Varuvan Vadivelan Institute of Technology

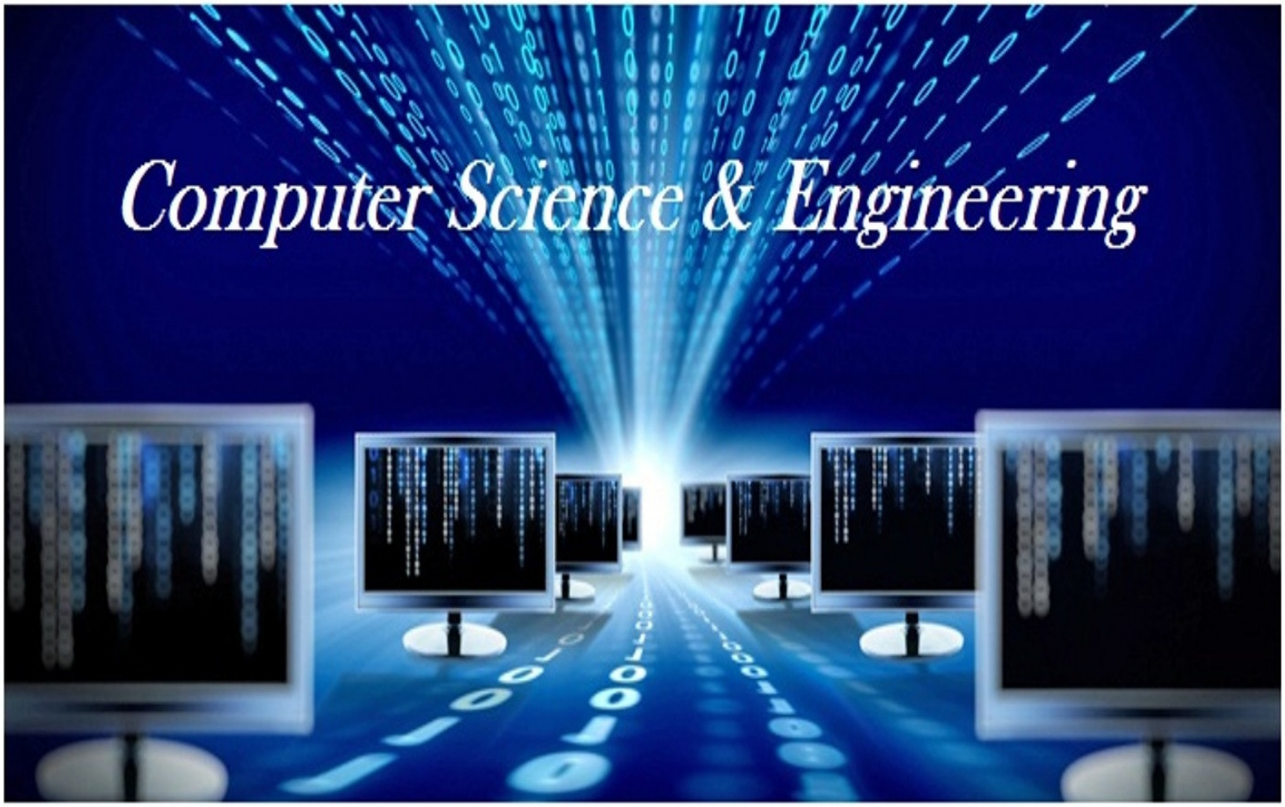
LAB MANUAL

Regulation : *2013*

Branch : *B.E. – All Branches*

Year & Semester : *I Year / I Semester*

GE6161 – COMPUTER PRACTICES LAB



ANNA UNIVERSITY: CHENNAI

REGULATION - 2013

GE6161 – COMPUTER PRACTICES LABORATORY

LIST OF EXPERIMENTS:

1. Search, generate, and manipulate data using MS office / Open Office
2. Presentation and Visualization – graphs, charts, 2D, 3D
3. Problem formulation, Problem Solving and Flowcharts
4. C Programming using Simple statements and expressions
5. Scientific problem solving using decision making and looping.
6. Simple programming for one dimensional and two dimensional arrays.
7. Solving problems using String functions
8. Programs with user defined functions – Includes Parameter Passing
9. Program using Recursive Function and conversion from given program to flow chart.
10. Program using structures and unions.

TOTAL PERIODS: 45

INDEX

S.NO	DATE	TITLE	SIGNATURE OF THE STAFF	REMARKS
MICROSOFT WORD				
1		Creating Advertisement		
2		Curriculum Vitae		
3		Scientific Notations		
4		Creating Timetable and Conversion		
5		Mail merge and Letter preparation		
6		Drawing flowchart		
MICROSOFT EXCEL				
7		Spread sheet chart (Line, XY, Bar and Pie)		
8		Spread sheet formula editor		

INDEX

S.NO	DATE	TITLE	SIGNATURE OF THE STAFF	REMARKS
		C PROGRAMS		
9		Area and Circumference of the Circle		
10		Ternary operator		
11		Finding the roots of quadratic equation		
12		Armstrong number		
13		Factorial of a number		
14		Fibonacci Series		
15		Sum of the digits, Reverse and Palindrome		
16		Pascal's Triangle		
17		Matrix Multiplication		
18		String Concatenation		
19		String Comparison		
20		String Copy		
21		String Length		
22		Payroll processing using Union		
23		Employee details using Structure		
24		Call by value and call by reference		

INTRODUCTION

Basic Concepts of C :

C was originally developed by Dennis Ritchie between 1969 and 1973 at Bell Labs, and used to re-implement the Unix operating system. It is the most widely used programming languages of all time. C has been standardized by the American National Standards Institute (ANSI) since 1989 and subsequently by the International Organization for Standardization (ISO).

Uses of C language:

- ❖ Database Systems
- ❖ Language Interpreters
- ❖ Compilers and Assemblers
- ❖ Operating Systems
- ❖ Network Drivers
- ❖ Word Processors

Features of C language:

- C is robust language with rich set of built-in functions and operators
- Programs written in C are efficient and fast.
- C is highly portable.
- C is basically a collection of C library functions.
- C is easily extensible.

C Data Types:

Primary Data Types	Secondary Data Types
✓ Character	Array
✓ Integer	Pointer
✓ Float	Structure
✓ Double	Union
✓ Void	Enum

C supports following conditional statements:

(i) if statement

(ii) if else statement

(iii)else if statement

(iv)switch statement

C supports following types of loops:

(i) while loops

(ii) do while loops

(iii) for loops

C Functions:

C function is a self contained block of statements that can be executed repeatedly whenever we need it.

- Provides modularity.
- Provides reusable code.
- Debugging and editing tasks are easy
- Programs can be modularized into smaller parts

Two types of functions in C:

Built in(Library) Functions

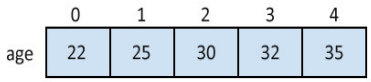
User Defined Functions

C Arrays:

An array is a data structure in C, that can store a fixed size sequential collection of elements of same data type. There are three types of arrays:

- ✓ One-dimensional array
- ✓ Two-dimensional array
- ✓ Multi-dimensional array

A pictorial representation of the array:



C Strings:

In C, the one-dimensional array of characters are called strings, which is terminated by a null character '\0'.

Ex. No : 1

Date :

CREATING ADVERTISEMENT

AIM:

To prepare an advertisement for a company with some specifications.

- Attractive page border.
- Use at least one Clip Art.
- Design name using Word Art.
- Use bullets.

ALGORITHM:

Step 1: Open a new word document using File → New option.

Step 2: Go to Page Layout → Page Borders, under the Page border tab choose the appropriate style and color, etc, and click OK.

Step 3: Go to Page Layout → Page Color, choose the appropriate color.

Step 4: Go to Insert → Clip Art, search for a relevant picture from the collections and insert it in to the page.

Step 5: Go to Insert → Word Art, choose the appropriate style from the list, type the company name and click OK.

Step 6: Type the company details in the document and do the following steps for various styles.

Step 7: For bullets and numbering, select the appropriate style from the menu and apply to the paragraph.

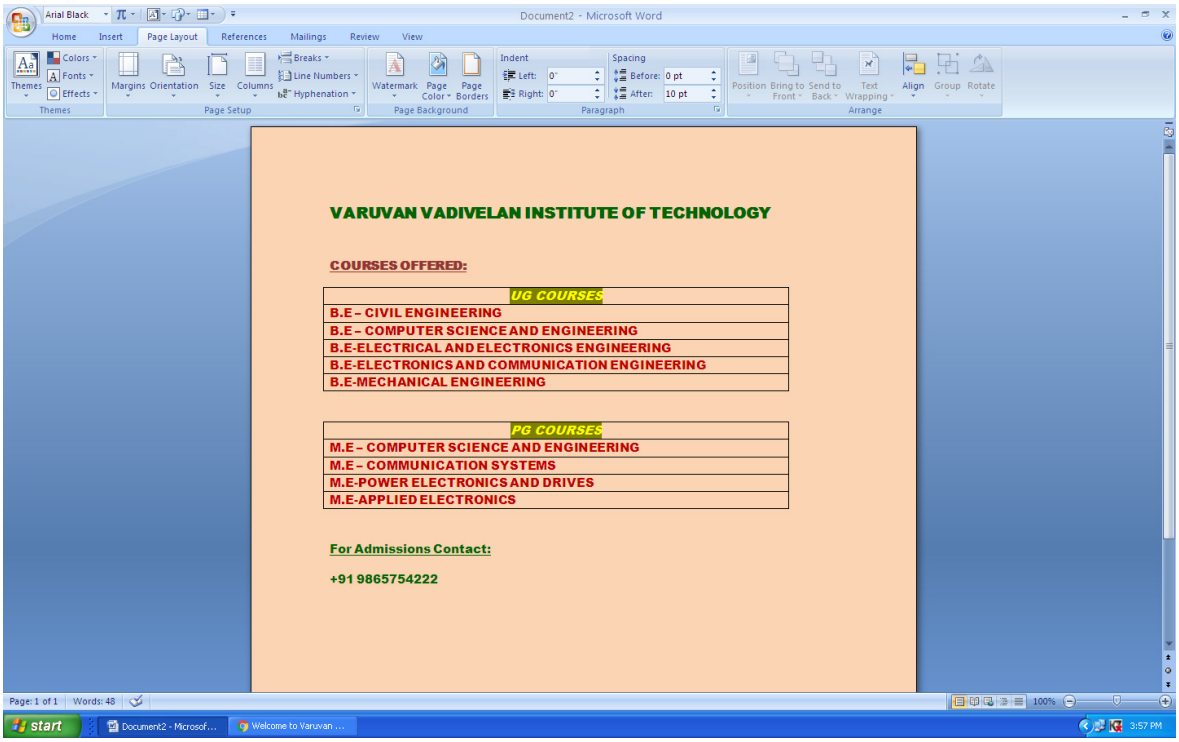
Step 8: For alignment, select the particular word or statement or paragraph in the document then press CTRL+L (left) or CTRL+R (right) or CTRL+E (center) or CTRL+J (justify) or select the relevant button from the formatting toolbar.

Step 9: For bold facing, select the particular word or statement or paragraph then press CTRL+B or select the on the formatting toolbar.

Step 10: For italic style, select the particular word then press CTRL+I or select the on the formatting toolbar..

.

