

# 'C' Programming Training Module

Classroom



20

## 'C' PROGRAMMING TRAINING MODULE

*The C programming training module will make the reader accustomed to C language. This material will help the reader in understanding the basic of the C language, C functions & the use of C language to build simple projects.*

---

<b>DELIVERY METHOD</b>	<ul style="list-style-type: none"><li>• 25% Self-paced Learning</li><li>• 75% Instructor led Training</li></ul>
<b>VERSION</b>	<ul style="list-style-type: none"><li>• 2020</li></ul>
<b>LEARNING OBJECTIVES</b>	<ul style="list-style-type: none"><li>• Illustrate features, application, errors &amp; structure of C program</li><li>• Identify keywords, identifier, constant &amp; Choose appropriate datatype</li><li>• Classification of operators, precedence &amp; associativity</li><li>• Use different Input/output statement, Escape sequences</li><li>• Type casting, simple C programs</li><li>• Contrast the use of simple if, if else statement</li><li>• Contrast the use of nested if else with programs</li><li>• Contrast the use of if else ladder with programs</li><li>• Criticize the limitation of conditional operator &amp; goto statement</li><li>• Examine the use switch statement &amp; write programs based on choice</li><li>• Write switch programs based on choices &amp; other programs</li><li>• Discuss use of for, while &amp; do while loop with syntax</li><li>• Simple &amp; complex while &amp; do while loop programs</li><li>• Compare while with Do While, distinguish break &amp; continue</li><li>• Determine the use of different type of pointer &amp; its importance</li><li>• Define Function &amp; Contrast the use of call by value &amp; call by reference</li><li>• Make programs based on modularizing approach</li><li>• Concept of storage class &amp; programs related to recursion</li><li>• Concept &amp; program related to 1d, 2d array</li><li>• Use of string &amp; string handling library functions</li></ul>

	<ul style="list-style-type: none"> <li>• Implementation of string handling library functions</li> <li>• Concept &amp; program related to structure</li> <li>• Concept of union &amp; enumerated data type</li> <li>• Appraise the usage of preprocessor directives like macro substitution, file inclusion etc.</li> <li>• Make use of different file handling library functions</li> <li>• Write simple &amp; complex programs related to file handling</li> <li>• Concepts &amp; dynamic memory allocation programs</li> <li>• Complex programs related to pointer &amp; DMA</li> <li>• Discuss the concept of data structure &amp; introduction to stack, queue &amp; linked list</li> <li>• Make use of command line argument while writing any c program</li> </ul>
<b>PREREQUISITES SKILLS</b>	<ul style="list-style-type: none"> <li>• Computer Science Fundamentals</li> <li>• Basic Knowledge of Applied Mathematics &amp; Algorithm</li> </ul>
<b>DURATION</b>	<ul style="list-style-type: none"> <li>• 30 Hours</li> </ul>
<b>SKILL LEVEL</b>	<ul style="list-style-type: none"> <li>• Basic-Intermediate</li> </ul>
<b>HARDWARE REQUIREMENTS</b>	<ul style="list-style-type: none"> <li>• Processor: 2 GHZ or Higher</li> <li>• GB RAM: 8 GB</li> <li>• GB DISK: 80 GB</li> <li>• Network Requirement: Yes</li> </ul>

**Notes:**

*The following unit and exercise durations are estimates and might not reflect every class experience. The estimates do not include the duration of optional exercises or sections. Student in this course use a dosbox, gcc or online compiler to perform the exercises.*



---

# COURSE AGENDA:MODULE-1

---

## Lecture I - Fundamentals of C Programming Language

**Duration:** 3 Hrs.

<b>Overview</b>	This unit explains basic concept of c language to develop simple programs & identify type of error while compilation & execution.
<b>Learning Objectives</b>	After completing this unit, you should be able to <ul style="list-style-type: none"><li>• Illustrate features, application, errors &amp; structure of C program</li><li>• Identify keywords, identifier, constant &amp; Choose appropriate datatype</li><li>• Classification of operators, precedence &amp; associativity</li><li>• Use different Input/output statement, Escape sequences</li><li>• Use type casting &amp; write simple C programs</li></ul>
<b>Lab Exercise</b>	<ul style="list-style-type: none"><li>• Write a program to swap two numbers without using third variable.</li><li>• Write a program to rotate value of three variables X, Y, Z using fourth variable such as assign value of Y to X, Z to Y, X to Z.</li><li>• Write a program to calculate the area of triangle using formula <math>at=vs(s-a)(s-b)(s-c)</math></li><li>• Basic salary of an employee is input through the keyboard. The DA is 25% of the basic salary while the HRA is 15% of the basic salary. Provident Fund is deducted at the rate of 10% of the gross salary (BS+DA+HRA). Program to calculate the Net Salary.</li><li>• Write a program to find whether no is even or odd, positive or negative using bitwise operator.</li></ul>



## Lecture II – Decision Making Statements in C Language

Duration: 3 Hrs.

<b>Overview</b>	This unit allows you to make a decision, based upon the result of a condition. & build logics to solve various problems.
<b>Learning Objectives</b>	After completing this unit, you should be able to <ul style="list-style-type: none"><li>• Contrast the use of simple if, if else statement</li><li>• Contrast the use of nested if else with programs</li><li>• Contrast the use of if else ladder with programs</li><li>• Criticize the limitation of conditional operator &amp; goto statement</li><li>• Examine the use switch statement &amp; write programs based on choice.</li></ul>
<b>Lab Exercise</b>	<ul style="list-style-type: none"><li>• Write a program to determine the roots of quadratic equation.</li><li>• Write a program to find the largest of three numbers using nested if else.</li><li>• Write a program to find the smallest of three numbers using conditional operator.</li><li>• Write a program to find whether the year is leap year or not, without using logical operator.</li><li>• Write a program to receive marks of physics, chemistry &amp; maths from user &amp; check its eligibility for course if<ol style="list-style-type: none"><li>a) Marks of physics &gt; 40</li><li>b) Marks of chemistry &gt; 50</li><li>c) Marks of math's &gt; 60</li><li>d) Total of physics &amp; math's marks &gt; 150</li></ol>or<ol style="list-style-type: none"><li>e) Total of three subjects marks &gt; 200</li></ol></li><li>• Write the program to find the quadrant for the given coordinates using if else ladder.</li><li>• Write a program to find the value of y for a particular value of n. The a, x, b, n is input by user<ol style="list-style-type: none"><li>if n=1 <math>y=ax\%b</math></li><li>if n=2 <math>y=ax^2+b^2</math></li><li>if n=3 <math>y=a-bx</math></li><li>if n=4 <math>y=a+x/b</math></li></ol></li></ul>



## Lecture III – Loop: While, Do-While, For Loop

**Duration:** 3 Hrs.

<b>Overview</b>	C loops execute a block of commands a specified number of times until a condition is met. In this chapter, you will learn about all the looping statements of C programming along with their use.
<b>Learning Objectives</b>	After completing this unit, you should be able to <ul style="list-style-type: none"><li>• Discuss use of for, while &amp; do while loop with syntax</li><li>• Write simple &amp; complex while &amp; do while loop programs</li><li>• Compare while with Do While, distinguish break &amp; continue</li></ul>
<b>Lab Exercise</b>	<ul style="list-style-type: none"><li>• Write a program to construct a Fibonacci series upto n terms.</li><li>• Write a program to find sum of a Fibonacci series up to 100 terms.</li><li>• Write a program to calculate the factorial for given number.</li><li>• Write a program to find the sum of digits for a given five digit number.</li><li>• Write a program to reverse the digits of a four digit number.</li><li>• Write a program to find whether the number is Armstrong number.</li><li>• Write a program to find whether the number is palindrome or not.</li></ul>



## Lecture IV – Nested Loop

Duration: 3 Hrs.

<b>Overview</b>	Nested loop means a loop statement inside another loop statement.
<b>Learning Objectives</b>	After completing this unit, you should be able to <ul style="list-style-type: none"><li>• Use of Loop inside loop</li><li>• Write simple &amp; complex loop programs for generating patterns</li><li>• Solving problem of <math>n^2</math> complexity</li></ul>
<b>Lab Exercise</b>	<ul style="list-style-type: none"><li>• Write a program to generate sum of series <math>1!+2!+3!+\dots+n!</math></li><li>• Write a program to find the sum of following series <math>1-X1/1!+X2/2!-\dots+Xn/n!</math>.</li><li>• Write C program to evaluate the sum of series <math>-14+34-54+\dots+n^4</math> terms.</li><li>• Write a program to print the entire prime no between 1 and 300.</li><li>• Write a program to print out all the Armstrong number between 100 and 500.</li><li>• Write a program to draw the following figure:  * * * * * *  3 2 1 2 1 1  * * * * * *  • Write a program to receive a five-digit no and display as like 24689: 2 4 6 8 9</li></ul>



## Lecture V – Function

Duration: 3 Hrs.

Overview	This unit gives explanation to divide larger program into modules called as Function.
Learning Objectives	After completing this unit, you should be able to <ul style="list-style-type: none"><li>• Define Function &amp; Contrast the use of call by value &amp; call by reference</li><li>• Make programs based on modularizing approach</li></ul>
Lab Exercise	<ul style="list-style-type: none"><li>• Write a function that return sum of all the odd digits of a given positive no entered through keyboard.</li><li>• Write a program to print area of rectangle using function &amp; return its value to main function.</li><li>• Write a program to calculate the factorial for given number using function.</li><li>• Write a program to find sum of Fibonacci series using function.</li><li>• Write factorial function &amp; use the function to find the sum of series <math>S=1!+2!+-----n!</math>.</li></ul>





## Lecture VI – Recursive Function

**Duration:** 3 Hrs.

<b>Overview</b>	This unit gives option for calling function itself inside function
<b>Learning Objectives</b>	After completing this unit, you should be able to <ul style="list-style-type: none"><li>• Define Recursive Function &amp; Using loop of function by calling itself again and again</li><li>• Describe the recursive function working</li></ul>
<b>Lab Exercise</b>	<ul style="list-style-type: none"><li>• Write a program to find the factorial of given number using recursion.</li><li>• Write a program to find the sum of digits of a 5 digit number using recursion.</li><li>• Write a program to construct a Fibonacci series upto n terms using recursion.</li><li>• Write a program to calculate the GCD of given numbers using recursion.</li></ul>



## Lecture VII – Array

Duration: 3 Hrs.

