department.
3. Program Educational Objectives (PEOs).
4. Program Outcomes (POs).
5. University Syllabus.
6. Course Outcomes (COs).
7. Course Overview.
8. List of Experiments mapped with COs.
9. DO's and DON'Ts.
10. General Safety Precautions.
11. Guidelines for students for report preparation.
12. Lab Experiments.

Vision of the Institute

"Instilling core human values and facilitating competence to address global challenges by providing Quality Technical Education."

Mission of the Institute

M1 - Enhancing technical expertise through innovative research and education, fostering creativity and excellence in problem-solving.

M2 - Cultivating a culture of ethical innovation and user-focused design, ensuring technological progress enhances the well-being of society.

M3 - Equipping individuals with the technical skills and ethical values to lead and innovate responsibly in an ever-evolving digital landscape.

Vision of Applied Science Department

- To inculcate a strong foundation in budding technocrats in the field of basic sciences and technology empowering them to learn engineering better and contribute to make a better world.

Mission of Applied Science Department

M1: To provide a strong foundation of knowledge and practical skills enabling technocrats to utilize scientific principles to give solutions to complex engineering problems.

M2: To guide students towards self-directed learning, self-discipline, and active engagement through innovative teaching and learning approaches.

M3: To inculcate values and ethics in students and make them responsible citizens of India.

Programme Educational Objectives (PEOs)

PEO1: Students basic concepts in applied science will be enhanced that is necessary for success in industry or in engineering practices as well as advanced study.

PEO2: Students will be equipped with problem-solving, laboratory, and design skills essential for technical careers focused on addressing critical challenges.

PEO3: Students will possess the ability to maintain the environmental serenity while adapting to the dynamic changes in the industry

Program Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze

complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

PO 9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply theseto one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

University Syllabus

SR.NO	Name Of Practicals
1.	WAP that accepts the marks of 5 subjects and finds the sum and percentage marks obtained by the student.
2.	WAP that calculates the Simple Interest and Compound Interest. The Principal, Amount, Rate of Interest and Time are entered through the keyboard.
3.	WAP to calculate the area and circumference of a circle.
4.	WAP that accepts the temperature in Centigrade and converts into Fahrenheit using the formula C/5=(F-32)/9.
5.	WAP that swaps values of two variables using a third variable
6.	WAP that checks whether the two numbers entered by the user are equal or not.
7.	WAP to find the greatest of three numbers.
8.	WAP that finds whether a given number is even or odd.
9.	WAP that tells whether a given year is a leap year or not.
10.	WAP that accepts marks of five subjects and finds percentage and prints grades according to the following criteria: Between 90-100% -----Print 'A' 80-90% ---------- Print 'B' 60-80% ---------- Print 'C' Below 60%----------Print 'D'
11.	WAP that takes two operands and one operator from the user, perform the operation, and prints the result by using Switch statement
12.	WAP to print the sum of all numbers up to a given number.
13.	WAP to find the factorial of a given number.
14,	WAP to print sum of even and odd numbers from 1 to N numbers.
15.	WAP to print the Fibonacci series.
16.	WAP to check whether the entered number is prime or not
17.	WAP to find the sum of digits of the entered number.
18.	WAP to find the reverse of a number.
19.	WAP to print Armstrong numbers from 1 to 100.
20.	WAP to convert binary number into decimal number and vice versa.
21.	WAP that simply takes elements of the array from the user and finds the sum of these elements.
22.	WAP that inputs two arrays and saves sum of corresponding elements of these arrays in a third array and prints them.
23.	WAP to find the minimum and maximum element of the array.
24.	WAP to search an element in a array using Linear Search.
25.	WAP to sort the elements of the array in ascending order using Bubble Sort technique.
26.	WAP to add and multiply two matrices of order nxn.
27.	WAP that finds the sum of diagonal elements of a mxn matrix
28.	WAP to implement strlen (), strcat (),strcpy () using the concept of Functions.
29.	.Define a structure data type TRAIN_INFO. The type contain Train No.: integer type Train name: string Departure Time: aggregate type TIME Arrival Time: aggregate type TIME Start station: string End station: string The structure type Time contains two integer members: hour and minute. Maintain a train timetable and implement the following operations: a. List all the trains (sorted according to train number) that depart

	from a particular section. b. List all the trains that depart from a particular station at a particular time. c. List all he trains that depart from a particular station within the next one hour of a given time. d. List all the trains between a pair of start station and end station.
30.	WAP to swap two elements using the concept of pointers.
31.	WAP to compare the contents of two files and determine whether they are same or not.
32.	WAP to check whether a given word exists in a file or not. If yes then find the number of times it occurs.

Course Outcomes

CO1	Able to implement the algorithms and draw flowcharts for solving Mathematical and Engineering problems.
CO2	Demonstrate an understanding of computer programming language concepts.
CO3	Ability to design and develop Computer programs, analyzes, and interprets the concept of pointers, declarations, initialization, operations on pointers and their usage.
CO4	Able to define data types and use them in simple data processing applications he/she must be able to use the concept of array of structures.
CO5	Develop confidence for self-education and ability for life-long learning needed for Computer language.