OUTPUT:



RESULT:

Thus the advertisement has been created with some specifications in Microsoft word successfully and verified.

Ex. No: 2

Date:

CURRICULUM VITAE

AIM:

To create curriculum vitae (CV) of a B.E graduate with the specification.

- Table to show qualifications with heading.
- Left & Right margins
- Page numbers in the footer on the right side.
- Use Watermark.

ALGORITHM:

Step 1: Open a blank document.

Step 2: Type a Bio-data briefly then goto Insert →Table→Insert→Table→Select number of rows & columns→Ok for qualifications.

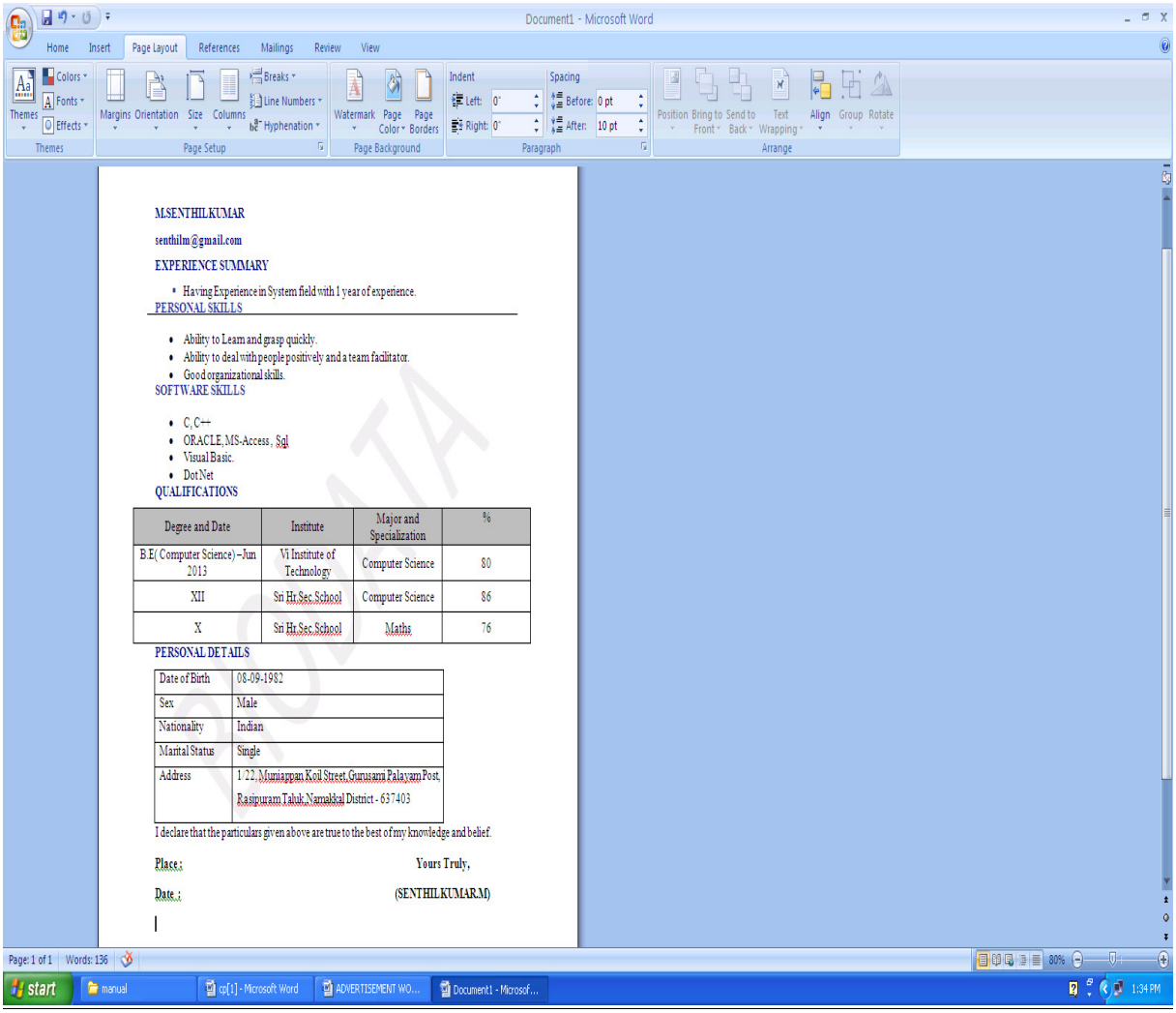
Step 3: Go to Page Layout→Margins→Assign→Left & Right Margins

Step 4: Go to Insert →Page Numbers→Select footer on the right side→Ok.

Step 5: Go to Page Layout→Watermark→Customize text as Bio-data→Ok.

Step 6: Save the Document.

OUTPUT:



RESULT:

Thus the curriculum vitae (CV) has been created with some specifications in Microsoft word successfully and verified.

Ex. No: 3

Date:

SCIENTIFIC NOTATIONS

AIM:

To create a MS-WORD document for the following scientific notation

i. A= $a_0^2 + a_1^3 + b_0^2 + b_1^{-4}$

ii. $x_1\ y_1 + x\frac{(y+z)^2}{x^2+y^2}$

iii. $x = \frac{-b+\sqrt{b^2-4ac}}{2a}$

iv. $T(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\ \ t_{ij}\ \ x_{ij}$

v. $2c_2H_6(9) + 7o_2(g) \rightarrow 4co_2\ (g) + 6H_2o$

.

ALGORITHM:

Step 1: Open a blank document.

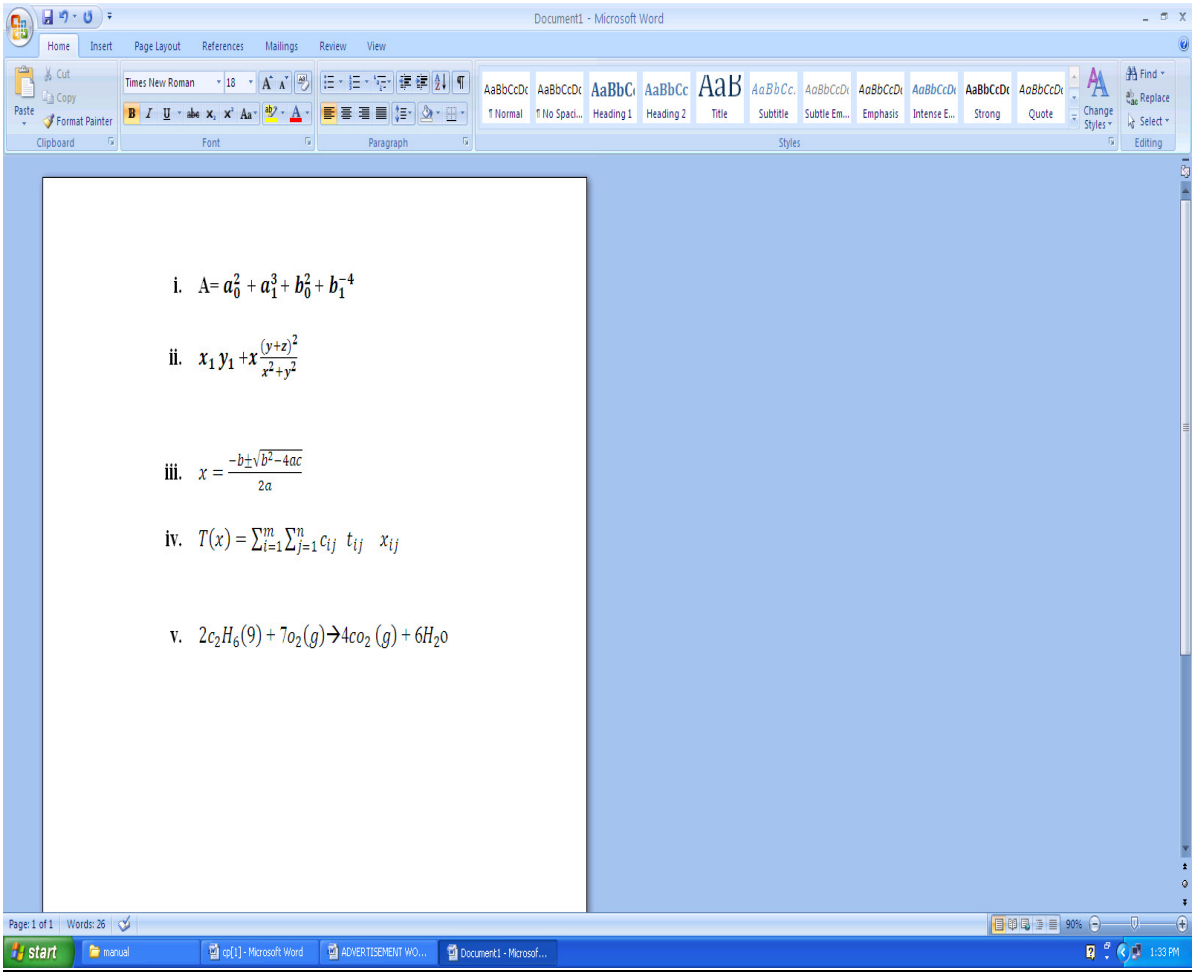
Step 2: Go to Insert→Equation→Insert Equation→Select the specific format. For eg , e^x for a^2,a^{-4}

Step 3: Select $\sum$ for $\sum_{i=1}^{m}\sum_{j=1}^{n}$

Step 4: Select $\frac{}{}$ for $\frac{x(y+z)^2}{x^2+y^2}$

Step 5: Save the document.

OUTPUT:



RESULT:

Thus the scientific notations has been created in Microsoft word successfully and verified.

Ex. No: 4

Date :

CREATING TIME TABLE & CONVERSION

AIM:

To prepare a class timetable using Merge rows, Split row, Insert rows, columns and convert the table into text format.

ALGORITHM:

Step1: Open a blank document.

Step 2: Insert→Table→Insert Table→Select No of rows & columns→Ok.

Step 3: Select two cells Right click → Merge Cells.