<b>Overview</b>	This Lecture gives the knowledge about grouping of similar data type
<b>Learning Objectives</b>	<p>After completing this unit, you should be able to</p> <ul style="list-style-type: none"><li>• Define list/set of similar data type</li><li>• Sorting numbers</li><li>• Defining 1D and 2D Array</li><li>• Working with matrices</li><li>• Performing different operation on matrix</li></ul>
<b>Lab Exercise</b>	<ul style="list-style-type: none"><li>• Write a program to search a specific number in list of array.</li><li>• Write program to sort the list of n integers using bubble sorting technique.</li><li>• Write a program to convert decimal number in to binary number.</li><li>• Write a program to convert binary number in to decimal number.</li><li>• Write a program to delete duplicate element in a list of 10 elements &amp; display it on screen.</li><li>• Write a program to merge two sorted array &amp; no element is repeated during merging.</li><li>• Write a program to find the largest number of 4X4 matrix &amp; also calculate the average of all.</li><li>• Write a program to evaluate the addition of diagonal elements of two square matrixes.</li><li>• Write a program to find the transpose of a given matrix &amp; check whether it is symmetric or not.</li><li>• Write a program to print the multiplication of two N*N (Square) matrix.</li></ul>



## Lecture VIII – String

Duration: 3 Hrs.

Overview	Array of character is known as String.
Learning Objectives	<p>After completing this unit, you should be able to</p> <ul style="list-style-type: none"><li>• Learn different operation on string</li><li>• Use Predefined function on string as well implement it in your own algorithm</li></ul>
Lab Exercise	<ul style="list-style-type: none"><li>• "Write the implementation of the following string handling functions a) strlen() b) strcpy() c) strcat() d) strcmp() e) strrev()"</li><li>• Write a program in C to check whether the given string is a palindrome or not.</li><li>• Write program to sort the array of character (String) in alphabetical order like STRING in GINRST.</li><li>• Write a program to remove all the blank space from the string &amp; print it, also count the no of characters.</li><li>• Write a program to store the following string "zero", "one" -----"five". Print the no in words, given in figure as 3205.</li></ul>



## Lecture IX – Structure & File Handling

Duration: 3 Hrs.

<b>Overview</b>	This unit gives explanation to build a small software project using file handling & to select the appropriate derived datatype like structure.
<b>Learning Objectives</b>	After completing this unit, you should be able to <ul style="list-style-type: none"><li>• Appraise the usage of preprocessor directives like macro substitution, file inclusion etc</li><li>• Make use of different file handling library functions</li><li>• Write simple &amp; complex programs related to file handling</li><li>• Understand concept &amp; write program related to structure</li><li>• Understand concept of union &amp; enumerated data type</li></ul>
<b>Lab Exercise</b>	<ul style="list-style-type: none"><li>• Write a program to compare two given dates. To store a date uses a structure that contains three members namely day, month and year. If the dates are equal then display message equal otherwise unequal.</li><li>• Define a structure that can describe a hotel. It should have the member that includes the name, address, grade, room charge and number of rooms. Write a function to print out hotel of given grade in order of room charges.</li><li>• Define a structure called cricket with player name, team name, batting average, for 50 players &amp; 5 teams. Print team wise list contains names of player with their batting average.</li><li>• Write a c program to copy &amp; count the character content of one file says a.txt to another file b.txt.</li><li>• Write a program to take 10 integers from file and write square of these integer in other file.</li><li>• Write a program to read number from file and then write all 'odd' number to file ODD.txt &amp; all even to file EVEN.txt.</li><li>• Write a program to print all the prime number, between 1 to 100 in file prime.txt.</li></ul>



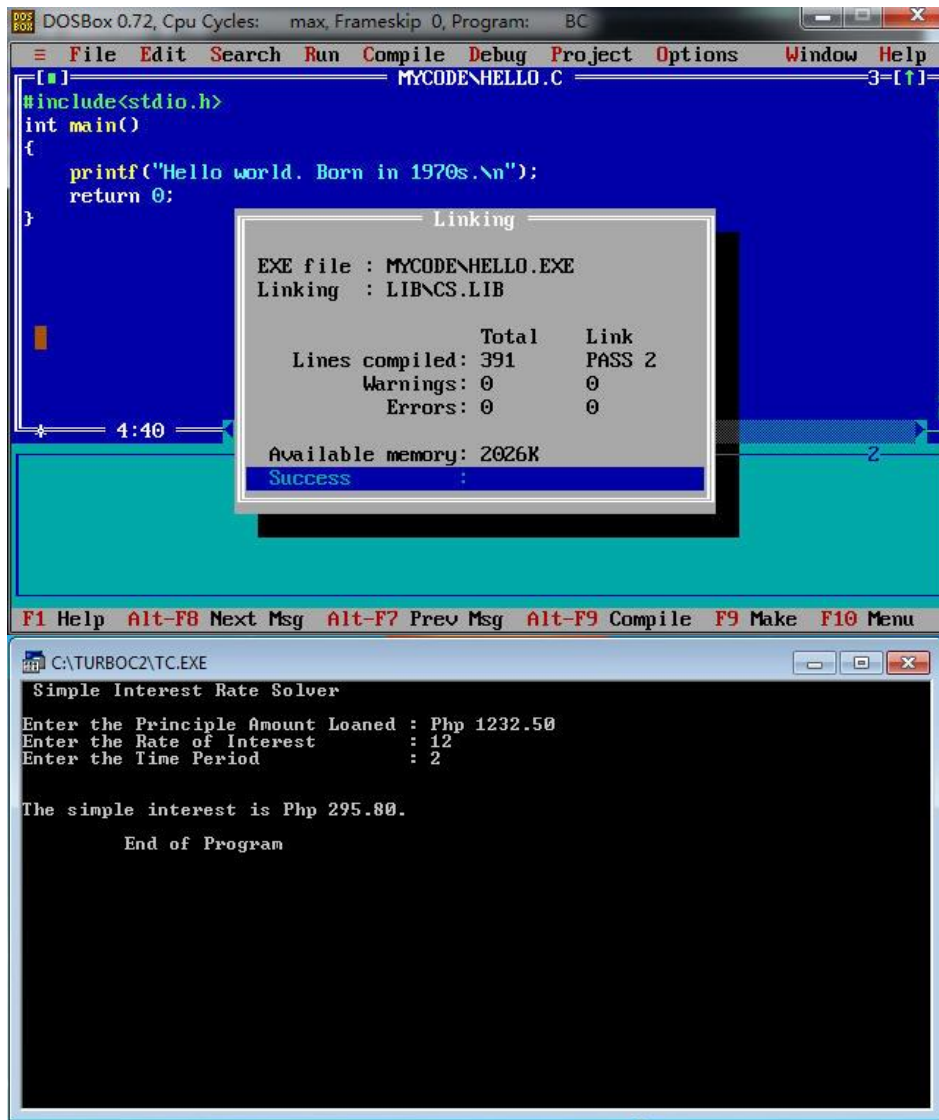
## Lecture X – Pointer, Dynamic Memory Allocation, Command Line Argument

Duration: 3 Hrs.

<b>Overview</b>	This unit gives explanation to solve problem using dynamic memory allocation technique & determine the role of pointer while writing complex c programs.
<b>Learning Objectives</b>	After completing this unit, you should be able to <ul style="list-style-type: none"><li>• Determine the use of different type of pointer &amp; its importance</li><li>• Understand concepts &amp; build dynamic memory allocation programs</li><li>• Write complex programs related to pointer &amp; DMA</li><li>• Make use of command line argument while writing any c program</li></ul>
<b>Lab Exercise</b>	<ul style="list-style-type: none"><li>• Write the following C program using pointer<ul style="list-style-type: none"><li>a) To sort the list of numbers through pointer</li><li>b) To reverse the string through pointer."</li></ul></li><li>• Write a C program to find the largest no among 20 integers array using dynamic memory allocation.</li><li>• Using Dynamic Memory Allocation, Write a C program to find the transpose of given matrix.</li><li>• Write a program to find the factorial of given number using command line argument.</li><li>• Write a program to find the sum of digits of a 5 digit number using command line argument.</li><li>• Write a program to construct a Fibonacci series up to n terms using command line argument.</li></ul>



## SAMPLE INTERFACE



```
DOSBox 0.72, Cpu Cycles: max, Frameskip 0, Program: BC
File Edit Search Run Compile Debug Project Options Window Help
[ ] MYCODENHELLO.C 3=[F1]
#include<stdio.h>
int main()
{
    printf("Hello world. Born in 1970s.\n");
    return 0;
}
Linking
EXE file : MYCODENHELLO.EXE
Linking : LIB\CS.LIB
Total Link
Lines compiled: 391 PASS 2
Warnings: 0 0
Errors: 0 0
Available memory: 2026K
Success :
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
C:\TURBOC2\TC.EXE
Simple Interest Rate Solver
Enter the Principle Amount Loaned : Php 1232.50
Enter the Rate of Interest : 12
Enter the Time Period : 2
The simple interest is Php 295.80.
End of Program
```