Course Overview

In C programming lab we provides a comprehensive introduction to the ANSI C language, emphasizing portability and structured design. Students are introduced to all major language elements including fundamental data types, flow control, and standard function libraries. Thorough treatment is given to the topics of string and character manipulation, dynamic memory allocation, standard I/O, macro definition, and the C runtime library. The C training course explains the use of aggregate structures, unions, and pointers early on so the students can practice extensively in the hands on labs. Structured programming constructs and varargs functions are also covered. Emphasis is given to the processing of command line arguments and environment variables so students will be able to write flexible, user-friendly programs.

List of Experiments mapped with COs

SR.NO	Name Of Practicals	CO
1.	WAP that accepts the marks of 5 subjects and finds the sum and percentage marks obtained by the student.	CO1
2.	WAP that calculates the Simple Interest and Compound Interest. The Principal, Amount, Rate of Interest and Time are entered through the keyboard.	CO1
3.	WAP to calculate the area and circumference of a circle.	CO1
4.	WAP that accepts the temperature in Centigrade and converts into Fahrenheit using the formula C/5=(F-32)/9.	CO1
5.	WAP that swaps values of two variables using a third variable	CO1
6.	WAP that checks whether the two numbers entered by the user are equal or not.	
7.	WAP to find the greatest of three numbers.	CO2
8.	WAP that finds whether a given number is even or odd.	CO2
9.	WAP that tells whether a given year is a leap year or not.	CO2
10.	WAP that accepts marks of five subjects and finds percentage and prints gradesaccording to the following criteria: Between 90-100%------Print 'A' 80-90%----------Print 'B' 60-80%----------Print 'C' Below 60% ---------- Print 'D'	CO1
11.	WAP that takes two operands and one operator from the user, perform the operation, and prints the result by using Switch statement	CO2
12.	WAP to print the sum of all numbers up to a given number.	CO2
13.	WAP to find the factorial of a given number.	CO2
14,	WAP to print sum of even and odd numbers from 1 to N numbers.	CO2
15.	WAP to print the Fibonacci series.	CO2
16.	WAP to check whether the entered number is prime or not	CO2
17.	WAP to find the sum of digits of the entered number.	CO2

18.	WAP to find the reverse of a number.	CO5
19.	WAP to print Armstrong numbers from 1 to 100.	CO5
20.	WAP to convert binary number into decimal number and vice versa.	CO5
21.	WAP that simply takes elements of the array from the user and finds the sum of these elements.	CO4
22.	WAP that inputs two arrays and saves sum of corresponding elements of these arrays in a third array and prints them.	CO4
23.	WAP to find the minimum and maximum element of the array.	CO4
24.	WAP to search an element in a array using Linear Search.	CO4
25.	WAP to sort the elements of the array in ascending order using Bubble Sort technique.	CO4
26.	WAP to add and multiply two matrices of order nxn.	C04
27.	WAP that finds the sum of diagonal elements of a mxn matrix	CO4
28.	WAP to implement strlen (), strcat (),strcpy () using the concept of Functions.	CO4
29.	.Define a structure data type TRAIN_INFO. The type contain Train No.: integer type Train name: string Departure Time: aggregate type TIME Arrival Time: aggregate typeTIME Start station: string End station: string The structure type Time contains two integer members: hour and minute. Maintain a train timetable and implement the following operations: a. List all the trains (sorted according to train number) that depart	CO4
	from a particular section. b. List all the trains that depart from a particular station at a particular time. c. List all he trains that depart from a particular station within the next one hour of a given time. d. List all the trains between a pair of start station and end station.	CO4

30.	WAP to swap two elements using the concept of pointers.	CO3
31.	WAP to compare the contents of two files and determine whether they are same or not.	CO3
32.	WAP to check whether a given word exists in a file or not. If yes then find the number of times it occurs.	CO3

DOs and DON'Ts

DOs

1. Login-on with your username and password.
2. Log off the Computer every time when you leave the Lab.
3. Arrange your chair properly when you are leaving the lab.
4. Put your bags in the designated area.
5. Ask permission to print.

DON'Ts

1. Do not share your username and password.
2. Do not remove or disconnect cables or hardware parts.
3. Do not personalize the computer setting.
4. Do not run programs that continue to execute after you log off.
5. Do not download or install any programs, games or music on computer in Lab.
6. Personal Internet use chat room for Instant Messaging (IM) and Sites is strictly prohibited.
7. No Internet gaming activities allowed.
8. Tea, Coffee, Water & Eatables are not allowed in the Computer Lab.

General Safety Precautions

Precautions (In case of Injury or Electric Shock)

1. To break the victim with live electric source, use an insulator such as firewood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.
2. Unplug the risk of faulty equipment. If the main circuit breaker is accessible, turn the circuit off.
3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.