Select one cell Right click→Split Cell

Select one row Right click →Insert→Insert One row above or below

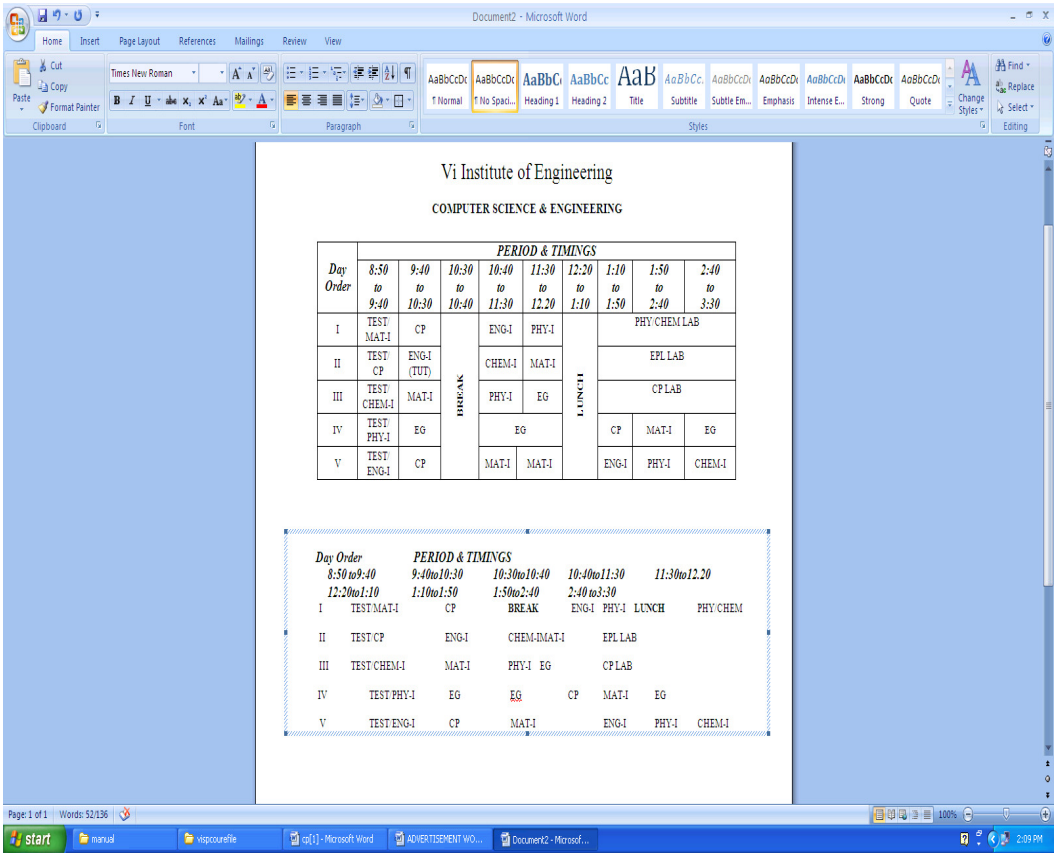
Select one column Right click →Insert→Insert One column left or right

Step 4: Type a Class Timetable with Headings

Step 5: Go to Layout→Convert to text→Select Tabs→ Ok

Step 6: Save the document as Table and Text Format

OUTPUT:



Vi Institute of Engineering

COMPUTER SCIENCE & ENGINEERING

Day Order	PERIOD & TIMINGS								
	8:50 to 9:40	9:40 to 10:30	10:30 to 10:40	10:40 to 11:30	11:30 to 12.20	12:20 to 1:10	1:10 to 1:50	1:50 to 2:40	2:40 to 3:30
I	TEST/ MAT-I	CP	BREAK	ENG-I	PHY-I	LUNCH	PHY/CHEM LAB		
II	TEST/ CP	ENG-I (TUT)		CHEM-I	MAT-I		EPL LAB		
III	TEST/ CHEM-I	MAT-I		PHY-I	EG		CP LAB		
IV	TEST/ PHY-I	EG		EG			CP	MAT-I	EG
V	TEST/ ENG-I	CP		MAT-I	MAT-I		ENG-I	PHY-I	CHEM-I

Day Order	PERIOD & TIMINGS						
	8:50 to9:40	9:40to10:30	10:30to10:40	10:40to11:30	11:30to12.20	12:20to1:10	1:10to1:50
I	TEST/MAT-I	CP	BREAK	ENG-I	PHY-I	LUNCH	PHY/CHEM
II	TEST/CP	ENG-I	CHEM-IMAT-I		EPL LAB		
III	TEST/CHEM-I	MAT-I	PHY-I	EG	CP LAB		
IV	TEST/PHY-I	EG	EG	CP	MAT-I	EG	
V	TEST/ENG-I	CP	MAT-I		ENG-I	PHY-I	CHEM-I

RESULT

Thus the class time table has been created & table is converted into text in Microsoft word successfully and verified.

Ex. No: 5

Date :

MAIL MERGE & LETTER PREPARATION

AIM:

To create a WORD document to call letters for an interview using Mail Merge send to10 candidates

ALGORITHM:

Step 1: Open a blank document

Step 2: Goto Mailings in Menu → Start Mail merge →Letters

Step 3: Type a interview call letter with FROM address and leave some

Space for TO address

Step 4: Goto →Select recipients → Type a new list →Customize the

Columns →Ok

Step 5: Type a 10 address new some fields→->Ok→->save it

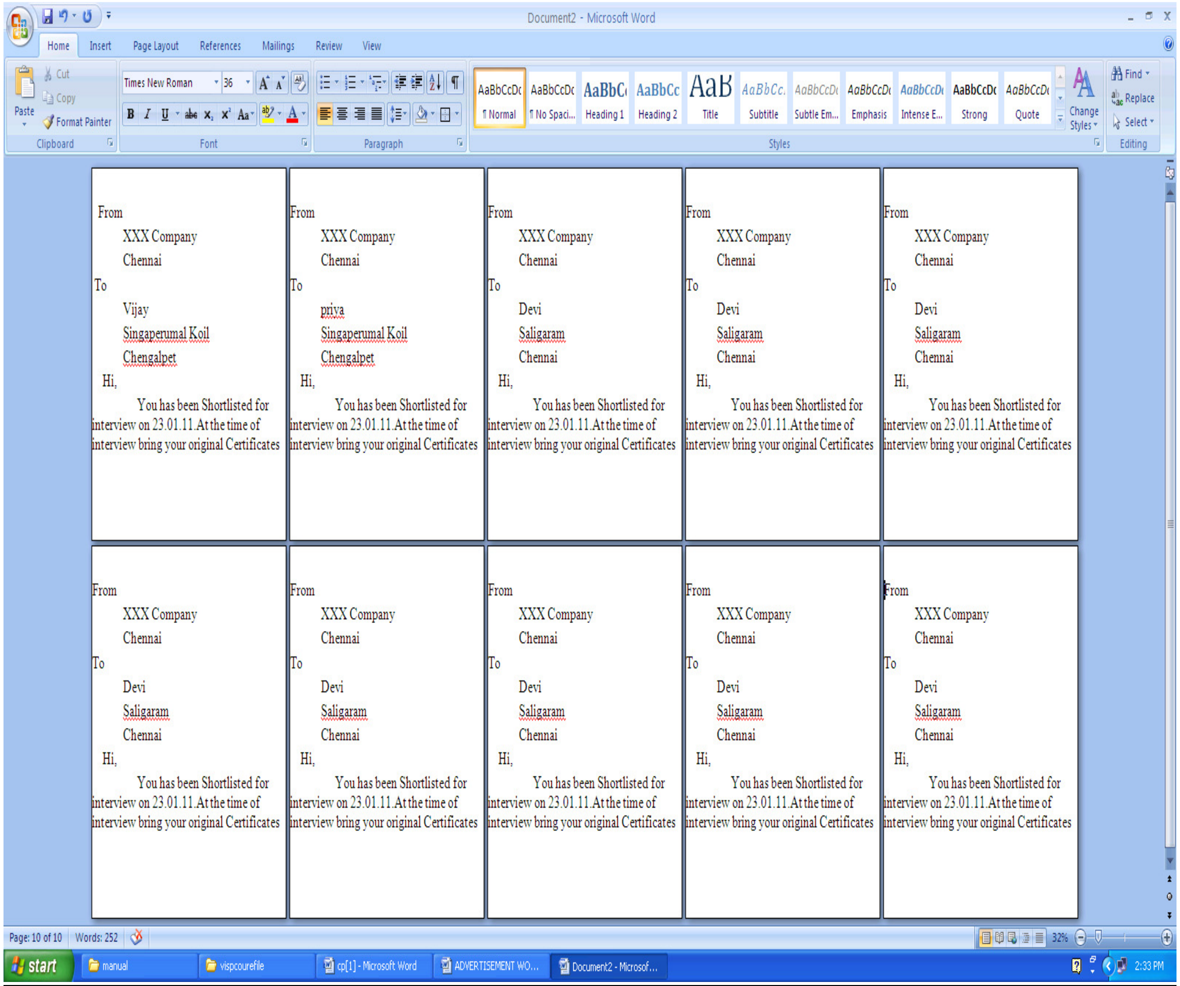
Step 6: Goto →Select recipients → Use Existing list →open a file→ ->Ok

Step 7: Under the TO Address insert the Merge fields & preview the results

Step 8: Goto Finish Merge->Edit individual Documents → All →Ok

Step 9: Save the document

OUTPUT:



RESULT:

Thus the Mail Merge has been created in Microsoft word successfully and Verified.

Ex. No: 6

Date :

DRAWING FLOW CHART

AIM:

To create a flowchart in WORD to find the greatest of three numbers

ALGORITHM:

Step 1: Open a blank document

Step 2: Go to Insert → shapes→Flowchart

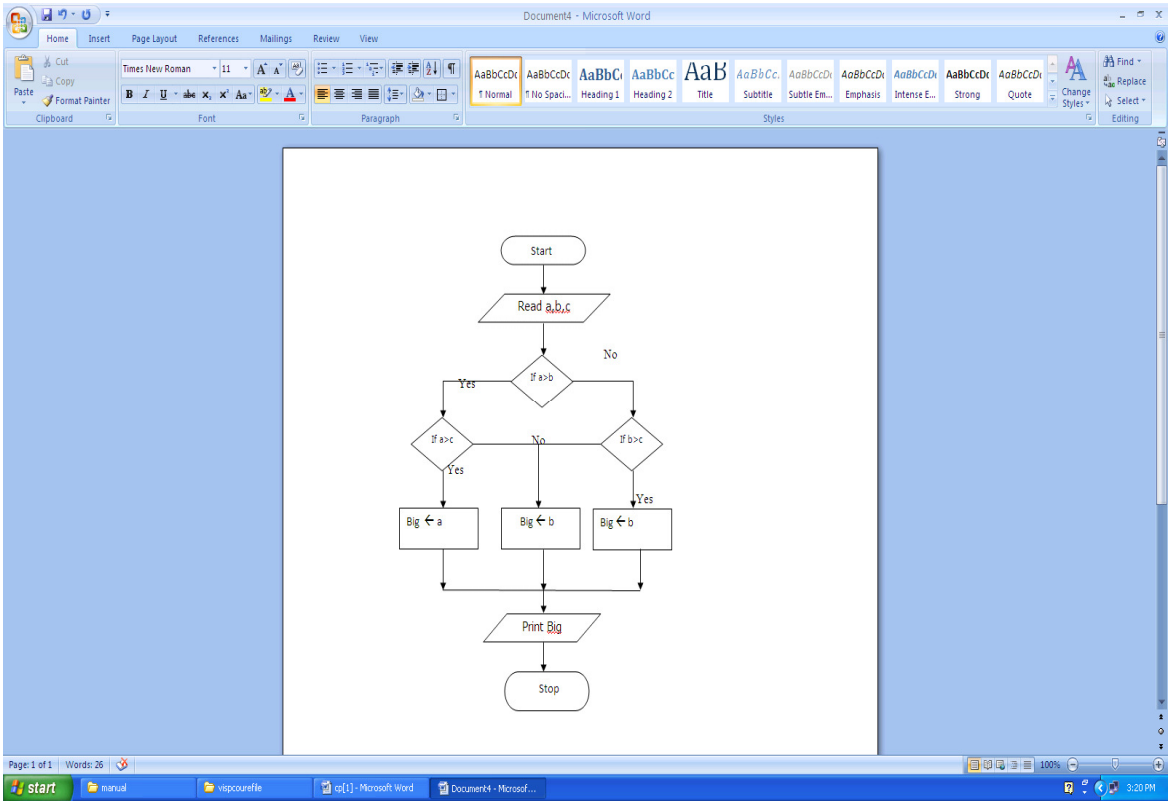
Step 3: Insert the Correct shapes for Input box, decision box, Calculation box and Output box