Sample Program Output



# PROGRAM SOLUTION

---

<b>Problem Statement</b>	Write a program to swap two numbers without using third variable.
<b>Concept</b>	<ul style="list-style-type: none"><li>Basic Concept of Swapping two number : <math>A = A + B</math> <math>B = A - B</math> <math>A = A - B</math></li></ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; void main() {     int a,b;     printf("Enter two Number : ");     scanf("%d %d",&amp;a,&amp;b);     a=a+b;     b=a-b;     a=a-b;     printf("A = ",a);     printf("B = ",b); }</pre>



<b>Problem Statement</b>	Write a program to rotate value of three variables X, Y, Z using fourth variable such as assign value of Y to X, Z to Y, X to Z.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take a temporary variable and save value of first variable in temp</li> <li>• Copy value of all the variable to its previous variable starting from second variable</li> <li>• Copy the value of first variable to last variable by temp.</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; void main() {     int x,y,z,temp;     printf("Enter three Number : ");     scanf("%d %d %d",&amp;x,&amp;y,&amp;z);     temp=x;     x=y;     y=z;     z=temp;     printf("x = ",x);     printf("y = ",y);     printf("z = ",z); }</pre>





<b>Problem Statement</b>	Write a program to calculate the area of triangle using formula $at=\sqrt{s(s-a)(s-b)(s-c)}$
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Use math.h library</li> <li>• <math>s=(a+b+c)/2</math></li> <li>• use sqrt function for square root</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; #include&lt;math.h&gt; int main() {     float a, b, c, Perimeter, s, Area;     printf("\nPlease Enter three sides of triangle\n");     scanf("%f%f%f",&amp;a,&amp;b,&amp;c);     Perimeter = a+b+c;     s = (a+b+c)/2;     Area = sqrt(s*(s-a)*(s-b)*(s-c));     printf("\n Perimeter of Traiangle = %.2f\n", Perimeter);     printf("\n Semi Perimeter of Traiangle = %.2f\n",s);     printf("\n Area of triangle = %.2f\n",Area);     return 0; }</pre>



<b>Problem Statement</b>	Basic salary of an employee is input through the keyboard. The DA is 25% of the basic salary while the HRA is 15% of the basic salary. Provident Fund is deducted at the rate of 10% of the gross salary (BS+DA+HRA). Program to calculate the Net Salary.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• BS input from user</li> <li>• <math>DA = 0.25 * BS</math></li> <li>• <math>HRA = 0.15 * BS</math></li> <li>• Gross Salary = BS+DA+HRA</li> <li>• <math>PF = 0.1 * \text{Gross Salary}</math></li> <li>• Net Salary = Gross Salary - PF</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; int main() {     float basic, da, hra, ta, gs;     float pf, it;     float net_salary;     printf("Enter Basic Salary : ");     scanf("%f", &amp;basic);     da = (basic*25)/100;     hra = (basic*15)/100;     gs = basic+da+hra;     pf = (gs*10)/100;     net_salary = gs-pf;     printf("Net Salary is: \$%.02f\n", net_salary);     return 0; } </pre>



<b>Problem Statement</b>	Write a program to find whether no is even or odd, positive or negative using bitwise operator.
<b>Concept</b>	<ul style="list-style-type: none"><li>• 5 = 101, 6 = 110 i.e. last bit is 1 then number is odd else number is even</li><li>• Similarly, first bit in signed integer denotes the sign i.e. if first bit is 1 then number is negative else number is positive</li></ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     int num;     printf("Enter any number: ");     scanf("%d", &amp;num);     if(num &amp; 1)     {         printf("%d is odd.", num);     }     else     {         printf("%d is even.", num);     }     return 0; }</pre>



<b>Problem Statement</b>	<b>Write a program to determine the roots of quadratic equation.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Find Discriminant, <math>D = b^2 - 4ac</math></li> <li>• Apply condition on D for finding roots</li> <li>• Case 1: <math>D &gt; 0</math>, roots are real and unequal</li> <li>• Case 2: <math>D = 0</math>, roots are real and equal</li> <li>• Case 3: <math>D &lt; 0</math>, roots are imaginary numbers</li> </ul>
<b>Code</b>	<pre>#include &lt;math.h&gt; #include &lt;stdio.h&gt; int main() {     double a, b, c, discriminant, root1, root2, realPart, imagPart;     printf("Enter coefficients a, b and c: ");     scanf("%lf %lf %lf", &amp;a, &amp;b, &amp;c);     discriminant = b * b - 4 * a * c;     if (discriminant &gt; 0) {         root1 = (-b + sqrt(discriminant)) / (2 * a);         root2 = (-b - sqrt(discriminant)) / (2 * a);         printf("root1 = %.2lf and root2 = %.2lf", root1, root2);     }     else if (discriminant == 0) {         root1 = root2 = -b / (2 * a);         printf("root1 = root2 = %.2lf;", root1);     }     else {         realPart = -b / (2 * a);         imagPart = sqrt(-discriminant) / (2 * a);         printf("root1 = %.2lf+%.2lfi and root2 = %.2lf-%.2fi", realPart, imagPart, realPart, imagPart);     }     return 0; }</pre>



<b>Problem Statement</b>	Write a program to find the largest of three numbers using nested if else.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• If A &gt; B then A &gt; C then A is Largest</li> <li>• If B &gt; A then B &gt; C then B is Largest</li> <li>• Else C is Largest</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     double n1, n2, n3;     printf("Enter three different numbers: ");     scanf("%lf %lf %lf", &amp;n1, &amp;n2, &amp;n3);     if (n1 &gt;= n2 &amp;&amp; n1 &gt;= n3)         printf("%.2f is the largest number.", n1);     if (n2 &gt;= n1 &amp;&amp; n2 &gt;= n3)         printf("%.2f is the largest number.", n2);     if (n3 &gt;= n1 &amp;&amp; n3 &gt;= n2)         printf("%.2f is the largest number.", n3);     return 0; }</pre>



<b>Problem Statement</b>	Write a program to find the smallest of three numbers using conditional operator.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• (condition) ? True Value : False Value</li> <li>• Min = (A &lt; B) ? ((C &lt; A) ? C : A) : ((C &lt; B) ? C : B)</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main(void) {     int a, b, c, temp, min;     printf ("Enter three nos. separated by spaces: ");     scanf ("%d%d%d", &amp;a, &amp;b, &amp;c);     temp = (a &lt; b) ? a : b;     min = (c &lt; temp) ? c : temp;     printf ("The Minimum of the three is: %d", min);     return 0; }</pre>



<b>Problem Statement</b>	Write a program to find whether the year is leap year or not, without using logical operator
<b>Concept</b>	<ul style="list-style-type: none"> <li>Year divisible by 4 and 400 is leap but year divisible by 100 and rest is not leap year</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     int year;     printf("\n Please Enter any year you wish \n ");     scanf(" %d ", &amp;year);     if (( year%400 == 0)    (( year%4 == 0 ) &amp;&amp;( year%100 != 0)))         printf("\n %d is a Leap Year. \n", year);     else         printf("\n %d is not the Leap Year. \n", year);     return 0; }</pre>



<b>Problem Statement</b>	Write a program to receive marks of physics, chemistry & maths from user & check its eligibility for course if a) Marks of physics > 40 b) Marks of chemistry > 50 c) Marks of math's > 60 d) Total of physics & math's marks > 150 or e) Total of three subjects marks > 200
<b>Concept</b>	<ul style="list-style-type: none"> <li>Use 'AND'(&amp;&amp;) and 'OR'(   ) for combining two or more condition in if statement.</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; void main() { int p,c,m,t,mp;   printf("Eligibility Criteria :\n");   printf("Marks in Maths &gt;=40\n");   printf("and Marks in Phy &gt;=50\n");   printf("and Marks in Chem&gt;=60\n");   printf("and Total in all three subject &gt;=150\n");   printf("or Total in Maths and Physics &gt;=200\n");   printf("-----\n");   printf("Input the marks obtained in Physics :");   scanf("%d",&amp;p);   printf("Input the marks obtained in Chemistry :");   scanf("%d",&amp;c);   printf("Input the marks obtained in Mathematics :");   scanf("%d",&amp;m);   printf("Total marks of Maths, Physics and Chemistry : %d\n",m+p+c);   printf("Total marks of Maths and Physics : %d\n",m+p);   if (m&gt;=40)     if(p&gt;=50)       if(c&gt;=60)         if((m+p+c)&gt;=150    (m+p)&gt;=200)           printf("The candidate is eligible for admission.\n");         else           printf("The candidate is not eligible.\n");       else         printf("The candidate is not eligible.\n");     else       printf("The candidate is not eligible.\n");   else     printf("The candidate is not eligible.\n"); } </pre>





<b>Problem Statement</b>	Write the program to find the quadrant for the given coordinates using if else ladder
<b>Concept</b>	<ul style="list-style-type: none"> <li>If X &amp; Y both are +ve then 1<sup>st</sup> quadrant else if X is -ve and Y is +ve then 2<sup>nd</sup> quadrant else if X &amp; Y both are -ve then 3<sup>rd</sup> quadrant else 4<sup>th</sup> quadrant</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; void main() {     int co1,co2;     printf("Input the values for X and Y coordinate : ");     scanf("%d %d",&amp;co1,&amp;co2);     if( co1 &gt; 0 &amp;&amp; co2 &gt; 0)         printf("The coordinate point (%d,%d) lies in the First quadrant.\n",co1,co2);     else if( co1 &lt; 0 &amp;&amp; co2 &gt; 0)         printf("The coordinate point (%d,%d) lies in the Second quadrant.\n",co1,co2);     else if( co1 &lt; 0 &amp;&amp; co2 &lt; 0)         printf("The coordinate point (%d, %d) lies in the Third quadrant.\n",co1,co2);     else if( co1 &gt; 0 &amp;&amp; co2 &lt; 0)         printf("The coordinate point (%d,%d) lies in the Fourth quadrant.\n",co1,co2);     else if( co1 == 0 &amp;&amp; co2 == 0)         printf("The coordinate point (%d,%d) lies at the origin.\n",co1,co2); }</pre>