Precautions (In case of Fire)

1. Turn the equipment off. If the power switch is not immediately accessible, take plug off.
2. If fire continues, try to curb the fire if possible, by using the fire extinguisher or by covering it with a heavy cloth if possible isolate the burning equipment from the other surrounding equipment.
3. Sound the fire alarm by activating the nearest alarm switch located in the hallway.

Guidelines to Students for Report Preparation

All students are required to maintain a record of the experiments conducted by them. Guidelines for its preparation are as follows:-

1) All files must contain a title page followed by an index page. ***The files will not be signed by the faculty without an entry in the index page.***

2) Student's Name, Roll number and date of conduction of experiment must be written on all pages.

3) For each experiment, the record must contain the following
   (i) Aim/Objective of the experiment
   (ii) Pre-experiment work (as given by the faculty)
   (iii) Lab assignment questions and their solutions
   (iv) Test Cases (if applicable to the course)
   (v) Results/ output

Note:
1. Students must bring their lab record along with them whenever they come for the lab.
2. Students must ensure that their lab record is regularly evaluated.

PROGRAM-1

WAP that accepts the marks of 5 subjects and finds the sum and percentage marksobtained by the student.

```
#include<stdio.h>
int main()
  {
        int hindi, english,
        science,math,computer,sum ;float per;
        printf("Enter marks of
        Hindi=");
        scanf("%d",&hindi);
        printf("Enter marks of
        English=");
        scanf("%d",&english);
        printf("Enter marks of
        Science=");
        scanf("%d",&science);
        printf("Enter marks of
        Math="); scanf("%d",&math);
        printf("Enter marks of
        Computer=");
        scanf("%d",&computer);

        sum=hindi+english+science+math+comp
        uter;printf("Sum of marks=%d",sum);

        per=(float)sum/5;
        printf("Percentage of marks=%f",per);

  return 0;
  }
```

Output:

```
Enter marks of Hindi=56
Enter marks of English=65
Enter marks of Science=78
Enter marks of Math=86
Enter marks of Computer=56

Sum of marks=341
Percentage of marks=68.199997
```

PROGRAM-2

WAP that calculates the Simple Interest and Compound Interest. The Principal,Amount, Rate of Interest and Time are entered through the keyboard.

```
#include<stdio.h>
int main()
  {
        float p, r, t, a, si, ci;
        printf("Enter
        Principle=");
        scanf("%f",&p);
        printf("Enter Rate=");
        scanf("%f",&r);
        printf("Enter Time=");
        scanf("%f",&t);

        si=(p*r*t)/100;

        printf("Simple
        Interest=%f",si);a =
        p*(pow((1 + r / 100), t));
        ci =  a - p;
        printf("\nCompound Interest=%f",ci);

      return 0;
  }
```

Output:

```
Enter Principle=100
Enter Rate=10
Enter Time=3
Simple Interest=30.000000
Compound Interest=33.100010
```

PROGRAM-3

WAP to calculate the area and circumference of a circle.

```
#include<stdio.h>
int main()
  {
        float  r,c,a;
        printf("Enter
        radius=");
        scanf("%f",&r);

        float pi=3.14;

        c=2*pi*r
        ;
        a=pi*r*r
        ;
        printf("\nCircumference of a
        circle=%f",c);printf("\nArea of a
        circle=%f",a);
   return 0;

  }
```

Output:

```
Enter radius=7

Circumference of a circle=43.960003
Area of a circle=153.860016
```

PROGRAM-4

WAP that accepts the temperature in Centigrade and converts into Fahrenheit usingthe formula C/5=(F-32)/9.

```
#include<stdio.h

>int main()
{
        float c,f;
        printf("Enter
        Centigrade=");
        scanf("%f",&c);
        f=(9*c)/5+32;

        printf("Fahrenheit=%f",
f);return 0;
}
```

Output:

```
Enter Centigrade=36
Fahrenheit=96.800003
```

PROGRAM-5

WAP that swaps values of two variables using a third variable

```
#include<stdio.h
>
#include<conio.
h>void main() {
 int a, b, tempvar;
 printf("enter the value of a:
 ");scanf("%d", & a);
 printf("enter the value of b:
 ");scanf("%d", & b);
 tempvar =
 a;a = b;
 b = tempvar;
 printf("After swapping \n");
 printf("value of a is : %d \n",
 a);printf("value of b is : %d
 ", b); getch();
}
```

Output 1:

```
Enter the value of a: 23
Enter the value of b: 43
After swapping
Value of a is : 43
Value of b is : 23
```

PROGRAM-6

WAP that checks whether the two numbers entered by the user are equal or not.

```
#include<stdio.h>
void main()
  {
        int a,b;
        printf("Enter
        a=");
        scanf("%d",&a);
        printf("Enter
        b=");
        scanf("%d",&b);

        if(a==b)
        {
              printf("\na and b are equal.");
        }
        else
        {
              printf("\na and b are not equal.");
        }
getch();

  }
```