Step 4: Select the box and Right Click → Add Text

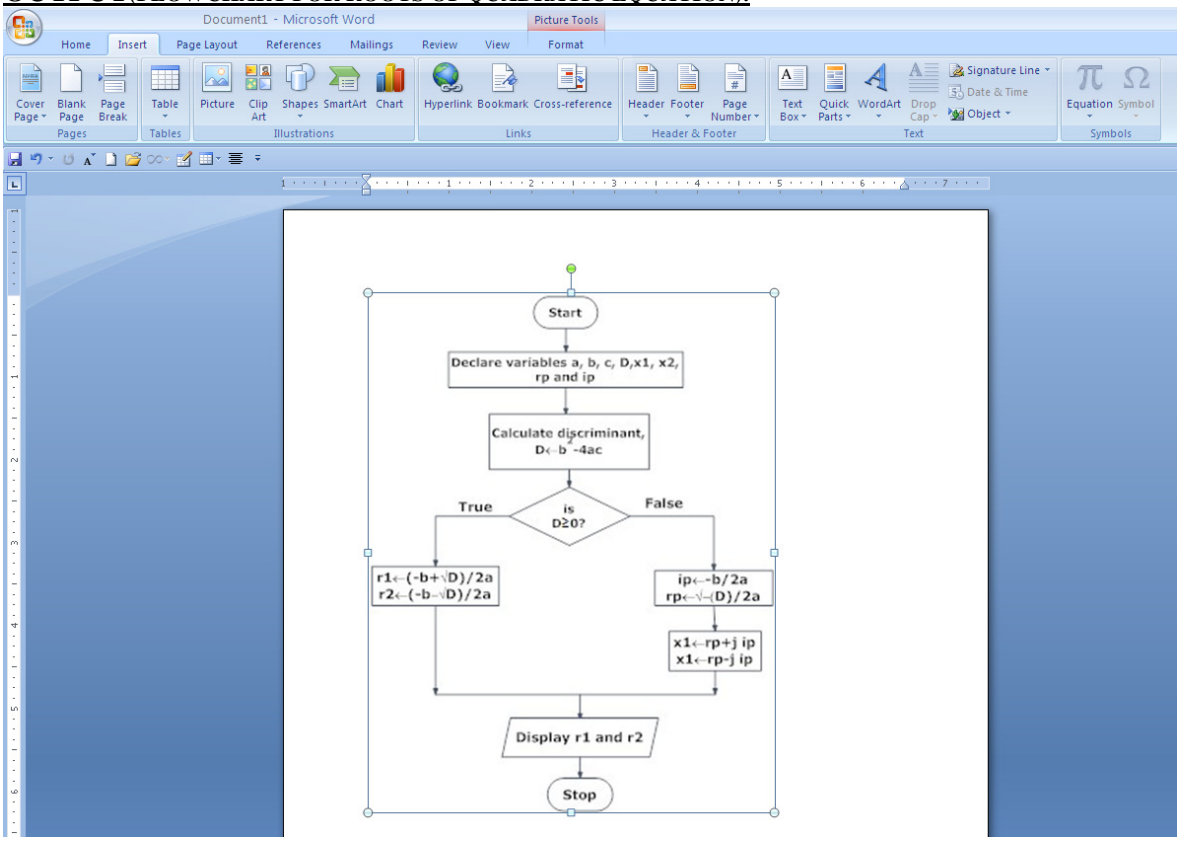
Step 5: Use Arrows for Link

Step 6: Save the document

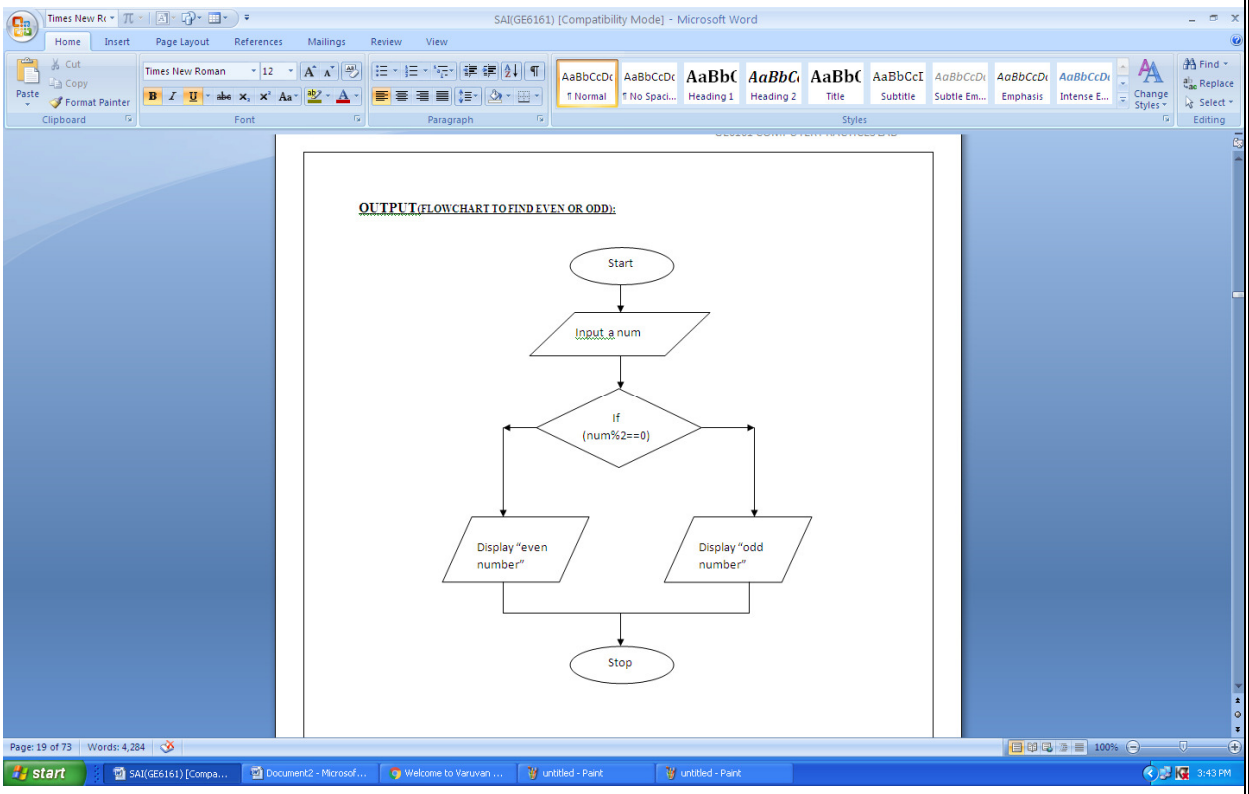
OUTPUT(FLOWCHART FOR BIGGEST OF TWO NUMBERS):



OUTPUT(FLOWCHART FOR ROOTS OF QUADRATIC EQUATION):



OUTPUT(FLOWCHART TO FIND EVEN OR ODD):



RESULT:

Thus the flowchart has been drawn using Ms-word successfully and verified.

Ex. No: 7

Date :

SPREAD SHEET CHART (Line,XY,Bar and Pie)

AIM:

To create a EXCEL to analyze the marks of the students of a class using various Chart (Line,XY,Bar and Pie).

ALGORITHM:

Step 1:. Open a Microsoft Excel Worksheet.

Step 2: Place the Cursor on the desired cell and start entering the required Student details

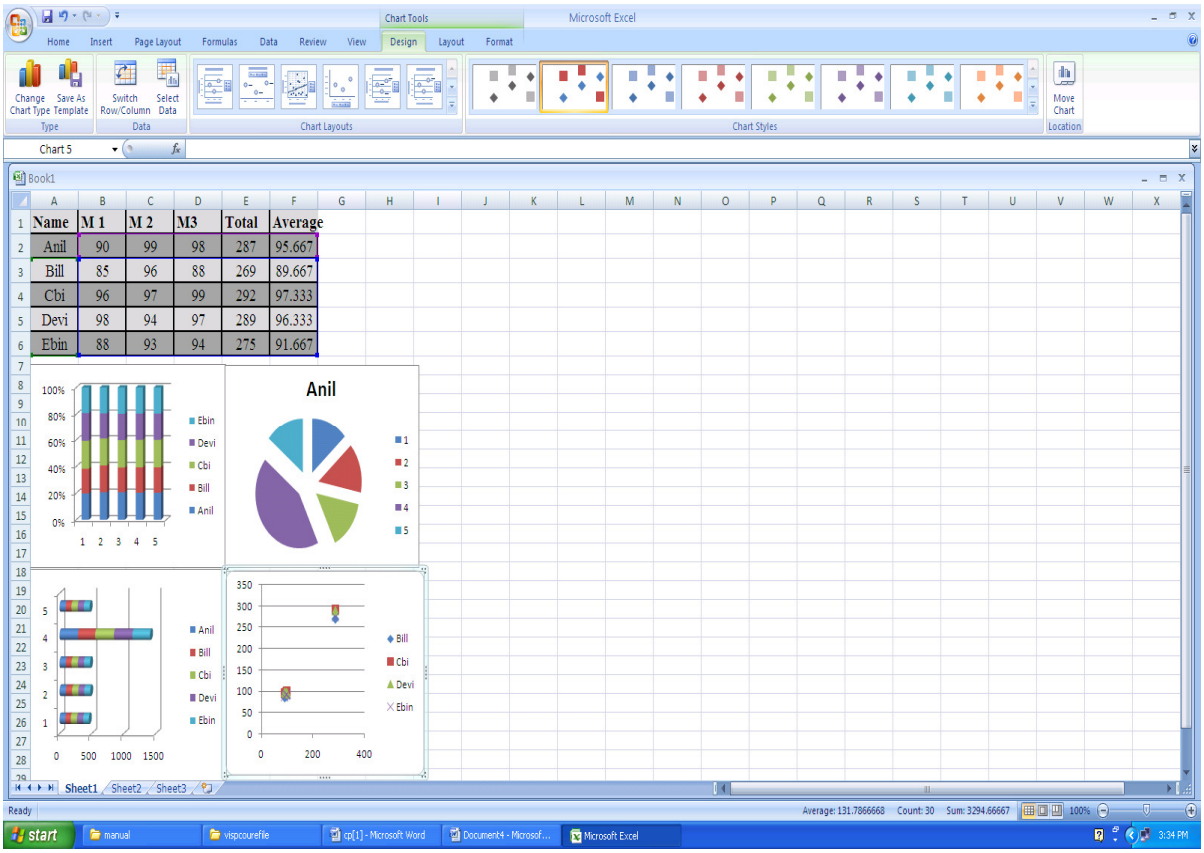
Step 3: To find the Total and Average using formula (Total = m1+m2+m3)

Average = (Total / 3)

Step 4: Select the table and goto Insert → Chart → Choose one type of Chart

Step 5: Reselect the table again and Insert → Chart → Choose another type of Chart. Repeat these steps for all types of charts.

OUTPUT:



RESULT:

Thus the Spreadsheet charts (Line,XY,Bar and Pie) for students marks has been created Successfully and verified.