<b>Problem Statement</b>	<p>Write a program to find the value of y for a particular value of n. The a, x, b, n is input by user</p> <p>a) if n=1 <math>y=ax\%b</math>  b) if n=2 <math>y=ax^2+b^2</math>  c) if n=3 <math>y=a-bx</math>  d) if n=4 <math>y=a+x/b</math></p>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Clears the concept of switch case</li> <li>• For Every possible input of n case statement are implemented</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     int n,a,b,x;     float y;     printf("Enter n: ");     scanf("%d", &amp;n);     printf("Enter a: ");     scanf("%d", &amp;a);     printf("Enter b: ");     scanf("%d", &amp;b);     printf("Enter x: ");     scanf("%d", &amp;x);     switch(n){ case 1: y=(a*x)%b; break; case 2: y=(a*x*x)+(b*b); break; case 3: y=a-(b*x); break; case 4: y=a+(x/b); break; } return 0; }</pre>



<b>Problem Statement</b>	<b>Write a program to construct a Fibonacci series upto n terms.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• A = -1 and B = 1</li> <li>• For loop or while loop upto n terms</li> <li>• Terms of Fibonacci series are generated by adding O/P = A + B and update value A = B and B = O/P in each iteration.</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     int i, n, t1 = 0, t2 = 1, nextTerm;     printf("Enter the number of terms: ");     scanf("%d", &amp;n);     printf("Fibonacci Series: ");     for (i = 1; i &lt;= n; ++i) {         printf("%d, ", t1);         nextTerm = t1 + t2;         t1 = t2;         t2 = nextTerm;     }     return 0; }</pre>



<b>Problem Statement</b>	<b>Write a program to find sum of a Fibonacci series up to 100 terms.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• A = -1, B = 1 and Sum = 0</li> <li>• For loop or while loop upto 100.</li> <li>• Terms of Fibonacci series are generated by adding O/P = A + B and update value A = B, B = O/P and Sum = Sum + O/P in each iteration.</li> <li>• Print Sum after completion of loop</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; #include &lt;math.h&gt; int main() {     int f1,f2,f3,n,i=2,s=1;     f1=0;     f2=1;     printf("How many terms do you \nwant in Fibonacci series? : ");     scanf("%d",&amp;n);     printf("\nFibonacci Series Upto %d Terms:\n\n",n);     printf("%d, %d",f1,f2);     while(i&lt;n)     {         f3=f1+f2;         printf(", %d",f3);         f1=f2;         f2=f3;         s=s+f3;         i++;     }     printf("\n\nSum of Fibonacci Series : %d",s);     return 0; } </pre>



<b>Problem Statement</b>	Write a program to calculate the factorial for given number.
<b>Concept</b>	<ul style="list-style-type: none"><li>• Factorial=1</li><li>• Initialize a loop from 1 to n</li><li>• Update Factorial =Factorial * loop value</li><li>• Print Factorial after completion of loop</li></ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     int c, n, f = 1;     printf("Enter a number to calculate its factorial\n");     scanf("%d", &amp;n);     for (c = 1; c &lt;= n; c++)         f = f * c;     printf("Factorial of %d = %d\n", n, f);     return 0; }</pre>



Problem Statement	Write a program to find the sum of digits for a given five digit number.
Concept	<ul style="list-style-type: none"> <li>• Input N</li> <li>• Sum =0</li> <li>• While loop until N become 0</li> <li>• In each iteration : temp=n%10 Sum = Sum + temp N = N /10</li> <li>• Print Sum After Loop</li> </ul>
Code	<pre>#include &lt;stdio.h&gt; int main() {     int n, t, sum = 0, remainder;     printf("Enter an integer\n");     scanf("%d", &amp;n);     t = n;     while (t != 0)     {         remainder = t % 10;         sum = sum + remainder;         t = t / 10;     }     printf("Sum of digits of %d = %d\n", n, sum); }</pre>



Problem Statement	Write a program to reverse the digits of a four digit number.
Concept	<ul style="list-style-type: none"> <li>• Input N</li> <li>• Rev =0</li> <li>• While loop until N become 0</li> <li>• In each iteration : temp=n%10 Rev = (Rev * 10) + temp N = N /10</li> <li>• Print Rev After Loop</li> </ul>
Code	<pre>#include&lt;stdio.h&gt; int main() {     int a,rev;     scanf("%d",&amp;a);     while(a&gt;=1)     {         rev = a%10;         printf("%d",rev);         a = a/10;     }     Printf("%d",rev); }</pre>



Problem Statement	Write a program to find whether the number is Armstrong number.
Concept	<ul style="list-style-type: none"> <li>• Input N</li> <li>• Val= N</li> <li>• Sum =0</li> <li>• While loop until N become 0</li> <li>• In each iteration : temp=n%10 Sum = Sum + temp^4 N = N /10</li> <li>• If Sum == Val then Armstrong number else not</li> </ul>
Code	<pre>#include &lt;stdio.h&gt; int main() {     int num, originalNum, remainder, result = 0;     printf("Enter a three-digit integer: ");     scanf("%d", &amp;num);     originalNum = num;     while (originalNum != 0) {         remainder = originalNum % 10;         result += remainder * remainder * remainder;         originalNum /= 10;     }     if (result == num)         printf("%d is an Armstrong number.", num);     else         printf("%d is not an Armstrong number.", num);     return 0; }</pre>





Problem Statement	Write a program to find whether the number is palindrome or not.
Concept	<ul style="list-style-type: none"> <li>• Input N</li> <li>• Val= N</li> <li>• Rev =0</li> <li>• While loop until N become 0</li> <li>• In each iteration :               <ul style="list-style-type: none"> <li>temp=n%10</li> <li>Rev = (Rev * 10) + temp</li> <li>N = N /10</li> </ul> </li> <li>• If Rev == Val then Palindrome number else not</li> </ul>
Code	<pre>#include &lt;stdio.h&gt; int main() {     int n, reversedN = 0, remainder, originalN;     printf("Enter an integer: ");     scanf("%d", &amp;n);     originalN = n;     while (n != 0) {         remainder = n % 10;         reversedN = reversedN * 10 + remainder;         n /= 10;     }     if (originalN == reversedN)         printf("%d is a palindrome.", originalN);     else         printf("%d is not a palindrome.", originalN);     return 0; }</pre>



Problem Statement	Write a program to generate sum of series $1!+2!+3!+\dots+n!$
Concept	<ul style="list-style-type: none"> <li>• Sum = 0</li> <li>• First for loop or while loop upto n</li> <li>• Nested loop inside first loop to calculate factorial of that loop value which we have already done and after each completion of inside loop sum = sum + Factorial of loop value returned</li> <li>• Print Sum after termination of first loop</li> </ul>
Code	<pre> #include&lt;stdio.h&gt; int main() {     int num,i,j,fact,sum=0;     printf("Enter the last number of series:\n");     scanf("%d",&amp;num);     for(i=1;i&lt;=num;i++)     {         fact=1;         if(i!=num)             printf("%d!+ ",i);         else             printf("%d!= ",i);         for(j=1;j&lt;=i;j++)             fact=fact*j;         sum=sum+fact;     }     printf("%d",sum);     return 0; } </pre>



<b>Problem Statement</b>	Write a program to find the sum of following series $1 - X^1/1! + X^2/2! - \dots + X^n/n!$ .
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Sum = 0</li> <li>• First for loop or while loop from 0 upto n</li> <li>• Nested loop inside first loop to calculate factorial of that loop value which we have already done and after each completion of inside loop <math>sum = sum + ((-1)^{loop\ value} * (X^{loop\ value}) / \text{Factorial of loop value})</math> returned</li> <li>• Here <math>-1^{loop\ value}</math> will generate +ve or -ve sign</li> <li>• Print Sum after termination of first loop</li> </ul>
<b>Code</b>	<pre> #include &lt;math.h&gt; #include &lt;stdio.h&gt; double Series(double x, int n) {     double sum = 1, term = 1, fct, j, y = 2, m;     int i;     for (i = 1; i &lt; n; i++) {         fct = 1;         for (j = 1; j &lt;= y; j++) {             fct = fct * j;         }         term = term * (-1);         m = term * pow(x, y) / fct;         sum = sum + m;         y += 2;     }     return sum; } int main() {     double x = 9;     int n = 10;     printf("%.4f", Series(x, n));     return 0; } </pre>



<b>Problem Statement</b>	Write C program to evaluate the sum of series $-1^4+3^4-5^4+\dots+n^4$ terms.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Sum = 0</li> <li>• K=1</li> <li>• For loop or while loop from 1 upto n with increment of 2</li> <li>• Sum = Sum + ((-1^K) * (loop value^4))</li> <li>• Update K =K+1</li> <li>• Here -1^K will generate +ve or -ve sign</li> <li>• Print Sum after termination of loop</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; int main() {     int num,i,j,fact,sum=0;     printf("Enter the last number of series:\n");     scanf("%d",&amp;num);     k=1;     for(i=1;i&lt;=num;i+=2)     {         sum=sum+((k*-1)*(i*i*i*i));         k++;     }     printf("%d",sum);     return 0; }</pre>



<b>Problem Statement</b>	<b>Write a program to print the entire prime no between 1 and 300.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• First for loop or while loop from 1 upto 300</li> <li>• Nested loop inside first loop to check for prime:  Loop from 2 to first loop value-1 to check the divisibility of first loop value  If any number between 2 to first loop value divides first loop value then its not prime else print that particular number as prime</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; main() { int number, div, ifprime; for (number=2;number&lt;=300;number++) { for (div=2; div&lt;number; div++) { if (number%div==0) { ifprime=0; break; } ifprime=1; } if (ifprime) { printf("\n%d", number); } } } }</pre>



<b>Problem Statement</b>	Write a program to print out all the Armstrong number between 100 and 500.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• First for loop or while loop from 100 upto 500</li> <li>• Nested loop inside first loop to check for Armstrong number which we have done already</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; int main(){     int num,r,sum,temp;     for(num=100;num&lt;=500;num++){         temp=num;         sum = 0;         while(temp!=0){             r=temp%10;             temp=temp/10;             sum=sum+(r*r*r);         }         if(sum==num)             printf("%d ",num);     }     return 0; }</pre>



<b>Problem Statement</b>	<p>Write a program to draw the following figure:</p> <pre> * * * * * *  3 2 1 2 1 1        *      * *     * * * </pre>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• For loop of i from 1 to n Nested loop of j from 1 to i Print *</li> <li>• For loop of i from n to 1 Nested loop of j from i to 1 Print j</li> <li>• For loop of i from 1 to n Print n-i spaces For loop of j from 1 to i Print '*&lt;space&gt;'</li> </ul>
<b>Code</b>	For Pattern Program try Nested Loop concept to create your own pattern as fun.