Output:

```
Enter a=10
Enter b=12

a and b are not equal.
```

PROGRAM-7

WAP to find the greatest of three numbers

```
#include<stdio.h>

int main() {

    double n1, n2, n3;

    printf("Enter three different
    numbers: ");scanf("%lf %lf %lf",
    &n1, &n2, &n3);

    if (n1 >= n2 && n1 >= n3)
      printf("%.2f is the largest number.", n1);

    if (n2 >= n1 && n2 >= n3)
      printf("%.2f is the largest number.", n2);


    if (n3 >= n1 && n3 >= n2)
      printf("%.2f is the largest number.", n3);

    return 0;
  }
```

Output:

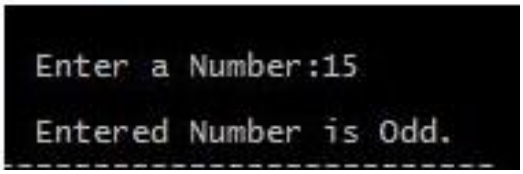
```
Enter the numbers A, B and C: 2 8 1
8 is the largest number.
```

PROGRAM-8

WAP that finds whether a given number is even or odd.

```
# include < stdio.h >
int main( )
   {
   int num ;
   printf(" Enter a Number :
   ") ;scanf("%d ",& num) ;
   if ( num  % 2 == 0 )
   {
   printf("\n Entered Number is Even.") ;
   }
   else
   {
   printf("\n Entered Number is Odd.") ;
   }
   return 0 ;
   }
```

Output of Program:



PROGRAM-9

WAP that tells whether a given year is a leap year or not

```
#include<stdio.h>
void main()
  {
        int  year;
        printf("Enter a
        number=");
        scanf("%d",&year);

        if((year%4==0||year%400==0)&&year%100!=0)
        {
              printf("\nYear is leap year.");
        }
        else
        {
              printf("\nYear is not leap year.");
        }
  getch()
  ;
  }
```

Output:

```
Enter a number=1600

Year is not leap year.
```

PROGRAM-10

WAP that accepts marks of five subjects and finds percentage and prints gradesaccording to the following criteria:

Between 90-100% ---- Print 'A'

80-90% --------- Print 'B'

60-80% ---------- Print 'C'

Below 60% ----- Print 'D'

```
#include<stdio.h>
void main()
  {
        int hindi, english,
        science,math,computer,sum ;float per;
        printf("Enter marks of
        Hindi=");
        scanf("%d",&hindi);
        printf("Enter marks of
        English=");
        scanf("%d",&english);
        printf("Enter marks of
        Science=");
        scanf("%d",&science);
        printf("Enter marks of
        Math="); scanf("%d",&math);
        printf("Enter marks of Computer=");
        scanf("%d",&computer);
        sum=hindi+english+science+math+comp
        uter;printf("\nSum of marks=%d",sum);
        per=(float)sum/5;
        printf("\nPercentage of
        marks=%f",per);
        if(per>=90&&per<=100)
        {
              printf("\nGrade A");
        }
        else if(per>=80&&per<90)
        {
              printf("\nGrade B");
        }
        else if(per>=60&&per<80)
        {
              printf("\nGrade C");
        }
        else if(per<60)
        {
              printf("\nGrade D");
        }
  getch();
```

}

Output:

```
Enter marks of Hindi=45
Enter marks of English=65
Enter marks of Science=89
Enter marks of Math=78
Enter marks of Computer=65

Sum of marks=342
Percentage of marks=68.400002
Grade C
```

PROGRAM-11

WAP that takes two operands and one operator from the user,perform the operation, and prints the result by using Switch statement

```
#include<stdio.h>
void main()
  {
        int choice,a,b;
        printf("Select your
        choice:\n");printf("1-
        Add:\n");
        printf("2- Sub:\n");
        printf("3- Mul:\n");
        printf("4- Div:\n");
        printf("5- Mod:\n");
        printf("Enter number a=");
        scanf("%d",&a);
        printf("Enter number b=");
        scanf("%d",&b);
        printf("Enter your
        choice=");
        scanf("%d",&choice);
        switch(choice)
        {
              case 1:
                    printf("Add of a and b
                    %d",(a+b));break;
              case 2:
                    printf("Sub of a and b %d",(a-
                    b));break;
              case 3:
                    printf("Mul of a and b
                    %d",(a*b));break;
              case 4:
                    printf("Div of a and b
                    %d",(a/b));break;
               case 5:
                    printf("Mod of a and b
                    %d",(a%b));break;
               default
                    printf("Wrong choice.");
                    :


        }
  return 0;
   }
```

Output:

```
Select your choice:
1- Add:
2- Sub:
3- Mul:
4- Div:
5- Mod:
Enter number a=7
Enter number b=5
Enter your choice=3
Mul of a and b 35
```