Ex. No: 8

Date :

SPREAD SHEET FORMULA EDITOR

AIM:

To create a spreadsheet to calculate HRA , DA, TA, PF, LIC. Gross Salary , Net Salary from the below given data

HRA=18% of basic Pay TA=12% of Basic Pay DA=15% of Basic Pay

PF =10% of Basic Pay LIC =7% of Basic Pay Deduction= PF + LIC

Gross Salary = Basic Pay + HRA + DA + TA Net Salary = Gross Salary – Deduction

ALGORITHM:

Step 1. Open a Microsoft Excel Worksheet

Step 2. Type the details about the employees and Basic Salary.

Step 3. For HRA & DA, move to corresponding row & column and assign the formula =18/100* BS (row & column) For DA , move to corresponding row & column and assign the formula =15/100* BS (row & column)

. **Step 4.** For TA & PF, move to corresponding row & column and assign the formula =12/100* BS (row & column) For PF ,move to corresponding row & column and assign the formula =10/100* BS (row & column)

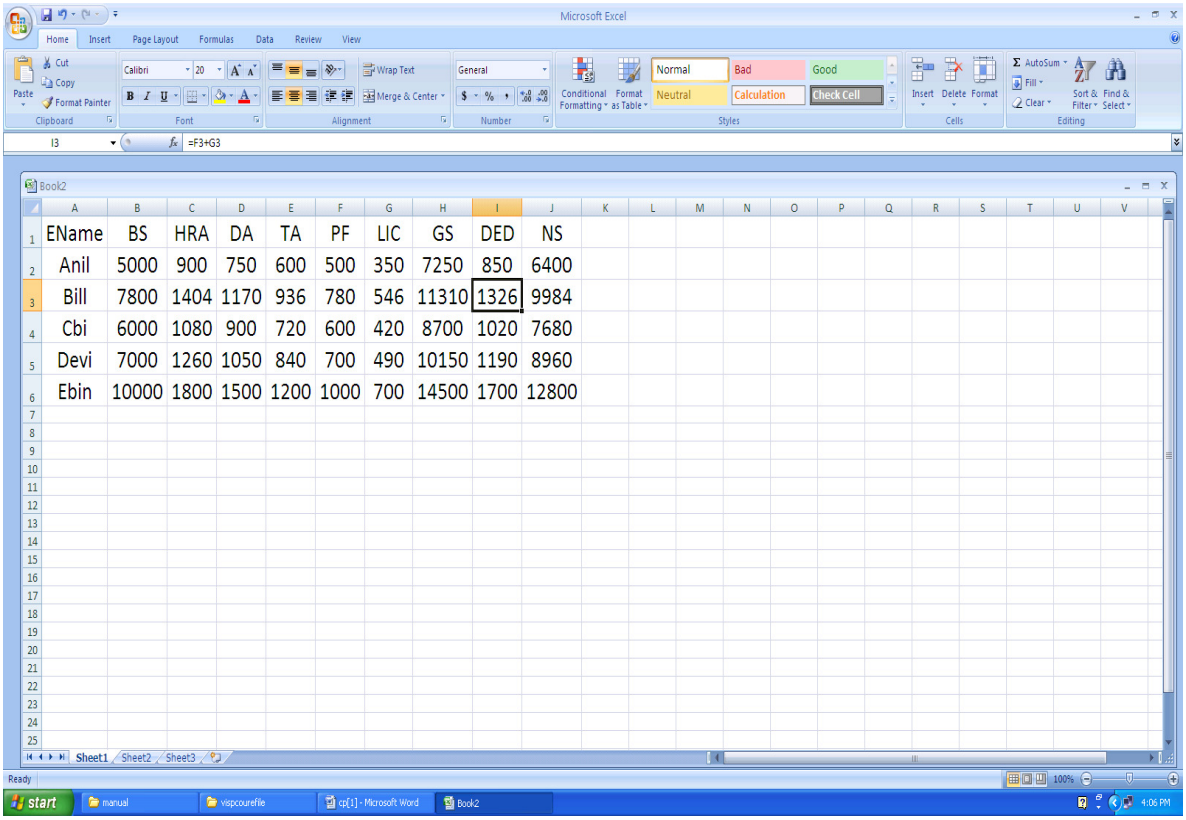
Step 5. For LIC & GS, move to corresponding row & column and assign the formula =7/100* BS,For GS ,move to corresponding row & column and assign the formula = Basic Pay + HRA + DA + TA

Step 6. Likewise for Deduction and Net Salary

Step 7. Save the ExcelSheet

Step 8. Stop the program

OUTPUT



EName	BS	HRA	DA	TA	PF	LIC	GS	DED	NS
Anil	5000	900	750	600	500	350	7250	850	6400
Bill	7800	1404	1170	936	780	546	11310	1326	9984
Cbi	6000	1080	900	720	600	420	8700	1020	7680
Devi	7000	1260	1050	840	700	490	10150	1190	8960
Ebin	10000	1800	1500	1200	1000	700	14500	1700	12800

RESULT:

Thus the Spreadsheet to calculate HRA , DA, TA, PF, LIC. Gross Salary , Net Salary from the given data has been created Successfully and verified.

Ex. No : 9

Date :

AREA AND CIRCUMFERENCE OF THE CIRCLE

AIM:

To write a C program to find the area and circumference of the circle

ALGORITHM:

Step 1: Start the program.

Step 2: Input the radius of the Circle.

Step 3: Find the area and circumference of the circle using the formula

Area =3.14*r*r

Circum=2*3.14*r

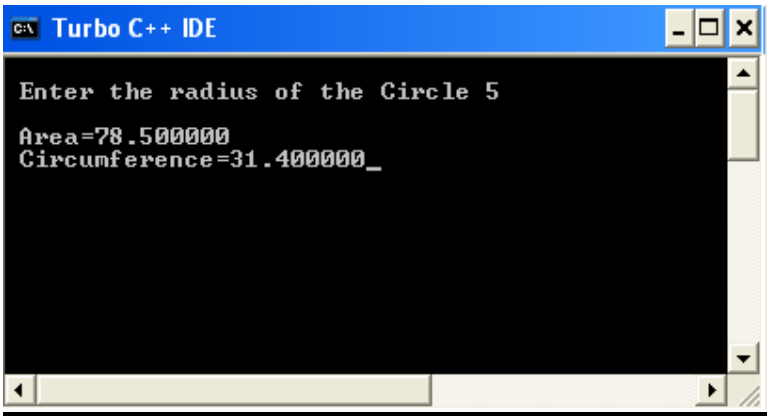
Step 4: Print the area and Circumference

Step 5: Stop the Program

PROGRAM: (AREA AND CIRCUMFERENCE OF THE CIRCLE)

```
#include<stdio.h>
#include<conio.h>
void main()
{
     float r,area,circum;
     clrscr();
     printf("\n Enter the radius of the Circle");
     scanf("%f",&r);
     area=3.14*r*r;
     circum=2*3.14*r;
     printf("\n Area=%f",area);
     printf("\n Circumference=%f",circum);
     getch();
}
```

INPUT AND OUTPUT:



RESULT:

Thus the C program to find the area and circumference of the circle has been created successfully and verified.

Ex. No : 10

Date :

TERNARY OPERATOR

AIM:

To write a C program to check the largest number among given two numbers.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the necessary variables.

Step 3: Check if(a > b)

Step 4: If true Print a.

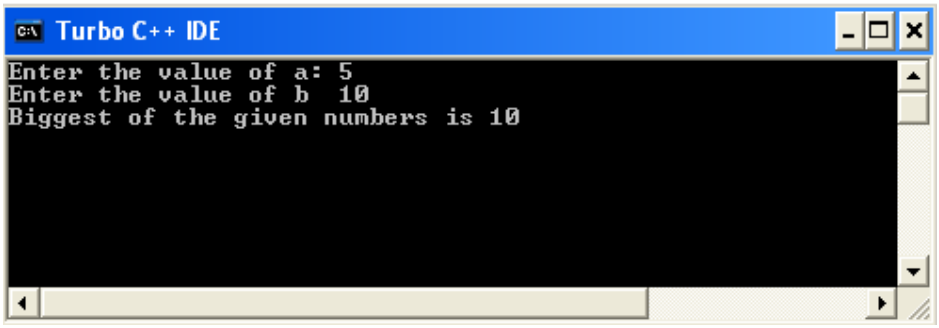
Step 5: Otherwise, Print b

Step 6: Stop the program

PROGRAM: (TERNARY OPERATOR)

```
#include<stdio.h>
#include<conio.h>
void main( )
{
int a,b,big; clrscr( );
printf("Enter the value of a: ");
scanf("%d",&a);
printf("Enter the value of b");
scanf("%d",&b);
big=(a>b)?a:b;
printf("Biggest of the given numbers is %d",big);
getch();
}
```

INPUT AND OUTPUT:



RESULT:

Thus the program for Conditional Statements has been executed successfully and the output was verified.

Ex. No : 11

Date :

FINDING THE ROOTS OF QUADRATIC EQUATION

AIM:

To write a C Program to find the roots of a Quadratic equation.

ALGORITHM:

Step 1: Start

Step 2: Read the variable a, b, c.

Step 3: Compute d= b*b - 4*a*c.

Step 4: Test the condition, d is greater than 0 using IF statement.

Calculate: r1= (-b + sqrt(d)) / (2*a).

Calculate: r2= (-b - sqrt(d)) / (2*a).

Print the roots r1 and r2.

Step 5: Else, test the condition, d is equal to 0 using IF statement.

Calculate: r1=r2 = -b / (2*a).

Print the roots r1 and r2.

Step 6: Else, compute real and imaginary as

Calculate: real = -b / (2*a).

Calculate imag = sqrt(-d)/(2*a).

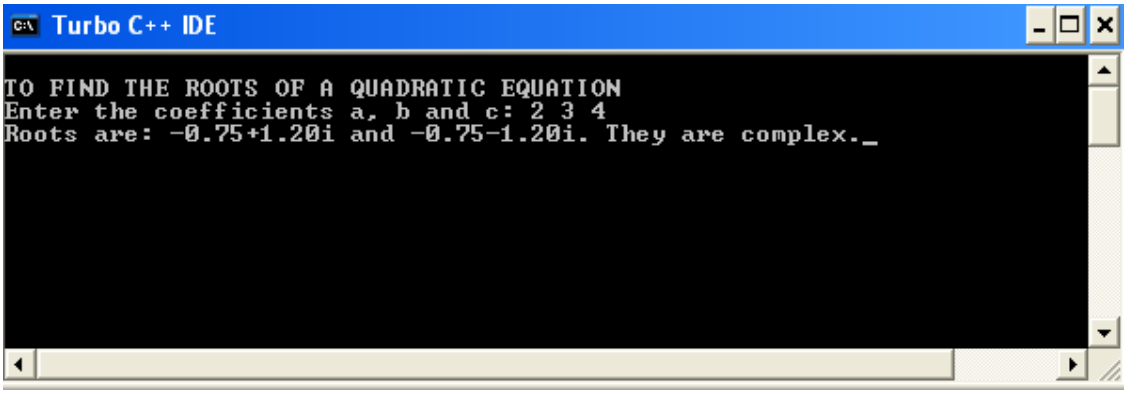
Print the real and imag.