<b>Problem Statement</b>	Write a program to receive a five-digit no and display as like 24689: 2 4 6 8 9
<b>Concept</b>	<ul style="list-style-type: none"> <li>• First reverse the number</li> <li>• Loop while rev != 0  temp=rev%10  rev=rev/10  print temp</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; int main() { int num,rev,d; printf("Enter the last number of series:\n"); scanf("%d",&amp;num); rev=0; while(num!=0) { d=num%10; rev=rev*10+d; num=num/10; } while(rev!=0) { d=rev%10; printf("%d\n",d); rev=rev/10; } }</pre>





<b>Problem Statement</b>	Write a function that return sum of all the odd digits of a given positive no entered through keyboard.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a function which return integer and having one integer parameter</li> <li>• In function:  sum=0  Loop while parameter n != 0  temp = n % 10  if temp is odd then sum = sum + temp  after completion of loop return sum</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int sumodd(int); int main() {     int n, sum;     printf("Enter the length : ");     scanf("%d", &amp;n);     sum = sumodd(n);     printf("Sum of odd digit : %d", sum);     return 0; } int sumodd(int n) {     int d,s=0;     while(n!=0)     {         d=n%10;         if(d%2==0)             s=s+d;         n=n/10;     }     return s; }</pre>



<b>Problem Statement</b>	Write a program to print area of rectangle using function & return its value to main function.
<b>Concept</b>	<ul style="list-style-type: none"><li>• Create a function which return integer value and having two integer parameters L and B for length and breadth respectively</li><li>• In function return <math>L * B</math></li></ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int areaRectangle(int, int); int main() {     int l, b, area;     printf("Enter the length : ");     scanf("%d", &amp;l);     printf("Enter the width : ");     scanf("%d", &amp;b);     area = areaRectangle(l, b);     printf("The area of the rectangle : %d", area);     return 0; } int areaRectangle(int length, int width) {     return length * width; }</pre>



<b>Problem Statement</b>	<b>Write a program to calculate the factorial for given number using function.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a function which return a integer value and having only one parameter n as number whose factorial is to be find out</li> <li>• In function calculate factorial by loop as we have done in loop lecture</li> <li>• Return factorial calculated</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int fact(int); void main() {     int no,factorial;         printf("Enter a number to calculate it's factorial\n");         scanf("%d",&amp;no);         factorial=fact(no);         printf("Factorial of the num(%d) = %d\n",no,factorial); } int fact(int n) {     int i,f=1;     for(i=1;i&lt;=n;i++)     {         f=f*i;     }     return f; }</pre>



<b>Problem Statement</b>	<b>Write a program to find sum of Fibonacci series using function.</b>
<b>Concept</b>	<ul style="list-style-type: none"><li>• Create a function which return integer value and having only one parameter n as number of Fibonacci terms</li><li>• In function calculate sum of Fibonacci series by loop as we have done in loop lecture</li><li>• Return Sum calculated</li></ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main(void) {     int i, n, first = 0, second = 1, sum = 1, third;     printf (" Enter the range \n");     scanf( "%d", &amp;n);     for(i = 2; i &lt; n; i++)     {         third = first + second;         sum = sum + third;         first = second;         second = third;     }     printf("Sum of Fibonacci series for given range is %d", sum);     return 0; }</pre>



<b>Problem Statement</b>	Write factorial function & use the function to find the sum of series $S=1!+2!+\dots+n!$ .
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a factorial function as created in last question</li> <li>• In Main block: Sum =0 a for loop of i from 1 to n: Temp=call factorial function for i Sum=Sum + Temp</li> <li>• After Loop termination print Sum</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int fact(int); void main() {     int n,sum=0,i;         printf("Enter a terms\n");         scanf("%d",&amp;n); for(i=1,i&lt;=n;i++) { sum=sum+fact(i); }      printf("Sum of series(%d) = %d\n",n,sum); } int fact(int n) {     int i,f=1;     for(i=1;i&lt;=n;i++)     {         f=f*i;     }     return f; }</pre>



<b>Problem Statement</b>	<b>Write a program to find the factorial of given number using recursion.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a function of factorial with return type of integer and having only one parameter n</li> <li>• If n=0 then return 1 else return n * call factorial function with n-1 again</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; long int multiplyNumbers(int n); int main() {     int n;     printf("Enter a positive integer: ");     scanf("%d",&amp;n);     printf("Factorial of %d = %ld", n, multiplyNumbers(n));     return 0; } long int multiplyNumbers(int n) {     if (n&gt;=1)         return n*multiplyNumbers(n-1);     else         return 1; }</pre>

<b>Problem Statement</b>	<b>Write a program to find the sum of digits of a 5 digit number using recursion.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a function of Sum with return type of integer and having only one parameter n</li> <li>• If n=0 then return 0 else return n%10 + call sum function with n/10 again</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int sum_of_digit(int n) {     if (n == 0)         return 0;     return (n % 10 + sum_of_digit(n / 10)); } int main() {     int num = 12345;     int result = sum_of_digit(num);     printf("Sum of digits in %d is %d\n", num, result);     return 0; }</pre>



Problem Statement	Write a program to construct a Fibonacci series upto n terms using recursion.
Concept	<ul style="list-style-type: none"> <li>• Create a function of fibonacci with return type of void and having three parameter a,b,n</li> <li>• First call the function with 0,1,n</li> <li>• If n = 0 then it will stop and doesn't print anything else temp=a+b a=b b=temp and we again call fibonacci function with new arguments of (a,b,n-1)</li> </ul>
Code	<pre>#include &lt;stdio.h&gt; int fibo(int); int main() {     int num;     int result;     printf("Enter the nth number in fibonacci series: ");     scanf("%d", &amp;num);     if (num &lt; 0)     {         printf("Fibonacci of negative number is not possible.\n");     }     else     {         result = fibo(num);         printf("The %d number in fibonacci series is %d\n", num, result);     }     return 0; } int fibo(int num) {     if (num == 0)     {         return 0;     }     else if (num == 1)     {         return 1;     }     else     {         return(fibo(num - 1) + fibo(num - 2));     } }}</pre>



<b>Problem Statement</b>	<b>Write a program to calculate the GCD of given numbers using recursion.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a function of hcf with return type of integer and having two parameter a and b</li> <li>• If b = 0 then it will return a</li> <li>• Else function hcf is called recursively giving arguments(b,a%b)</li> <li>• a=b</li> <li>• b=temp</li> <li>• and we again call fibonacci function with</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int hcf(int n1, int n2); int main() {     int n1, n2;     printf("Enter two positive integers: ");     scanf("%d %d", &amp;n1, &amp;n2);     printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));     return 0; } int hcf(int n1, int n2) {     if (n2 != 0)         return hcf(n2, n1 % n2);     else         return n1; }</pre>





<b>Problem Statement</b>	<b>Write a program to search a specific number in list of array.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Insert an array</li> <li>• For loop of i from 0 to length of array: If array[i] == searching object then print i as location and break the loop otherwise after loop completion print not found</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; #define MAX_SIZE 100 int main() {     int arr[MAX_SIZE];     int size, i, toSearch, found;     printf("Enter size of array: ");     scanf("%d", &amp;size);     printf("Enter elements in array: ");     for(i=0; i&lt;size; i++)     {         scanf("%d", &amp;arr[i]);     }     printf("\nEnter element to search: ");     scanf("%d", &amp;toSearch);     found = 0;     for(i=0; i&lt;size; i++)     {         if(arr[i] == toSearch)         {             found = 1;             break;         }     }     if(found == 1)     {         printf("\n%d is found at position %d", toSearch, i + 1);     }     else     {         printf("\n%d is not found in the array", toSearch);     }     return 0; }</pre>



<b>Problem Statement</b>	<b>Write program to sort the list of n integers using bubble sorting technique.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     int array[100], n, c, d, swap;     printf("Enter number of elements\n");     scanf("%d", &amp;n);     printf("Enter %d integers\n", n);     for (c = 0; c &lt; n; c++)         scanf("%d", &amp;array[c]);     for (c = 0; c &lt; n - 1; c++)     {         for (d = 0; d &lt; n - c - 1; d++)         {             if (array[d] &gt; array[d+1]) /* For decreasing order use &lt; */             {                 swap = array[d];                 array[d] = array[d+1];                 array[d+1] = swap;             }         }     }     printf("Sorted list in ascending order:\n");     for (c = 0; c &lt; n; c++)         printf("%d\n", array[c]);     return 0; }</pre>