PROGRAM-12

WAP to print the sum of all numbers up to a given number.

```
#include <stdio.h>
void main() {
int i, num, sum = 0;
printf("Enter the number: ");
scanf("%d", &num);
for(i = 0; i <= num; i++) {
sum = sum + i;
}
printf("%d", sum);
getch();
}
```

Output

```
Enter a positive integer: 100
Sum = 5050
```

PROGRAM-13

WAP to find the factorial of a given number.

```
#include <stdio.h>
int main() {
      int n, i;
      long fact = 1;
      printf("Enter an integer:
      ");scanf("%d", &n);

      // shows error if the user enters a negative
      integerif (n < 0)
         printf("Error! Factorial of a negative number doesn't
      exist.");else {
         for (i = 1; i <= n; ++i)
            {fact *= i;
         }
         printf("Factorial of %d = %llu", n, fact);
      }

      return 0;
   }
```

Output

```
Enter an integer: 10
Factorial of 10 = 3628800
```

PROGRAM-14

WAP to print sum of even and odd numbers from 1 to N numbers.

```
#include <stdio.h>

void main()
{
   int i, num, odd_sum = 0, even_sum = 0;

   printf("Enter the value of num\n");
   scanf("%d", &num);
   for (i = 1; i <= num; i++)
   {
      if (i % 2 == 0)
         even_sum = even_sum + i;
      else
         odd_sum = odd_sum + i;
   }
   printf("Sum of all odd numbers = %d\n", odd_sum);
   printf("Sum of all even numbers = %d\n", even_sum);
}

getch();
}
```

Output

```
Please Enter the Maximum Limit Value : 10

The Sum of Even Numbers upto 10  = 30
The Sum of Odd Numbers upto 10  = 25
```

PROGRAM-15

WAP to print the Fibonacci series.

```
#include<stdio.h>
int main()
   {
    int n1=0,n2=1,n3,i,number;
    printf("Enter the number of
    elements:");scanf("%d",&number);
    printf("\n%d %d",n1,n2);//printing 0 and 1
    for(i=2;i<number;++i)//loop starts from 2 because 0 and 1 are already printed
    {
     n3=n1+n2;
     printf("
     %d",n3);
     n1=n2;
     n2=n3;
    }
     return 0;
    }
```

Output:

```
Enter the number of elements:15
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

PROGRAM-16

WAP to check whether the entered number is prime or not

```
#include<stdio.h>
int main(){
   int n,i,m=0,flag=0;
   printf("Enter the number to check
   prime:");scanf("%d",&n);
   m=n/2;
   for(i=2;i<=m;i+
   +)
   {
   if(n%i==0)
   {
   printf("Number is not
   prime");flag=1;
   break;
   }
   }
   if(flag==0)
   printf("Number is
   prime");return 0;
    }
```

Output:

```
Enter the number to check prime:56
Number is not prime

Enter the number to check prime:23
Number is prime
```

PROGRAM-17

WAP to find the sum of digits of the entered number

```
#include<stdio.h>
int main()
   {
   int n,sum=0,m;
   printf("Enter a
   number:");
   scanf("%d",&n);
   while(n>0)
   {
   m=n%10
   ;
   sum=sum+
   m;n=n/10;
   }
   printf("Sum
   is=%d",sum);return 0;
   }
```

Output

```
Enter a number:654
Sum is=15


Enter a number:123
Sum is=6
```

PROGRAM-18

WAP to find the reverse of a number.

```
#include<stdio.h>
 int main()
{
int n, reverse=0, rem;
printf("Enter a number: ");
scanf("%d", &n);
while(n!=0)
  {
     rem=n%10;
     reverse=reverse*10+rem;
     n/=10;
  }
  printf("Reversed Number: %d",reverse);
return 0;
}
```

Output:

```
Enter a number: 123
Reversed Number: 321
```

PROGRAM-19

WAP to print Armstrong numbers from 1 to 100.

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

int main()
{
   int i, sum, num, count = 0;
   printf(
      "All Armstrong number between 1 and 1000 are:\n");
   for (i = 1; i <= 1000; i++) {
      num = i;

      while (num != 0) {
         num /= 10;
         count++;
      }
      num = i;
      sum = pow(num % 10, count)
            + pow((num % 100 - num % 10) / 10, count)
            + pow((num % 1000 - num % 100) / 100, count);
      if (sum == i) {
         printf("%d ", i);
      }
      count = 0;
   }
return 0;
}
```

Output

```
All Armstrong number between 1 and 1000 are:
1 2 3 4 5 6 7 8 9 153 370 371 407
```