Step 7: Stop

PROGRAM: (FINDING THE ROOTS OF QUADRATIC EQUATION)

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
void main()
{
     float a, b, c, d, r1,r2, real, imag; clrscr();
     printf("\nTO FIND THE ROOTS OF A QUADRATIC EQUATION");
     printf("\nEnter the coefficients a, b and c: ");
     scanf("%f%f%f",&a,&b,&c);
     d=b*b-4*a*c;
     if (d>0)
     {
          r1= (-b+sqrt(d))/(2*a);
          r2= (-b-sqrt(d))/(2*a);
          printf("Roots are: %.2f and %.2f.They are real and
          distinct.", r1 , r2);
     }
     else if (d==0)
     {
          r1 = r2 = -b/(2*a);
          printf("Roots are: %.2f and %.2f. They are real and
          equal.", r1, r2);
     }
     else
     {    real= -b/(2*a);
          imag = sqrt(-d)/(2*a);
          printf("Roots are: %.2f+%.2fi and %.2f-%.2fi. They
          are complex.", real, imag, real, imag);
     }
     getch();
     }
```

INPUT AND OUTPUT:



RESULT:

Thus the C program for finding roots of quadratic equation was executed and output was obtained.

Ex. No: 12

Date :

ARMSTRONG NUMBER

AIM:

To write a C Program to check whether the given number is Armstrong or not.

ALGORITHM:

Step 1: Start the program

Step2: Read the variable N

Step 3: Assign N1=N;

Step 4: Create Set a loop using the condition WHILE(N1!=0), if the condition true

REM=N1%10;

NUM=NUM+REM*REM*REM;

N1=N1/10;

Step 5: Else, check the condition IF(NUM=N), if the condition true

Step6: PRINT "Armstrong Number"

Step 7: Else PRINT "Not Armstrong Number"

Step 8: Stop the program

PROGRAM: (ARMSTRONG NUMBER)

```
#include <stdio.h>
#include <conio.h>
void main()
{
     int n, n1, rem, num=0; clrscr();
     printf("\nPRG TO CHECK WHETHER A GIVEN NO. IS
     ARMSTRONG NO. OR NOT");
     printf("Enter a positive integer: ");
     scanf("%d", &n);
     n1=n;
     while(n1!=0)
     {
          rem=n1%10;
```

```
            num=num+(rem*rem*rem);
            n1=n1/10;
        }
        if(num==n)
        printf("%d is an Armstrong number",n);
        else
        printf("%d is not an Armstrong number",n);
        getch();
    }
```

OUTPUT:

RESULT

Thus the C program to check whether the given number is Armstrong or not was executed and the output was obtained.

Ex. No: 13

Date :

FACTORIAL OF A NUMBER

AIM:

To write a program to calculate the factorial of the given number using functions**.**

ALGORITHM:

Step 1: Start the program

Step2: Enter a number.

Step 3: Set a loop to find the factorial of the given no using Fact=fact*i

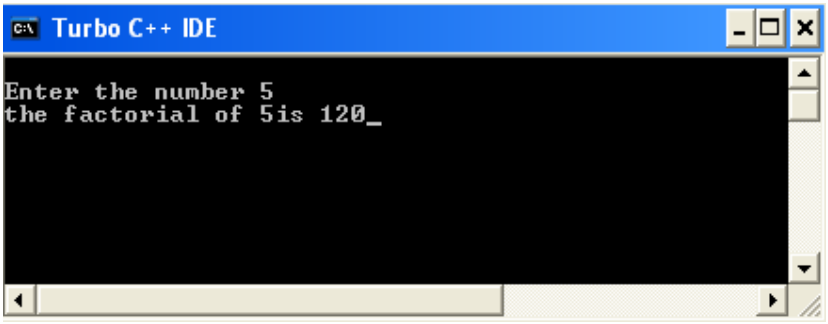
Step 4: Print the factorial of the given number.

Step 5: Stop the program

PROGRAM: (FACTORIAL OF A NUMBER)

```
#include<stdio.h>
void main()
{
     int fact=1,i,num;
     printf("Enter the number");
     scanf("%d",&num);
     for(i=1;i<=num;i++)
     {
          fact=fact*i;
     }
     printf("the factorial of %dis %d",num,fact);
     getch();
}
```

INPUT AND OUTPUT:



RESULT:

Thus the C program to calculate factorial of the given number using function is calculated successfully and verified.

Ex. No: 14

Date :

FIBONACCI SERIES

AIM:

To write a C program to find the Fibonacci series of the given number.

ALGORITHM:

Step 1. Start the program

Step 2. Enter the number.

Step 3. Check the number whether the number is zero or not.

If zero print zero value.If not zero go further.

Step 4. Set a loop upto the given number.

Step 5. Assign fib=fib+a; a=b; b=c;

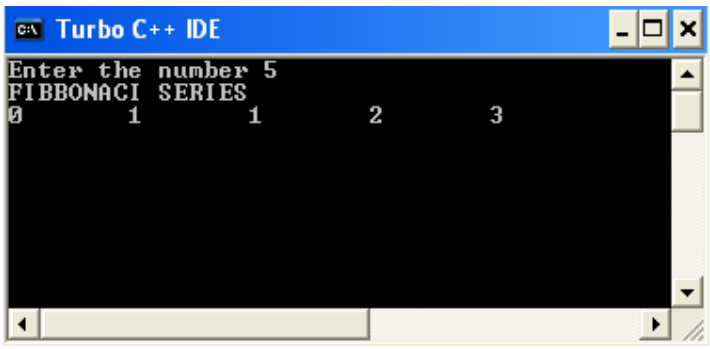
Step 6. Every increment in the loop prints the value of fib.

Step 7. After the execution of the loop, stop the program.

PROGRAM : (FIBONACCI SERIES)

```
#include<stdio.h>
#include<conio.h>
Void main()
     {
          int num,fib=0,a=0,b=1,i;
          clrscr();
          printf("Enter the number");
          scanf("%d",&num);
          printf("FIBBONACI SERIES\n");
          if(num==0)
          printf("0");
          else
          {
               for(i=0;i<num;i++)
               {
                    fib=fib+a;
                    a=b;b=fib;
                    printf("%d\t",fib);
                    getch();
               }
          }
     }
```

INPUT AND OUTPUT



RESULT:

Thus the C program to find Fibonacci series of the given number was executed and verified successfully.

Ex. No: 15

Date :

SUM OF DIGITS, REVERSE, PALINDROME

AIM:

To write a C program to find the sum & reverse of digits and Check is Palindrome or not

ALGORITHM:

Step 1. Start the program

Step 2. Enter the number.

Step 3. Set a loop upto the number is not equal to zero .

Rem←num%10 Sum←Sum+rem Rnum← rnum *10 +rem Num← num/10

Step 4. After the end of the loop print the sum and reverse no of the digit.

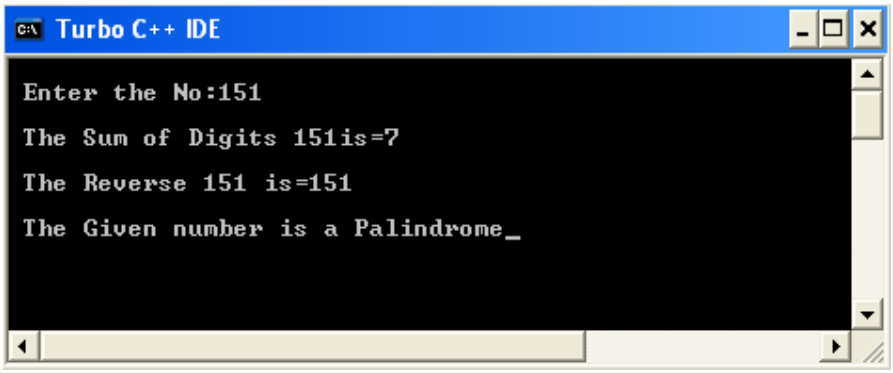
Step 5. Find whether the reverse no is equal to the input number. If equal, then it is Palindrome.

Step 6. Stop the program.

PROGRAM : (SUM OF DIGITS, REVERSE, PALINDROME)

```
#include<stdio.h>
#include<conio.h>
void main()
{
      unsigned long int a,num,sum=0,rnum=0,rem;
      clrscr();
      printf("\n Enter the No:");
      scanf("%ld",&num);
      a=num;
      while(num!=0)
      {
            rem=num%10;
            sum=sum+rem;
            rnum=rnum*10+rem;
            num=num/10;
      }
      printf("\n The Sum of Digits %ldis=%ld\n",a,sum);
      printf("\n The Reverse %ld is=%ld\n",a,rnum);
      if(a==rnum)
      printf("\n The Given number is a Palindrome");
      else
      printf("\n The Given number is not a Palindrome");
      getch();
}
```

INPUT AND OUTPUT



RESULT

Thus the C program to find the sum & reverse of digits and to check whether it is Palindrome or not was executed and verified successfully.

Ex. No: 16

Date :

PASCAL'S TRIANGLE

AIM:

To write a C program to print Pascal's triangle.

ALGORITHM:

Step 1: Start the program

Step 2: Read input n.

Step 3: Create the first loop to print n lines.

Step 4: Check l<= n

Step 5: Create the second loop to generate 40 spaces initially, then reduce by 40-3*l equation.

Step 6: Check i>0

Step 7: Create the third loop to generate and print digits.

Step 8: Use m=m*(l-j+1)/j to print digits in each line .

Step 9: Stop the execution.

PROGRAM: (PASCAL'S TRIANGLE)