Problem Statement	Write a program to convert decimal number in to binary number.
Concept	<ul style="list-style-type: none"> <li>• <math>d_{n-1} \dots d_3 d_2 d_1 d_0</math> The decimal number is equal to the sum of binary digits (<math>d_n</math>) times their power of 2 (<math>2^n</math>): decimal = <math>d_0 \times 2^0 + d_1 \times 2^1 + d_2 \times 2^2 + \dots</math></li> <li>• And Vice Versa</li> </ul>
Code	<pre>#include &lt;math.h&gt; #include &lt;stdio.h&gt; long long convert(int n); int main() {     int n;     printf("Enter a decimal number: ");     scanf("%d", &amp;n);     printf("%d in decimal = %lld in binary", n, convert(n));     return 0; } long long convert(int n) {     long long bin = 0;     int rem, i = 1, step = 1;     while (n != 0) {         rem = n % 2;         printf("Step %d: %d/2, Remainder = %d, Quotient = %d\n", step++, n, rem, n / 2);         n /= 2;         bin += rem * i;         i *= 10;     }     return bin; }</pre>



Problem Statement	Write a program to convert binary number in to decimal number.
Concept	<ul style="list-style-type: none"> <li>• <math>d_{n-1} \dots d_3 d_2 d_1 d_0</math> The decimal number is equal to the sum of binary digits (<math>d_n</math>) times their power of 2 (<math>2^n</math>): decimal = <math>d_0 \times 2^0 + d_1 \times 2^1 + d_2 \times 2^2 + \dots</math></li> <li>• And Vice Versa</li> </ul>
Code	<pre>#include &lt;math.h&gt; #include &lt;stdio.h&gt; int convert(long long n); int main() {     long long n;     printf("Enter a binary number: ");     scanf("%lld", &amp;n);     printf("%lld in binary = %d in decimal", n, convert(n));     return 0; } int convert(long long n) {     int dec = 0, i = 0, rem;     while (n != 0) {         rem = n % 10;         n /= 10;         dec += rem * pow(2, i);         ++i;     }     return dec; }</pre>



<b>Problem Statement</b>	Write a program to delete duplicate element in a list of 10 elements & display it on screen.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• To find duplicate elements we need two loops.</li> <li>• First loop is used to select each element of array</li> <li>• Inner loop checks next subsequent elements for duplicates</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     int n, a[100], b[100], count = 0, c, d;     printf("Enter number of elements in array\n");     scanf("%d", &amp;n);     printf("Enter %d integers\n", n);     for (c = 0; c &lt; n; c++)         scanf("%d", &amp;a[c]);     for (c = 0; c &lt; n; c++)     {         for (d = 0; d &lt; count; d++)         {             if(a[c] == b[d])                 break;         }         if (d == count)         {             b[count] = a[c];             count++;         }     }     printf("Array obtained after removing duplicate elements:\n");     for (c = 0; c &lt; count; c++)         printf("%d\n", b[c]);     return 0; }</pre>



<b>Problem Statement</b>	Write a program to merge two sorted array & no element is repeated during merging.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take two sorted array let a and b</li> <li>• Take 3<sup>rd</sup> empty array say c</li> <li>• Now take three variables i,j,k for a,b,c respectively</li> <li>• Compare a[i]with b[j] and insert smallest of both in c[k] now only increase k and either of i or j depend on from which array smallest element came.</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; void main() {     int array1[50], array2[50], array3[100], m, n, i, j, k = 0;     printf("\n Enter size of array Array 1: ");     scanf("%d", &amp;m);     printf("\n Enter sorted elements of array 1: \n");     for (i = 0; i &lt; m; i++)     {         scanf("%d", &amp;array1[i]);     }     printf("\n Enter size of array 2: ");     scanf("%d", &amp;n);     printf("\n Enter sorted elements of array 2: \n");     for (i = 0; i &lt; n; i++)     {         scanf("%d", &amp;array2[i]);     }     i = 0;     j = 0;     while (i &lt; m &amp;&amp; j &lt; n)     {         if (array1[i] &lt; array2[j])         {             array3[k] = array1[i];             i++;         }         else         {             array3[k] = array2[j];             j++;         }         k++;     }     if (i &gt;= m)     {</pre>



```
while (j < n)
{
    array3[k] = array2[j];
    j++;
    k++;
}
}
if (j >= n)
{
    while (i < m)
    {
        array3[k] = array1[i];
        i++;
        k++;
    }
}
printf("\n After merging: \n");
for (i = 0; i < m + n; i++)
{
    printf("\n%d", array3[i]);
}
}
```



<b>Problem Statement</b>	Write a program to find the largest number of 4X4 matrix & also calculate the average of all.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take input from user</li> <li>• Set max = mat[0][0]</li> <li>• Run nested loop to traverse 2D matrix and compare every element of matrix with max, if element is greater than max then update max</li> <li>• Print max after loop</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; void main() {     int mat[4][4],i,j,max;     printf("Enter the elements of the matrix\n");     for(i=0;i&lt;4;i++)     {         for(j=0;j&lt;4;j++)         {             scanf("%d",&amp;mat[i][j]);         }     }     max=mat[0][0];     for(i=0;i&lt;4;i++)     {         for(j=0;j&lt;4;j++)         {             if(mat[i][j]&gt;max)                 max=mat[i][j];         }     }     print("%d",max); }</pre>





<b>Problem Statement</b>	Write a program to evaluate the addition of diagonal elements of two square matrixes.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take input from user</li> <li>• Sum=0</li> <li>• Run nested loop for matrix</li> <li>• If row==column then add its diagonal element, add it to sum</li> <li>• Print sum</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; void main() {     int mat[12][12];     int i,j,row,col,sum=0;     printf("Enter the number of rows and columns for 1st matrix\n");     scanf("%d%d",&amp;row,&amp;col);     printf("Enter the elements of the matrix\n");     for(i=0;i&lt;row;i++)     {         for(j=0;j&lt;col;j++)         {             scanf("%d",&amp;mat[i][j]);         }     }     printf("The matrix\n");     for(i=0;i&lt;row;i++)     {         for(j=0;j&lt;col;j++)         {             printf("%d\t",mat[i][j]);         }         printf("\n");     }     for(i=0;i&lt;row;i++)     {         for(j=0;j&lt;col;j++)         {             if(i==j)             {                 sum=sum+mat[i][j];             }         }     }      printf("The sum of diagonal elements of a square matrix = %d\n",sum); }</pre>



<b>Problem Statement</b>	<b>Write a program to find the transpose of a given matrix &amp; check whether it is symmetric or not.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take input from user for matrix a</li> <li>• Run nested loop of i,j for matrix</li> <li>• New matrix element <math>b[i][j] = a[j][i]</math></li> <li>• B is transpose matrix of a</li> <li>• Run nested loop of i,j for matrix</li> <li>• If <math>a[i][j]=b[i][j]</math> for each element then its symmetric else not</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; #define SIZE 3 int main() {     int A[SIZE][SIZE];     int B[SIZE][SIZE];     int row, col, isSymmetric;     printf("Enter elements in matrix of size 3x3: \n");     for(row=0; row&lt;SIZE; row++)     {         for(col=0; col&lt;SIZE; col++)         {             scanf("%d", &amp;A[row][col]);         }     }     for(row=0; row&lt;SIZE; row++)     {         for(col=0; col&lt;SIZE; col++)         {             B[row][col] = A[col][row];         }     }     isSymmetric = 1;     for(row=0; row&lt;SIZE &amp;&amp; isSymmetric; row++) </pre>



```
if(isSymmetric == 1)
{
    printf("\nThe given matrix is Symmetric matrix: \n");

    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            printf("%d ", A[row][col]);
        }
        printf("\n");
    }
}
else
{
    printf("\nThe given matrix is not Symmetric matrix.");
}
return 0;
}
```



<b>Problem Statement</b>	<b>Write a program to print the multiplication of two N*N (Square) matrix.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Matrix multiplication</li> <li>• <math>C[i][j]=C[i][j]+(A[i][k]*B[k][j])</math></li> <li>• Here k is 3<sup>rd</sup> nested loop after I J</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; #define N 4  void multiply(int mat1[][N], int mat2[][N], int res[][N]) {     int i, j, k;     for (i = 0; i &lt; N; i++)     {         for (j = 0; j &lt; N; j++)         {             res[i][j] = 0;             for (k = 0; k &lt; N; k++)                 res[i][j] += mat1[i][k]*mat2[k][j];         }     } }  int main() {     int mat1[N][N] = { {1, 1, 1, 1},                       {2, 2, 2, 2},                       {3, 3, 3, 3},                       {4, 4, 4, 4}};     int mat2[N][N] = { {1, 1, 1, 1},                       {2, 2, 2, 2},                       {3, 3, 3, 3},                       {4, 4, 4, 4}};     int res[N][N];     int i, j;     multiply(mat1, mat2, res);     printf("Result matrix is \n");     for (i = 0; i &lt; N; i++)     {         for (j = 0; j &lt; N; j++)             printf("%d ", res[i][j]);         printf("\n");     }     return 0; } </pre>



<b>Problem Statement</b>	<p>Write the implementation of the following string handling functions</p> <ol style="list-style-type: none"> <li>strlen()</li> <li>strcpy()</li> <li>strcat()</li> <li>strcmp()</li> <li>strrev()</li> </ol>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• strlen() find length of string</li> <li>• strcpy() copy from first variable to second</li> <li>• strcat() concat 2<sup>nd</sup> string to 1<sup>st</sup> string</li> <li>• strcmp() compare two string</li> <li>• strrev() reverses the given string</li> </ul>
<b>Code</b>	<pre> strlen()  #include &lt;stdio.h&gt; #include &lt;string.h&gt; int main( ) {     int len;     char array[20]="fresh2refresh.com" ;     len = strlen(array) ;     printf ( "\string length = %d \n" , len ) ;     return 0; }  strcpy()  #include &lt;stdio.h&gt; #include &lt;string.h&gt; int main( ) {     char source[ ] = "fresh2refresh" ;     char target[20]= "" ;     printf ( "\nsource string = %s", source ) ;     printf ( "\ntarget string = %s", target ) ;     strcpy ( target, source ) ;     printf ( "\ntarget string after strcpy( ) = %s", target ) ;     return 0; }  strcat()  #include &lt;stdio.h&gt; #include &lt;string.h&gt; </pre>



```

int main( )
{
    char source[ ] = " fresh2refresh" ;
    char target[ ]= " C tutorial" ;
    printf ( "\nSource string = %s", source ) ;
    printf ( "\nTarget string = %s", target ) ;
    strcat ( target, source ) ;
    printf ( "\nTarget string after strcat( ) = %s", target ) ;
}