PROGRAM-20

WAP to convert binary number into decimal number and vice versa.

```
#include  <stdio.h>
#include <conio.h>
void main()
{
   // declaration of variables
   int num, binary_num, decimal_num = 0, base = 1, rem;
   printf (" Enter a binary number with the combination of 0s and 1s \n");
   scanf (" %d", &num); // accept the binary number (0s and 1s)

   binary_num = num; // assign the binary number to the binary_num variable


   while ( num > 0)
   {
      rem = num % 10; /* divide the binary number by 10 and store the remainder in rem vari
able. */
      decimal_num = decimal_num + rem * base;
      num = num / 10; // divide the number with quotient
      base = base * 2;
   }

   printf ( " The binary number is %d \t", binary_num); // print the binary number
   printf (" \n The decimal number is %d \t", decimal_num); // print the decimal
   getch();
}
```

Output

```
Enter a binary number with the combination of 0s and 1s
1101
 The binary number is 1101
 The decimal number is 13
```

PROGRAM-21

WAP that simply takes elements of the array from the user and finds the sum of theseelements.

```
#include<stdio.h>

int main()
   {

      int arr[100], size, i, sum = 0;

      printf("Enter array
      size\n");
      scanf("%d",&size);

      //Get all elements using for loop and store it in
      arrayprintf("Enter array elements\n");
      for(i = 0; i < size; i++)
          scanf("%d",&arr[i]
          );

      //add all elements to the variable
      sum.for(i = 0; i < size; i++)
          sum = sum + arr[i]; // same as sum += arr[i];

      //print the result
      printf("Sum of the array = %d\n",sum);

      return 0;
   }
```

Output:

```
Enter size of the array: 5
Enter elements in array: 7
8
9
4
5
sum of an array is:33
```

PROGRAM-22

WAP that inputs two arrays and saves sum of corresponding elements of these arrays in a third array and prints them.

```
#include

<stdio.h>int

main (void)
{
   int first_array [5], second_array [5], sum_array
   [5];int i;

   //Input elements of first array
   printf ("***Input elements of first
   array***\n");for (i = 0; i < 5; i++)
   {
      printf ("Input %d element: ", i +
      1);scanf ("%d", &first_array
      [i]);
   }

   //Input elements of second array
   printf ("\n***Input elements of second
   array***\n");for (i = 0; i < 5; i++)
   {
      printf ("Input %d element: ", i +
      1);scanf ("%d", &second_array
      [i]);
   }

   //Finding the sum of corresponding elements and save into third
   arrayfor (i = 0; i < 5; i++)
   {
      sum_array [i] = first_array [i] + second_array [i];
   }

     printf ("\n***Array having the sum of corresponding
elements***\n");for (i = 0; i < 5; i++)
   {
      printf ("%d  ", sum_array [i]);
   }

   return 0;
};
```

OUTPUT

```
***Input elements of first array***
Input 1 element: 23
Input 2 element: 64
Input 3 element: 39
Input 4 element: 56
Input 5 element: 38

***Input elements of second array***
Input 1 element: 36
Input 2 element: 96
Input 3 element: 43
Input 4 element: 62
Input 5 element: 47

***Array having the sum of corresponding elements***
59 160 82 118 85
```

PROGRAM-23

WAP to find the minimum and maximum element of the array.

```
#include <stdio.h>
void main()
  {
    int arr1[100];
    int i, mx, mn,
    n;
     printf("\n\nFind maximum and minimum element in an array
      :\n");printf("..... \n");
      printf("Input the number of elements to be stored in the
      array :"); scanf("%d",&n);

      printf("Input %d elements in the array
      :\n",n);for(i=0;i<n;i++)
          {
               printf("element - %d : ",i);
               scanf("%d",&arr1[i]);
             }


    mx =
    arr1[0];mn
    = arr1[0];

    for(i=1; i<n; i++)
    {
       if(arr1[i]>mx)
       {
          mx = arr1[i];
       }


       if(arr1[i]<mn)
       {
          mn = arr1[i];
       }
    }
    printf("Maximum element is : %d\n", mx);
    printf("Minimum element is : %d\n\n", mn);
  }
  getch();
  }
```

```
Find maximum and minimum element in an array :
-------------------------------------------------
Input the number of elements to be stored in the array :3
Input 3 elements in the array :
element - 0 : 45
element - 1 : 25
element - 2 : 21
Maximum element is : 45
Minimum element is : 21
```

PROGRAM-24

WAP to search an element in a array using Linear Search.

```
#include<stdio.h>
int main()
   {
   int a[10],i,n,key,f=0;
   printf("enter the size of array:
   ");scanf("%d",&n);
   printf("enter n elements into array:
   ");for(i=0;i<n;i++)
   {
   scanf("%d",&a[i]);
   }
   printf("enter the key element to be
   searched: ");scanf("%d",&key);
   for(i=0;i<n;i++)
   {
   if(a[i]==key)
   {
   f=1
   ;
   break;
   }
   }
   if(f==1)
   printf("the element
   found");else
   printf("element not found");
   }
   return 0;
   }
```

Output:

```
enter the size of array: 5
enter n elements into array: 54 16 59 51 18
enter the key element to be searched: 59
the element found

...Program finished with exit code 0
Press ENTER to exit console.
```