```
#include<stdio.h>
#include<conio.h>
main()
{
     int i,j,l,m,n;
     clrscr();
     printf("How many rows?\n");
     scanf("%d",&n);
     printf("\n\t\t\t\t Pascal's Triangle\n");
     m=1;
     for(l=0;l<n;l++)
     {
          for(i=40-3*l;i>0;i--)
          printf(" ");
          for(j=0;j<=l;j++)
          {
               if((j==0)||(l==0))
               m=1;
               else
               m=(m*(l-j+1))/j;
               printf("%6d",m);
          }
          printf("\n");
     }
     getch();
}
```

OUTPUT:

RESULT

Thus the C program has been written to print the Pascal's triangle and the output was verified.

Ex. No: 17

Date :

MATRIX MULTIPLICATION

AIM:

To write a C program to perform Matrix Multiplication using array.

ALGORITHM:

Step 1: Start the program

Step 2: Declare variables

Step 3: Get the rows and columns of two matrices M, N, O, P respectively.

Step 4: Check N is not equal to O, if go to step 10.

Step 5: Set a loop and get the elements of first matrix A[i][i].

Step 6: Set a loop and get the elements of second matrix B[i][j].

Step 7: Repeat the step 6 until i<m, j<p and k<n.

Step 8: Initialize C[i][j]=0 and multiply the two matrices and store the resultant in C[i][j]= C[i][j]+A[i][k]*B[k][j].

Step 9: Print the resultant matrix C[i][j] and go to step 11.

Step 10: Print the message "Column of first matrix must be same as row of second matrix".

Step 11: Stop the program.

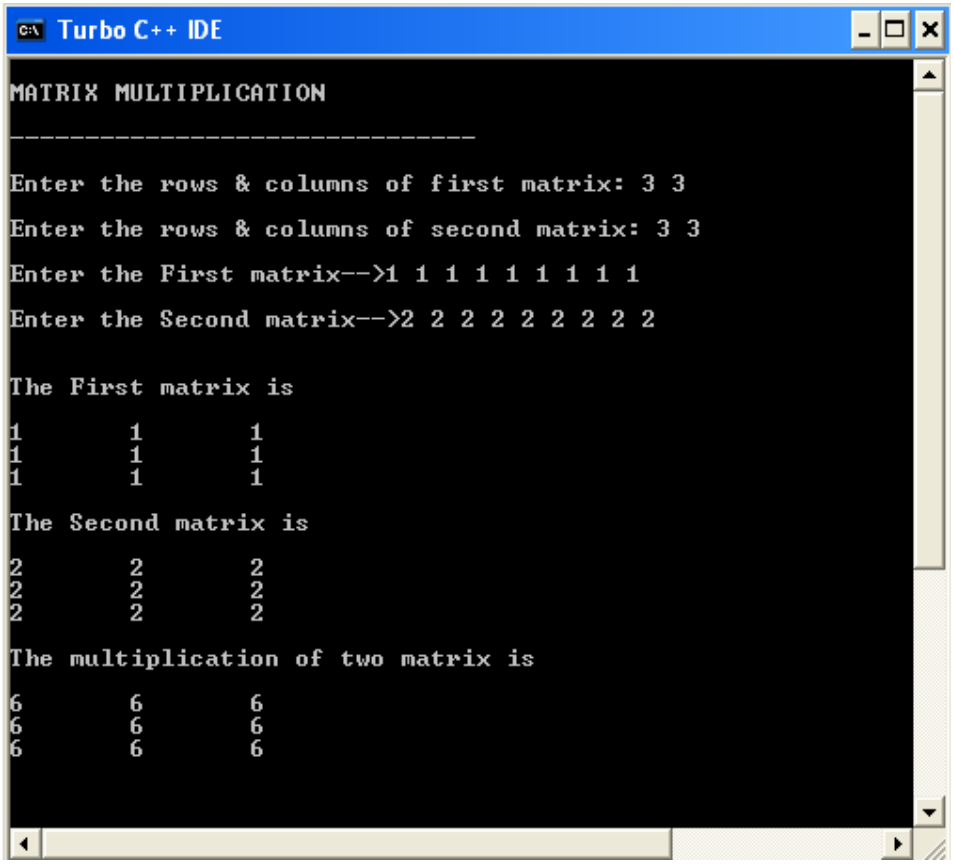
PROGRAM: (MATRIX MULTIPLICATION)

```
#include<stdio.h>
#include<conio.h>
#include<stdio.h>
void main()
{
     int a[10][10],b[10][10],c[10][10],i,j,k,m,n,o,p;
     clrscr();
     printf("\nMATRIX MULTIPLICATION\n");
     printf("\n------------------------------\n");
     printf("\nEnter the rows & columns of first matrix: ");
     scanf("%d %d",&m,&n);
     printf("\nEnter the rows & columns of second matrix: ");
     scanf("%d %d",&o,&p);
     if(n!=o)
     {
          printf("Matrix mutiplication is not possible");
          printf("\nColumn of first matrix must be same as row
          of second matrix");
     }
     else
     {
     printf("\nEnter the First matrix-->");
     for(i=0;i<m;i++)
     {
          for(j=0;j<n;j++)
          {
          scanf("%d",&a[i][j]);
          }
     }
     printf("\nEnter the Second matrix-->");
     for(i=0;i<o;i++)
     {
          for(j=0;j<p;j++)
          {
          scanf("%d",&b[i][j]);
```

```
        }
    }
    printf("\n\nThe First matrix is\n"); for(i=0;i<m;i++)
    {
        printf("\n");
        for(j=0;j<n;j++)
        {
        printf("%d\t",a[i][j]);
        }
    }
    printf("\n\nThe Second matrix is\n");
    for(i=0;i<o;i++)
    {
        printf("\n");
        for(j=0;j<p;j++)
        {
            printf("%d\t",b[i][j]);
        }
    }
    for(i=0;i<m;i++) //row of first matrix
        {
        for(j=0;j<p;j++) //column of second matrix
        {
        c[i][j]=0;
            for(k=0;k<n;k++)
            {
            c[i][j]= c[i][j]+a[i][k]*b[k][j];
            }
        }
    }
    }
    printf("\n\nThe multiplication of two matrix is\n");
    for(i=0;i<m;i++)
    {
    printf("\n");
        for(j=0;j<p;j++)
        {
```

```
            printf("%d\t",c[i][j]);
            }
      }
      getch();
}
```

OUTPUT:



RESULT

Thus the C program to perform Matrix Multiplication was executed successfully and the output was verified.

Ex. No: 18

Date :

STRING CONCATENATION

AIM:

To write a program to perform the string Concatenation using C.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the variables.

Step 3: Read input str1 and str2.

Step 4: Concatenate the two strings using for loop..

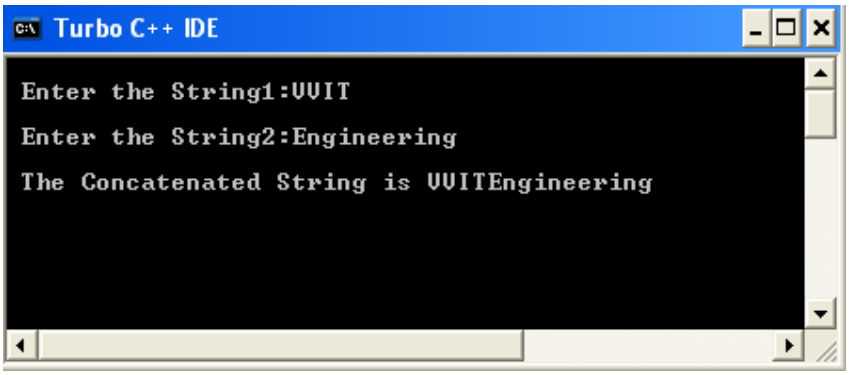
Step 5: Store the concatenated string into str[k]. Print the String

Step 6: Stop the program

PROGRAM: (STRING CONCATENATION)

```
#include<stdio.h>
#include<conio.h>
void main()
{
     int i,j,k;
     char str[10],str1[10],str2[20];
     clrscr();
     printf("\n Enter the String1:");
     gets(str1);
     printf("\n Enter the String2:");
     gets(str2);
     for(i=0,j=0;str1[i]!='\0';i++,j++)
     str[j]=str1[i];
     for(i=0,k=j;str2[i]!='\0';i++,k++)
     str[k]=str2[i];
     str[k]='\0';
     printf("\n The Concatenated String is %s",str);
     getch();
}
```

OUTPUT:



RESULT

Thus the C program to perform String Concatenation is created successfully and the output was verified.

Ex. No: 19

Date :

STRING COMPARISON

AIM:

To write a program to perform the string Comparison using C.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the variables.

Step 3: Read input str1 and str2.

Step 4: Compare the two strings using for loop..

Step5: If the strings are equal then print "Strings are equal".

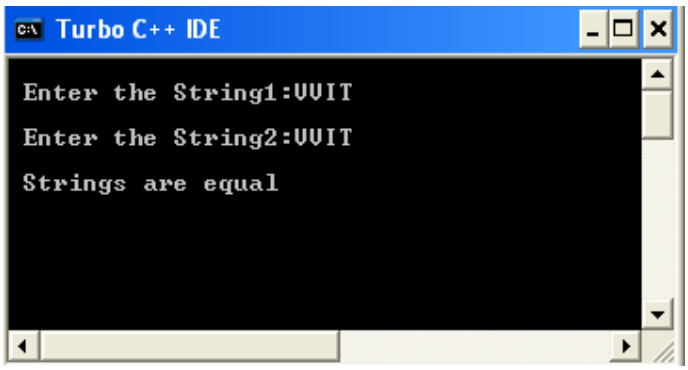
Step 6: If the strings are not equal then print "Strings are not equal".

Step 7: Stop the program

PROGRAM: (STRING COMPARISON)

```
#include<stdio.h>
#include<conio.h>
void main()
{
     int i;
     char str1[10],str2[10];
     printf("\n Enter the String1:");
     gets(str1);
     printf("\n Enter the String2:");
     gets(str2);
     for(i=0;str1[i]!='\0'||str2[i]!='\0';i++)
     if(str1[i]!=str2[i])
          {
          printf("\n Strings are not equal");
          break;
          }
     else
          {
          printf("\n Strings are equal");
          break;
          }
     getch();
}
```

OUTPUT:



RESULT

Thus the C program to perform String Comparison is created successfully and the output was verified.

Ex. No: 20

Date :

STRING COPY

AIM:

To write a program to perform the string copy using C.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the variables.

Step 3: Read input str1 and str2.

Step 4: Copy the two strings using for loop..

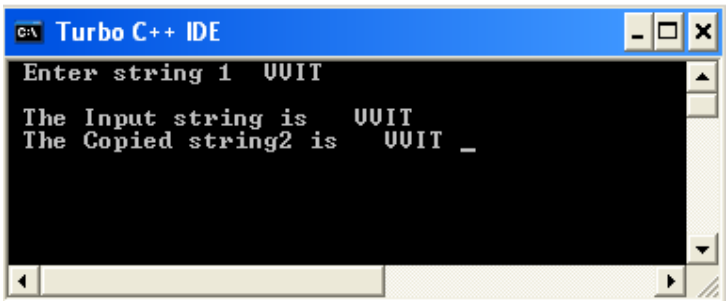
Step5: Store the string into str2. Print the String

Step 6: Stop the program

PROGRAM: (STRING COPY)

```
#include<stdio.h>
void main()
{
     int i;
     char str1[10],str2[10];
     clrscr();
     printf(" Enter string 1");
     gets(str1);
     for(i=0;str1[i]!='\0';i++)
     str2[i] = str1[i];
     str2[i] = '\0';
     printf("\n The Input string is %s",str1);
     printf("\n The Copied string2 is %s ",str2);
     getch();
}
```

OUTPUT:



RESULT

Thus the C program to perform String Copy is created successfully and the output was verified.

Ex. No: 21

Date :

STRING LENGTH

AIM:

To write a program to find the length of the string using C.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the variables.

Step 3: Read input str.

Step 4: Find the string length using for loop..

Step5: Print the String length.

Step 6: Stop the program

PROGRAM: (STRING LENGTH)