```

strcmp()

```

#include <stdio.h>
#include <string.h>
int main( )
{
    char str1[ ] = "fresh" ;
    char str2[ ] = "refresh" ;
    int i, j, k ;
    i = strcmp ( str1, "fresh" ) ;
    j = strcmp ( str1, str2 ) ;
    k = strcmp ( str1, "f" ) ;
    printf ( "\n%d %d %d", i, j, k ) ;
    return 0;
}

```

strrev()

```

#include<stdio.h>
#include<string.h>
int main()
{
    char name[30] = "Hello";
    printf("String before strrev( ) : %s\n",name);
    printf("String after strrev( ) : %s",strrev(name));
    return 0;
}

```



<b>Problem Statement</b>	Write a program in C to check whether the given string is a palindrome or not.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take input string from user</li> <li>• Check for palindrome same as we used in number, the only difference in string is either check it by ascii value or use strcmp for comparing characters.</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; #include &lt;string.h&gt; void isPalindrome(char str[]) {     int l = 0;     int h = strlen(str) - 1;     while (h &gt; l)     {         if (str[l++] != str[h--])         {             printf("%s is Not Palindrome", str);             return;         }     }     printf("%s is palindrome", str); } int main() {     isPalindrome("abba");     isPalindrome("abbccbba");     isPalindrome("geeks");     return 0; } </pre>



<b>Problem Statement</b>	Write program to sort the array of character (String) in alphabetical order like STRING in GINRST.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take string input from user</li> <li>• Use ascii value to sort the element of string as we know all the character have their own ascii value</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; #include &lt;string.h&gt; int main() {     char ch, input[100], output[100];     int no[26] = {0}, n, c, t, x;     printf("Enter some text\n");     scanf("%s", input);     n = strlen(input);     for (c = 0; c &lt; n; c++)     {         ch = input[c] - 'a';         no[ch]++;     }     t = 0;     for (ch = 'a'; ch &lt;= 'z'; ch++)     {         x = ch - 'a';         for (c = 0; c &lt; no[x]; c++)         {             output[t] = ch;             t++;         }     }     output[t] = '\0';     printf("%s\n", output);     return 0; } </pre>





<b>Problem Statement</b>	Write a program to remove all the blank space from the string & print it, also count the no of characters.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take string as input</li> <li>• Set count =0</li> <li>• Initiate a loop on each character of string and increase value of count</li> <li>• For removing space traverse string, when space appear just shift rest content to its left</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     char text[1000], blank[1000];     int c = 0, d = 0;     printf("Enter some text\n");     gets(text);     while (text[c] != '\0') {         if (text[c] == ' ') {             int temp = c + 1;             if (text[temp] != '\0') {                 while (text[temp] == ' ' &amp;&amp; text[temp] != '\0') {                     if (text[temp] == ' ') {                         c++;                     }                     temp++;                 }             }         }         blank[d] = text[c];         c++;         d++;     }     blank[d] = '\0';     printf("Text after removing blanks\n%s\n", blank);     return 0; }</pre>



<b>Problem Statement</b>	Write a program to store the following string “zero”, “one” -----“five”. Print the no in words, given in figure as 3205.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Input Number from user</li> <li>• Extract each digit one by one from right side print their value in string by using switch case.</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; int main() {     int n, num = 0;     printf("Enter any number to print in words: ");     scanf("%d", &amp;n);     while(n != 0)     {         num = (num * 10) + (n % 10);         n /= 10;     }     while(num != 0)     {         switch(num % 10)         {             case 0:                 printf("Zero ");                 break;             case 1:                 printf("One ");                 break;             case 2:                 printf("Two ");                 break;             case 3:                 printf("Three ");                 break;             case 4:                 printf("Four ");                 break;             case 5:                 printf("Five ");                 break;             case 6:                 printf("Six ");                 break;             case 7:                 printf("Seven ");</pre>



```
        break;
    case 8:
        printf("Eight ");
        break;
    case 9:
        printf("Nine ");
        break;
    }
    num = num / 10;
}
return 0;
}
```



<b>Problem Statement</b>	Write a program to compare two given dates. To store a date uses a structure that contains three members namely day, month and year. If the dates are equal then display message equal otherwise unequal.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a structure with element of day, month, year</li> <li>• Take two Input from user</li> <li>• If day, month, year all three of date 1 is equal to respective day, month, year of date 2 then print equal date else print unequal.</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; struct date { int day; int month; int year; }; void main() { struct date d1,d2; clrscr(); printf("Enter first date(dd/mm/yyyy):"); scanf("%d%d%d",&amp;d1.day,&amp;d1.month,&amp;d1.year); printf("\nEnter second date(dd/mm/yyyy):"); scanf("%d%d%d",&amp;d2.day,&amp;d2.month,&amp;d2.year); if((d1.day==d2.day)&amp;&amp;(d1.month==d2.month)&amp;&amp;(d1.year==d2.year)) printf("\nEQUAL"); else printf("\nUNEQUAL"); getch(); }</pre>



<b>Problem Statement</b>	Define a structure that can describe a hotel. It should have the member that includes the name, address, grade, room charge and number of rooms. Write a function to print out hotel of given grade in order of room charges.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a structure with element of hotel name, address, grade, room charge , number of room</li> <li>• Take Input from user</li> <li>• Sort the array of structure by room charge</li> <li>• Take grade from user to display hotel</li> <li>• Display all hotel having grade == input grade</li> </ul>
<b>Code</b>	<pre> #include&lt;stdio.h&gt; #include&lt;conio.h&gt; main( ) { struct hotel { char name[20]; char city[10]; char grade; int rc,nr; }; struct hotel ht[20],t; int i,n,j,c; char gr; clrscr( ); printf("enter no. of hotelsn"); scanf("%d",&amp;n); for(i=0;i&lt;n;i++) { printf("enter name of hotel n"); scanf("%s",&amp;ht[i].name); printf("enter name of city n"); scanf("%s",&amp;ht[i].city); printf("enter the grade n"); scanf("%s",.ht[i].grade); ht[i].grade=getche( ); printf("enter room charge n"); scanf("%d",&amp;ht[i].rc); printf("enter no of rooms n"); scanf("%d",&amp;ht[i].nr); } for(i=0;i&lt;n;i++) for(j=0;j&lt;n-i;j++) { </pre>



```

t=ht[j];
ht[j]=ht[j+i];
ht[j+1]=t;
}
printf("enter a grade to print the hotels n");
gr=getche();
clrscr();
printf("hotel name city grade roomcharge no of room");
for(i=0;i<n;i++)
if(gr==ht[i].grade)
printf("%s %s %c %d %d",ht[i].name,ht[i].city,ht[i].grade,ht[i].r c,ht[i].nr);
getch();
printf("enter a room charge to print hotels less than given charge n");
scanf("%d",&c);
printf("hotel name city grade roomcharge no of rooms");
for(i=0;i<n;i++)
if(c<=ht[i].rc)
printf("%s %s %c %d %d",ht[i].name,ht[i].city,h[i].grade,ht[i].rc ,ht[i].nr);
}

```



<b>Problem Statement</b>	Define a structure called cricket with player name, team name, batting average, for 50 players & 5 teams. Print team wise list contains names of player with their batting average.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a structure with element of player name, team name and batting average</li> <li>• Take Input of 50 players from user</li> <li>• Sort the array of structure by team name</li> <li>• Print only player name and batting average of all element</li> </ul>
<b>Code</b>	<pre> #include&lt;stdio.h&gt; #include&lt;conio.h&gt; #include&lt;string.h&gt; struct cricket {     char P_Name[20];     char T_Name[20];     float B_Ave; }; void main() {     struct cricket s[5],t;     int i,j,n=5;     float p;     clrscr();     printf("Enter Data Of %d Player\n",n);     for(i=0;i&lt;n;i++)     {         printf("\nEnter Player Name,Team Name And Bating Average For Player %d :- \n",i+1);         scanf("%s %s %f",s[i].P_Name,s[i].T_Name,&amp;p);         s[i].B_Ave=p;     }     for(i=1;i&lt;=n-1;i++)     {         for(j=0;j&lt;=n-i;j++)         {             if(strcmp(s[j-1].T_Name,s[j].T_Name)&gt;0)             {                 t=s[j-1];                 s[j-1]=s[j];                 s[j]=t;             }         }     } } </pre>



```
printf("\nAfter Teamwise Sorting...Player List Is");
for(i=0;i<n;i++)
{
    printf("\n%-20s %-20s %.2f",s[i].P_Name,s[i].T_Name,s[i].B_Ave);
}
getch();
}
```





<b>Problem Statement</b>	Write a c program to copy & count the character content of one file says a.txt to another file b.txt.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Open 1<sup>st</sup> file in read mode and 2<sup>nd</sup> file in write mode</li> <li>• Scan content from 1<sup>st</sup> file and write it in 2<sup>nd</sup> file.</li> <li>• Close both files.</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; int main() {     FILE *fptr1, *fptr2;     char filename[100], c;     printf("Enter the filename to open for reading \n");     scanf("%s", filename);     fptr1 = fopen(filename, "r");     if (fptr1 == NULL)     {         printf("Cannot open file %s \n", filename);         exit(0);     }     printf("Enter the filename to open for writing \n");     scanf("%s", filename);     fptr2 = fopen(filename, "w");     if (fptr2 == NULL)     {         printf("Cannot open file %s \n", filename);         exit(0);     }     c = fgetc(fptr1);     while (c != EOF)     {         fputc(c, fptr2);         c = fgetc(fptr1);     }     printf("\nContents copied to %s", filename);     fclose(fptr1);     fclose(fptr2);     return 0; } </pre>