PROGRAM-25

WAP to sort the elements of the array in ascending order using Bubble Sort technique.

```
#include <stdio.h>
#define MAXSIZE 10

void main()
{
   int
   array[MAXSIZE];
   int i, j, num, temp;

   printf("Enter the value of num
   \n");scanf("%d", &num);
   printf("Enter the elements one by one
   \n");for (i = 0; i < num; i++)
   {
      scanf("%d", &array[i]);
   }
   printf("Input array is
   \n");for (i = 0; i < num;
   i++)
   {
      printf("%d\n", array[i]);
   }
   /* Bubble sorting begins
   */for (i = 0; i < num; i++)
   {
      for (j = 0; j < (num - i - 1); j++)
      {
         if (array[j] > array[j + 1])
         {
            temp = array[j];
            array[j] = array[j +
            1]; array[j + 1] =
            temp;}}}

   printf("Sorted array
   is...\n");for (i = 0; i <
   num; i++)
   {
      printf("%d\n", array[i]);
   }

getch();
}
```

OUTPUT

```
Enter number of elements
6
Enter 6 integers
2
-4
7
8
4
7
Sorted list in ascending order:
-4
2
4
7
7
8
```

PROGRAM-26

WAP to add and multiply two matrices of order nxn.

```
#include<stdio.h>
#include<stdlib.h>
int main(){
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
system("cls");
printf("enter the number of row=");
scanf("%d",&r);
printf("enter the number of column=");
scanf("%d",&c);
printf("enter the first matrix element=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("enter the second matrix element=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&b[i][j]);
}
}

printf("multiply of the matrix=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
mul[i][j]=0;
for(k=0;k<c;k++)
{
mul[i][j]+=a[i][k]*b[k][j];
}
}
}
//for printing result
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
printf("%d\t",mul[i][j]);
}
printf("\n");
```

```
}
return 0;
}
```

Output:

```
enter the number of row=3
enter the number of column=3
enter the first matrix element=
1 1 1
2 2 2
3 3 3
enter the second matrix element=
1 1 1
2 2 2
3 3 3
multiply of the matrix=
6 6 6
12 12 12
18 18 18
```

PROGRAM-27

WAP that finds the sum of diagonal elements of a mxn matrix

```
#include<stdio.h>
void main ()
      {
     static int array[10][10];
        int i, j, m, n, a = 0, sum = 0;

        printf("Enetr the order of the matix
        \n");scanf("%d %d", &m, &n);

        if (m == n )
        {
         printf("Enter the co-efficients of the
           matrix\n");for (i = 0; i < m; ++i)
           {
              for (j = 0; j < n; ++j)
              {
                 scanf("%d", &array[i][j]);
              }
           }

           printf("The given matrix is
           \n");for (i = 0; i < m; ++i)
           {
              for (j = 0; j < n; ++j)
              {
                 printf(" %d", array[i][j]);
              }
              printf("\n");
           }

           for (i = 0; i < m; ++i)
           {
              sum = sum +
              array[i][i]; a = a +
              array[i][m - i - 1];
           }

           printf("\nThe sum of the main diagonal elements is = %d\n",
           sum);printf("The sum of the off diagonal elements is  =
           %d\n", a);

          else
           printf("The given order is not square
   matrix\n");getch();
```

```
Enter the order of the matix
2 2
Enter the co-efficients of the matrix
40 30
38 90
The given matrix is
 40 30
 38 90

The sum of the main diagonal elements is = 130
The sum of the off diagonal elements is   = 68
```

```
}
```

PROGRAM -28

WAP to implement strlen (), strcat (),strcpy () using the concept of Functions.

```
#include<stdio.h>
#include<string.h>
int main(){
   char ch[20]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};
     printf("Length of string is:
    %d",strlen(ch));return 0;
   }
```

Output:
Length of string is: 10

```
#include<stdio.h>
#include
<string.h>int
main(){
  char ch[10]={'h', 'e', 'l', 'l', 'o',
    '\0'};char ch2[10]={'c', '\0'};
    strcat(ch,ch2);
    printf("Value of first string is:
   %s",ch);return 0;
   }
```

Output:
Value of first string is: helloc

```
#include<stdio.h>
#include
<string.h>int
main(){
 char ch[20]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};
    char ch2[20];
    strcpy(ch2,ch
    );
    printf("Value of second string is:
   %s",ch2);return 0;
   }
```