```
#include<stdio.h>
#include<conio.h>
void main()
{
      int i;
      char str[20];
      clrscr();
      printf("\n Enter the String");
      gets(str);
      for(i=0;str[i]!='\0';++i);
      printf("\n The no of characters in the String is %d",i);
      getch();
}
```

OUTPUT:

RESULT

Thus the C program to find length of the string was created successfully and the output was verified.

Ex. No: 22

Date :

PAYROLL PROCESSING USING UNION

AIM:

To write a C Program to create payroll processing using union.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the Union Employee with data members such as name, eno, basic salary, net salary, gross.

Step 3: Get the details of an employee.

Step 4: Enter the employee details such as Name, Emp No and Basic salary.

Step5: Copy the emp.ename to Name by using STRCPY(NAME,EMP.ENAME) and assign ID=EMP.ENO, BASIC=EMP.BSAL.

Step 6: Calculate HRA=10%*BASIC, DA=35%*BASIC, and TDS=15%*BASIC

Step 7: Calculate GROSS=BASIC+HRA+DA and NET SALARY=GROSS-TDS

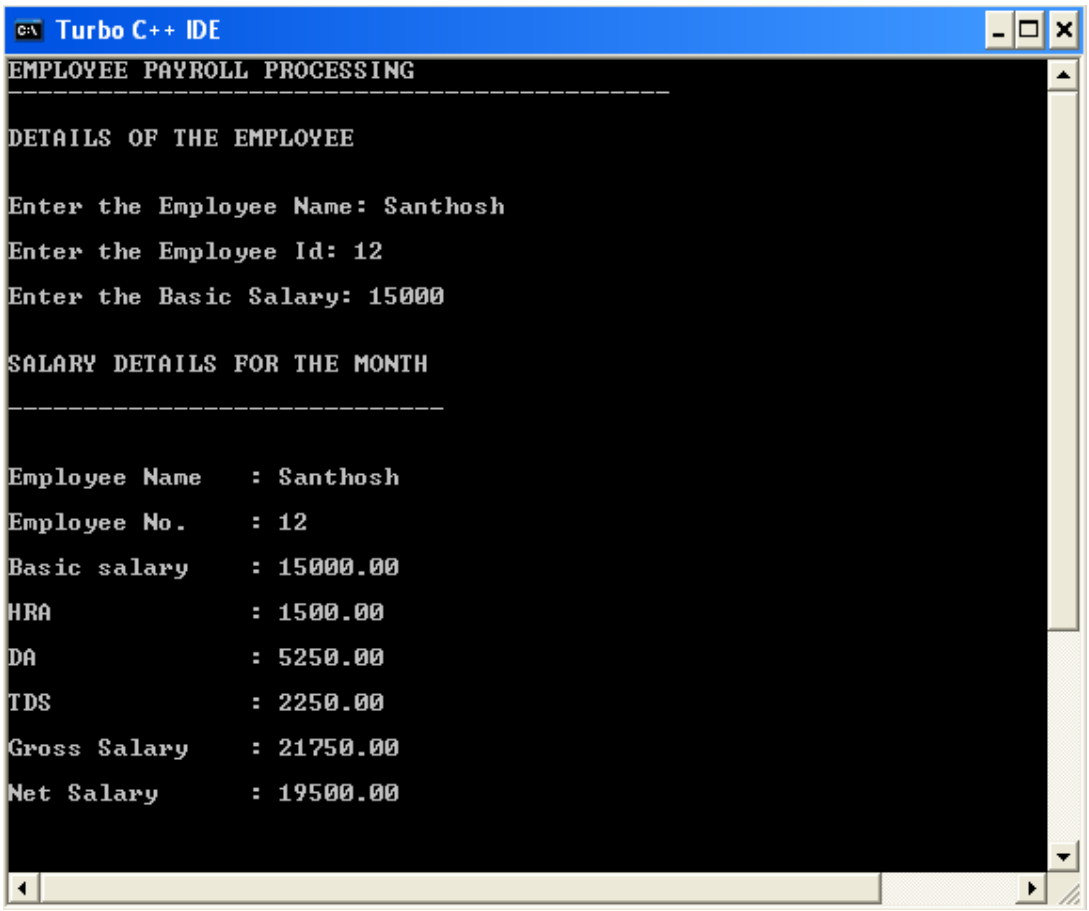
Step 8: Print Employee salary details such as Name, Id, Basic, HRA, DA, TDS, GROSS and NET SALARY.

Step 9: Stop the program.

PROGRAM: (PAYROLL PROCESSING USING UNION)

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
union employee
{
     char ename[30];
     int eno;
     float bsal;
};
void main()
{
     union employee emp; char name[30];
     int id;
     float basic,hra,da,tds,net,gross; clrscr();
     printf("\EMPLOYEE PAYROLL PROCESSING\n");
     printf("\--------------------------------\n");
     printf("\nDETAILS OF THE EMPLOYEE\n\n");
     printf("\nEnter the Employee Name: ");
     scanf("%s",emp.ename);
     strcpy(name,emp.ename);
     printf("\nEnter the Employee Id: ");
     scanf("%d",&emp.eno);
     id=emp.eno;
     printf("\nEnter the Basic Salary: ");
     scanf("%f",&emp.bsal);
     basic=emp.bsal;
     hra=basic*.10;
     da=basic*.35;
     tds=basic*.15;
     gross=basic+hra+da;
     net=gross-tds;
     printf("\n\nSALARY DETAILS FOR THE MONTH\n");
     printf("\n----------------------------\n");
     printf("\n\nEmployee Name\t: %s",name);
     printf("\n\nEmployee No.\t: %d",id);
     printf("\n\nBasic salary\t: %.2f",basic);
     printf("\n\nHRA\t\t: %.2f",hra);
     printf("\n\nDA\t\t: %.2f",da);
     printf("\n\nTDS\t\t: %.2f",tds);
     printf("\n\nGross Salary\t: %.2f",gross);
     printf("\n\nNet Salary\t: %.2f",net);
     getch();
}
```

OUTPUT:



RESULT

Thus a C program to implement the payroll processing using union was executed and the output was obtained

Ex. No: 23

Date :

EMPLOYEE DETAILS USING STRUCTURE

AIM:

To write a C program to print the employee details of employees using structure.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the variables using structure.

Step 3: Read total number of employees n.

Step 4: Read e[i].eno.e[i].ename,e[i].salary

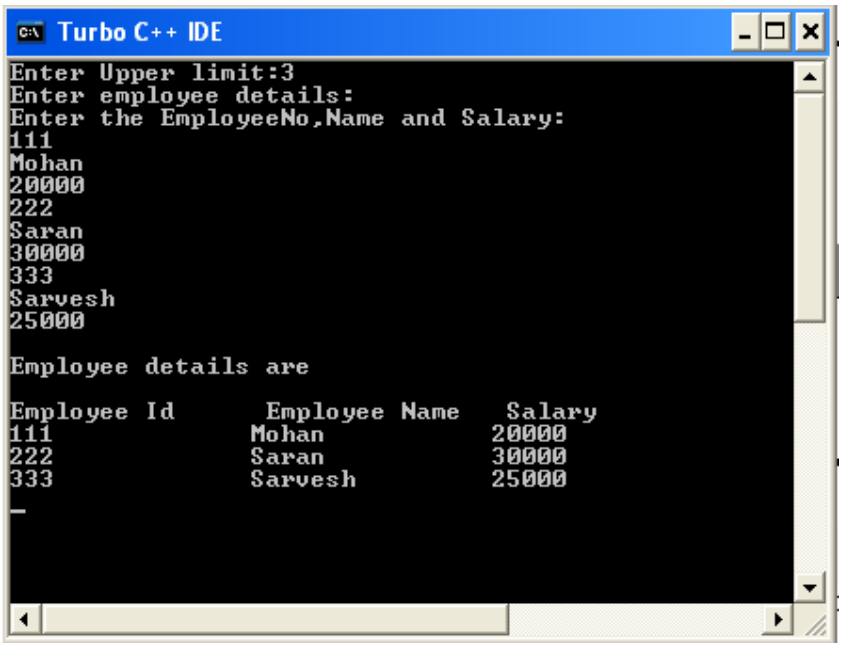
Step5: Print e[i].eno.e[i].ename,e[i].salary

Step 6: Stop the program

PROGRAM: (EMPLOYEE DETAILS USING STRUCTURE)

```
#include<conio.h>
struct emp
{
     int eno;
     char ename[10];
     int salary;
} e[10];
void main()
{
     int i,n;
     clrscr();
     printf("Enter Upper limit:");
     scanf("%d",&n);
     printf("Enter employee details:\n");
     printf("Enter the EmployeeNo,Name and
     Salary:\n");
     for(i=0;i<n;i++)
     scanf("%d%s%d",&e[i].eno,e[i].ename,&e[i].salary
     );
     printf("\n");
     printf("Employee details are \n\n");
     printf("Employee Id \t Employee Name \t
     Salary\n");
     for(i=0;i<n;i++)
     printf("%d\t\t%s\t\t%d\n",e[i].eno,e[i].ename,e[
     i].salary);
     getch();
}
```

OUTPUT:



RESULT

Thus the program to print the employee details using structure is created successfully and the output was verified.

Ex. No: 24

Date :

CALL BY VALUE AND CALL BY REFERENCE

AIM:

To write a C program to swap two numbers using pointers.

ALGORITHM:

Step 1: Start the program

Step 2: Declare the two pointer variables.

Step 3: Get the value for both the variables.

Step 4: Swap the values.

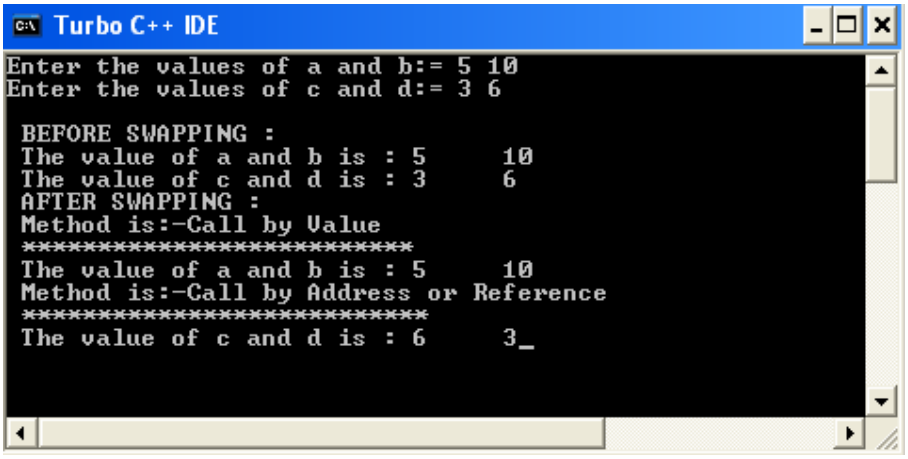
Step5: Print the result.

Step 6: Stop the program

PROGRAM: (CALL BY VALUE AND CALL BY REFERENCE)

```
#include<stdio.h>
#include<conio.h>
void swap(int,int);
void swap1(int*,int*);
void main()
{
     int a,b,c,d;
     clrscr();
     printf("Enter the values of a and b:= ");
     scanf("%d %d",&a,&b);
     printf("Enter the values of c and d:= ");
     scanf("%d %d",&c,&d);
     printf("\n BEFORE SWAPPING : ");
     printf("\n The value of a and b is : %d\t %d ",a,b);
     printf("\n The value of c and d is : %d\t %d ",c,d);
     printf("\n AFTER SWAPPING : ");
     swap(a,b);
     swap1(&c,&d);
     printf("\n Method is:-Call by Value");
     printf("\n **************************");
     printf("\n The value of a and b is : %d\t %d",a,b);
     printf("\n Method is:-Call by Address or Reference");
     printf("\n **************************");
     printf("\n The value of c and d is : %d\t %d",c,d);
     getch();
}
void swap(int c,int d)
{
     int t;
     t=c;
     c=d;
     d=t;
}
void swap1(int *a,int *b)
{
     int t;
     t=*a;
     *a=*b;
     *b=t;
}
```

OUTPUT:



RESULT

Thus, the given program has been executed successfully and the output was verified.