<b>Problem Statement</b>	Write a program to take 10 integers from file and write square of these integer in other file.
<b>Concept</b>	<ul style="list-style-type: none"><li>• Open 1<sup>st</sup> file in read mode and 2<sup>nd</sup> file in write mode</li><li>• Scan number from 1<sup>st</sup> file and write it in 2<sup>nd</sup> file by squaring it</li><li>• Close both files.</li></ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; void main() { FILE * f,*s; int a; f=fopen("abc.txt","r") s=fopen("xyz.txt","w") while((fscanf(f,"%d",&amp;a)!=EOF) {     fprintf(s,"%d",a*a); } fclose(f); fclose(s); }</pre>



<b>Problem Statement</b>	Write a program to read number from file and then write all 'odd' number to file ODD.txt & all even to file EVEN.txt.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Open main file in read mode</li> <li>• Open even_number file and odd_number file in write mode</li> <li>• Start scanning number in main file</li> <li>• If number % 2 is 0 then write that number in even_number file else write it in odd_number file</li> <li>• Close all the file properly.</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; int isEven(const int NUM); int isPrime(const int NUM); int main() {     FILE * fPtrIn,         * fPtrEven,         * fPtrOdd,         * fPtrPrime;     int num, success;     fPtrIn = fopen("data/numbers.txt", "r");     fPtrEven = fopen("data/even-numbers.txt", "w");     fPtrOdd = fopen("data/odd-numbers.txt", "w");     fPtrPrime = fopen("data/prime-numbers.txt", "w");     if(fPtrIn == NULL    fPtrEven == NULL    fPtrOdd == NULL    fPtrPrime ==     NULL)     {         printf("Unable to open file.\n");         printf("Please check whether file exists and you have read/write     privilege.\n");         exit(EXIT_FAILURE);     }     printf("File opened successfully. Reading integers from file. \n\n");     do     {         success = fscanf(fPtrIn, "%d", &amp;num);         if (isPrime(num))             fprintf(fPtrPrime, "%d\n", num);         else if (isEven(num))             fprintf(fPtrEven, "%d\n", num);         else             fprintf(fPtrOdd, "%d\n", num);     } while(success != -1);     fclose(fPtrIn);</pre>



```
fclose(fPtrEven);  
fclose(fPtrOdd);  
fclose(fPtrPrime);  
printf("Data written to files successfully.");  
return 0;  
}
```



<b>Problem Statement</b>	Write a program to print all the prime number, between 1 to 100 in file prime.txt.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Create a function to check prime number</li> <li>• Open file in write mode</li> <li>• Initiate a loop 1 to 100</li> <li>• Check every number for prime if it is prime then write it in file already open</li> <li>• After loop termination close the file properly to save</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; #include "fileopen.c" int is_prime(int n); int main () {     char fname[] = "primes.dat";     FILE *fp;     int m, n, i;     printf("Enter range : ");     scanf ("%d %d", &amp;m, &amp;n);     fp = fopen(fname, "at", "");     for (i = m; i &lt;= n; i++)     {         if (is_prime(i))             fprintf(fp, "%d ", i);     }     fclose (fp);     fp = fopen(fname, "rt", "");     printf("Prime numbers in primes.dat file:\n");     while (fscanf (fp, "%d", &amp;i) != EOF)         printf("%d ", i);     fclose (fp);     return 0;     int is_prime(int n)     int d;     for (d = 2; d &lt; n; d++)     {         if (n % d == 0)             break;     }     return (d == n) ? 1 : 0; } </pre>



<b>Problem Statement</b>	<p>Write the following C program using pointer</p> <p>a) To sort the list of numbers through pointer</p> <p>b) To reverse the string through pointer.</p>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Make a block of memory using calloc or malloc</li> <li>• Input the elements</li> <li>• Then apply bubble sort same as above just use pointer here</li> <li>• For reverse start a loop from 1<sup>st</sup> element to middle element and swap each element with element from last like 1<sup>st</sup> with last, 2<sup>nd</sup> with 2<sup>nd</sup> last, 3<sup>rd</sup> with 3<sup>rd</sup> last and so on.</li> </ul>
<b>Code</b>	<p>a) To sort the list of numbers through pointer</p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; #include &lt;alloc.h&gt; void main() {     int n,*p,i,j,temp;     clrscr();     printf("\nHOW MANY NUMBER: ");     scanf("%d",&amp;n);     p=(int *) malloc(n*2);     if(p==NULL)     {         printf("\nMEMORY ALLOCATION UNSUCCESSFUL");         exit();     }     for(i=0;i&lt;n;i++)     {         printf("\nENTER NUMBER %d: ",i+1);         scanf("%d",p+i);     }     for(i=0;i&lt;n;i++)     {         for(j=0;j&lt;n;j++)         {             if(*(p+i)&lt;*(p+j))             {                 temp=*(p+i);                 *(p+i)=*(p+j);                 *(p+j)=temp;             }         }     }     printf("\nTHE SORTED NUMBERS ARE:\n");</pre>



```

        for(i=0;i<n;i++)
            printf("%d ",*(p+i));
        getch();
    }

```

b) To reverse the string through pointer.

```

#include<stdio.h>

int string_length(char*);
void reverse(char*);
main()
{
    char string[100];
    printf("Enter a string\n");
    gets(string);
    reverse(string);
    printf("Reverse of entered string is \"%s\".\n", string);
    return 0;
}
void reverse(char *string)
{
    int length, c;
    char *begin, *end, temp;
    length = string_length(string);
    begin = string;
    end = string;
    for ( c = 0 ; c < ( length - 1 ) ; c++ )
        end++;
    for ( c = 0 ; c < length/2 ; c++ )
    {
        temp = *end;
        *end = *begin;
        *begin = temp;
        begin++;
        end--;
    }
}
int string_length(char *pointer)
{
    int c = 0;
    while( *(pointer+c) != '\0' )
        c++;
    return c;
}

```



<b>Problem Statement</b>	<b>Write a C program to find the largest no among 20 integers array using dynamic memory allocation.</b>
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Make a block of memory using calloc or malloc</li> <li>• Input the elements</li> <li>• Set max = first element</li> <li>• Run loop from second element to last element and compare if element is larger than max then update max</li> <li>• After loop print max</li> </ul>
<b>Code</b>	<pre> #include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; int main() {     int num;     float *data;     printf("Enter the total number of elements: ");     scanf("%d", &amp;num);     data = (float *)calloc(num, sizeof(float));     if (data == NULL) {         printf("Error!!! memory not allocated.");         exit(0);     }     for (int i = 0; i &lt; num; ++i) {         printf("Enter Number %d: ", i + 1);         scanf("%f", data + i);     }     for (int i = 1; i &lt; num; ++i) {         if (*data &lt; *(data + i))             *data = *(data + i);     }     printf("Largest number = %.2f", *data);     return 0; } </pre>





<b>Problem Statement</b>	Using Dynamic Memory Allocation, Write a C program to find the transpose of given matrix.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Make a block of memory using calloc or malloc</li> <li>• Input the order of square matrix</li> <li>• Input the elements</li> <li>• Initialize nested loop of i and j.</li> <li>• Swap element at i, j with element at j, i for transpose.</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; int main() { int *a; scanf("%d",&amp;n); a=(int*)calloc(n,n*sizeof(int)); for(i=0;i&lt;n;i++)     for(j=0;j&lt;n;j++)         scanf("%d",*(a+i)+j); for(i=0;i&lt;n;i++)     for(j=0;j&lt;n;j++)         printf("%d",*(*(a+j)+i)); }</pre>



<b>Problem Statement</b>	Write a program to find the factorial of given number using command line argument.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take one input as command line argument</li> <li>• Rest program Same as Factorial Program.</li> </ul>
<b>Code</b>	<pre>#include&lt;stdio.h&gt; #include&lt;stdlib.h&gt; int main(int argc, char* argv[]) {     int n=atoi(argv[1]);     int i,fact=1;     for(i=1;i&lt;=n;i++)         fact*=i;     printf("%d",fact);     return 0; }</pre>

<b>Problem Statement</b>	Write a program to find the sum of digits of a 5 digit number using command line argument.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take one input as command line argument</li> <li>• Rest program Same as Sum of Digit Program.</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; int main(int argc, char * argv[]) {     long num, temp, digit, sum = 0;     if(argc == 1    argc &gt; 2)     {         printf("Enter the number\n");         exit(1);     }     num = atoi (argv[1]) ;     temp = num;     while (num &gt; 0)     {         digit = num % 10;         sum = sum + digit;         num /= 10;     }     printf("Sum of the digits of %ld = %ld\n", temp, sum); }</pre>



<b>Problem Statement</b>	Write a program to construct a Fibonacci series up to n terms using command line argument.
<b>Concept</b>	<ul style="list-style-type: none"> <li>• Take one input as command line argument</li> <li>• Rest program Same as Fibonacci series.</li> </ul>
<b>Code</b>	<pre>#include &lt;stdio.h&gt; void main(int argc, char * argv[]) {     int n, last = 0, prev = 1, curr, cnt;     n = atoi(argv[1]);     printf("Printing first %d fibonacci nos. -&gt; ", n);     printf("%d ", last);     printf("%d ", prev);     cnt = 2;     while (cnt&lt;= n-1)     {         curr = last + prev;         last = prev;         prev = curr;         cnt++;         printf("%d ", curr);     }     printf("\n"); }</pre>