Output:
Value of second string is: javatpoint

PROGRAM -29

Define a structure data type TRAIN_INFO. The type contain Train No.: integer type Train name: string Departure Time: aggregate type TIME Arrival Time: aggregate type TIME Start station: string End station: string The structure type Time contains two integer members: hour and minute. Maintain a train timetable and implement the following operations: a. List all the trains (sorted according to train number) that depart from a particular section. b. List all the trains that depart from a particular station at a particular time. c. List all he trains that depart from a particular station within the next one hour of a given time. d. List all the trains between a pair of start station and end station.

```
#include<stdio.h>
#include<string.>
struct student
{  int roll;
   char name[50];
}st1,st2; //declaring variables for
structureint main( )
{
  //store first student
  informationst1.roll=101;
  strcpy(st1.name, "Amit");

 //store second student
  informationst2.roll=102;
  strcpy(st2.name, "Manoj");

  //printing first student information
  printf( "student 1 roll : %d\n",
  st1.roll);
  printf( "student 1 name : %s\n", st1.name);

  //printing second student information
  printf( "student 2 roll : %d\n", st2.roll);
  printf( "student 2 name : %s\n",
  st2.name);return 0;
}
```

Output:

```
student 1 roll : 101
student 1 name : Amit
student 2 roll : 102
student 2 name : Manoj
```

PROGRAM-30

WAP to swap two elements using the concept of pointers.

```
#include<stdio.h>

void swap(int*,

int*);

int main()
{
   int a, b;

   printf("Enter values for a and
   b\n");scanf("%d%d", &a, &b);

   printf("\n\nBefore swapping: a = %d and b = %d\n",

   a, b);swap(&a, &b);

   printf("\nAfter swapping: a = %d and b = %d\n", a, b);

   return 0;
}

void swap(int *x, int *y)
{
   int temp;

   temp = *x;
   *x  = *y;
   *y  = temp;
}
```

Output :
Enter values for a and b
100
200
Before swapping: a = 100 and b = 200
After swapping: a = 200 and b = 100

PROGRAM-31

WAP to compare the contents of two files and determine whether they are same or not

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

/* Function declaration */
int compareFile(FILE * fPtr1, FILE * fPtr2, int * line, int * col);


int main()
{
   /* File pointer to hold reference of input
   file */FILE * fPtr1;
   FILE * fPtr2;
   char
   path1[100];
   char
   path2[100];

   int diff;
   int line, col;


   /* Input path of files to
   compare */printf("Enter path of
   first file: "); scanf("%s", path1);
   printf("Enter path of second file:
   ");scanf("%s", path2);


   /* Open all files to compare
   */fPtr1  = fopen(path1,  "r");
   fPtr2 = fopen(path2, "r");

   /* fopen() return NULL if unable to open file in given
   mode. */if (fPtr1 == NULL || fPtr2 == NULL)
   {
      /* Unable to open file hence
      exit */printf("\nUnable to open
      file.\n");
      printf("Please check whether file exists and you have read
      privilege.\n");exit(EXIT_FAILURE);
```

```
    }


    /* Call function to compare file */
    diff = compareFile(fPtr1, fPtr2, &line, &col);

    if (diff == 0)
    {
       printf("\nBoth files are equal.");
    }
    else
    {
        printf("\nFiles are not equal.\n");
        printf("Line: %d, col: %d\n", line, col);
    }


    /* Finally close files to release resources */
    fclose(fPtr1);
    fclose(fPtr2);

    return 0;
}

int compareFile(FILE * fPtr1, FILE * fPtr2, int * line, int * col)
{
    char ch1, ch2;

    *line = 1;
    *col  = 0;

    do
    {
        // Input character from both files
        ch1 = fgetc(fPtr1);
        ch2 = fgetc(fPtr2);

        // Increment line
        if (ch1 == '\n')
        {
            *line += 1;
            *col = 0;
        }

        // If characters are not same then return -1
        if (ch1 != ch2)
            return -1;

        *col  += 1;

    } while (ch1 != EOF && ch2 != EOF);
```

```
    /* If both files have reached end */
    if (ch1 == EOF && ch2 == EOF)
        return 0;
    else
        return -1;
}
```

```
Enter path of first file: data\compare1.txt
Enter path of second file: data\compare2.txt

Files are not equal.
Line: 2, col: 14
```

PROGRAM-32

WAP to check whether a given word exists in a file or not. If yes then find the number of times it occurs.

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

        FILE*
        filePointer;int
        wordExist=0;
        int
        bufferLength=255;
        char search[100];
        printf("Enter word to be
        search=");scanf("%s",search);
        char line[bufferLength];
        filePointer = fopen("D:\\file.txt", "r");
        while(fgets(line, bufferLength,
        filePointer))
        {
                char *ptr = strstr(line,
                search);if (ptr != NULL)
                {
                        wordExist1;
                        break;
                }
        }
        fclose(filePointer
        );if
        (wordExist==1)
        {
                printf("Word exists.");
        }
        else
        {
                printf("Word doesn't exist.");
        }
  }
```

Output:

```
Enter word to be search=is
Word exists.
```

This lab manual has been updated by

Dr. Pawan Sharma

(hodaps@gnindia.dronacharya.info)

Crosschecked ByHOD APS

Verified By
Director, DGI Greater Noida

Please spare some time to provide your valuable feedback.