INDIRA GANDHI NATIONAL OPEN UNIVERSITY

MASTER OF COMPUTER APPLICATION

MCSL - 017 (C Language, Digital Circuit and Assembly Language) – Lab

Name	**: Gowtham R**
Enrolment Number	**: 187012107**
Year / Semester	**: I / I**
Study Centre Code	**: 2548**
Study Centre	**: Kongu Engineering College**

DIRECTORATE OF DISTANCE EDUCATION

INDIRA GANDHI NATIONAL OPEN UNIVERSITY

New Delhi

Certified that this is the bonafide record work done by **Mr.R.Gowtham** with Enrolment Number: **187012107** of MCA (Master of Computer Applications) – **I Semester** in the Programming Lab – **MCSL-017** in the directorate of Distance Education, Indira Gandhi National Open University, New Delhi, during the Calendar year 2018 – 2019

Date:

Submitted for MCA Degree course Practical Examination held on **09.01.2019** at the center **Sacred Heart College,Tirupattur, Vellore.**

Date : **Examiners**

1. Name :
   Signature :
2. Name :
   Signature :

C & Assembly LANGUAGE PROGRAMMING- LAB

Index

C AND ASSEMBLY LANGUAGE PROGRAMMING - LAB

SECTION – 1

C Programming Lab

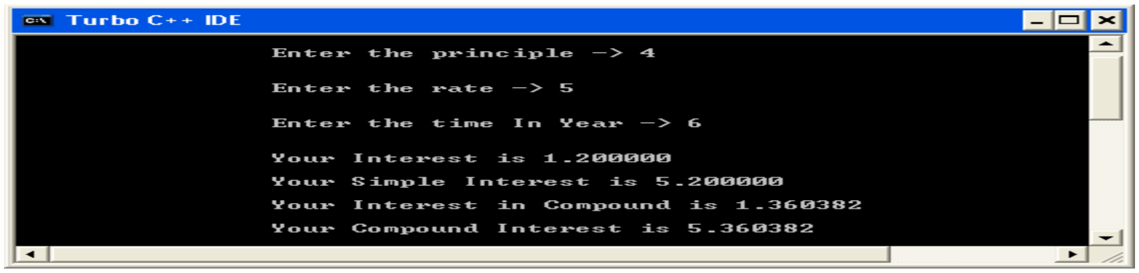
Session 1

Ex 1: Write an interactive program to calculate simple Interest and Compound Interest.

Code:

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

void main()
{
	float pri,amt,rate,si,ci,time,interest,i,j=1;
	clrscr();
	printf("\n\n\t\tEnter the principle -> ");
	scanf("%f",&pri);
	printf("\n\n\t\tEnter the rate -> ");
	scanf("%f",&rate);
	printf("\n\n\t\tEnter the time In Year -> ");
	scanf("%f",&time);
	// -------Programm For Simple Interest---------

	interest=(pri*rate*time)/100;
	si=pri+interest;
	printf("\n\n\t\tYour Interest is %f",interest);
	printf("\n\n\t\tYour Simple Interest is %f",si);

	// ------Programm For Compound Interest--------

	for(i=1; i<=time; i++)
	{
		j=(rate+100)/100*j;
	}
	ci=pri*j;
	interest=ci-pri;
	printf("\n\n\t\tYour Interest in Compound is %f",interest);
	printf("\n\n\t\tYour Compound Interest is %f",ci);
	getch();
}
```

Output:



```
Enter the principle -> 4

Enter the rate -> 5

Enter the time In Year -> 6

Your Interest is 1.200000
Your Simple Interest is 5.200000
Your Interest in Compound is 1.360382
Your Compound Interest is 5.360382
```

Ex 2: Write an interactive program that uses loop to input the income and calculate and report the owed tax amount. Make sure that your calculation is mathematically accurate and that transaction errors eliminated.

Assume that the United States of America uses the following income tax code formula for their annual income:

First US$ 5000 of income : 0% tax
Next US$ 10,000 of income : 10% tax
Next US$ 20,000 of income : 15% tax
An amount above US$35,000 : 20% tax

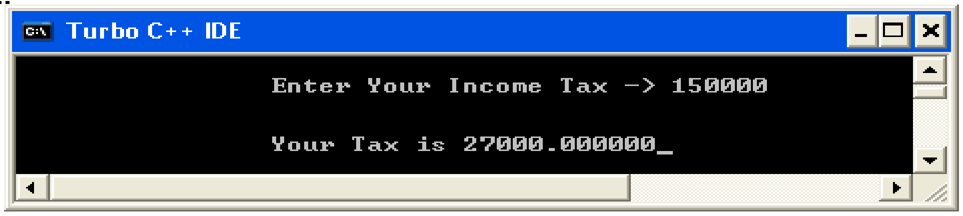
For example, somebody earning US$ 38,000 annually would owe US$5000x0.00+10,000x0.10+20,000x0.15+3000x0.20,
which comes to US$4600.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        float income,i=0,j=0,k=0,tax=0;
        clrscr();
        printf("\n\n\t\tEnter Your Income Tax -> ");
        scanf("%f",&income);

        //--------Calculate The Tax---------

        if(income>35000)
        {
                income=income-35000;
                i=10000*.10;
                j=20000*.15;
                k=income*.20;
                tax=i+j+k;
                printf("\n\n\t\tYour Tax is %f",tax);
        }
        else if(income>20000 && income<=35000)
        {
                income=income-15000;
                i=10000*.10;
                j=income*.15;
                tax=i+j;
                printf("\n\n\t\tYour Tax is %f",tax);
        }
        else if(income>10000 && income<=20000)
        {
                income=income-15000;
                i=10000*.10;
                j=income*.15;
                tax=i+j;
                printf("\n\n\t\tYour Tax is %f",tax);
        }
        else if(income>5000 && income<=10000)
        {
                income=income-5000;
                tax=income*.10;
```

```
            printf("\n\n\t\tYour Tax is %f",tax);
        }
        else
        {
            printf("\n\n\t\t You have no tax");
        }

        getch();

}
```

Output:



```
Turbo C++ IDE

          Enter Your Income Tax -> 150000

          Your Tax is 27000.000000_
```

Ex 3: Write an interactive program that reads in integers until a 0 is entered. If it encounters 0 as input, then it should display:

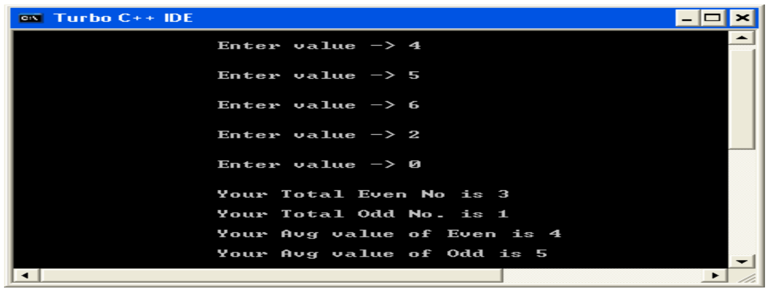
- the total no. of even and odd integers.
- average value of even integers.
- average value of odd integers.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int val,even=0,odd=0,sum_even=0,sum_odd=0,avg_even,avg_odd;
        clrscr();
        first:
                printf("\n\n\t\tEnter value -> ");
                scanf("%d",&val);
                if(val==0)
                {
                        goto last;
                }
                else if(val%2==0)
                {
                        even++;
                        sum_even=sum_even+val;
                        goto first;
                }
                else
                {
                        odd++;
                        sum_odd=sum_odd+val;
                        goto first;
                }
        last:
            avg_even=sum_even/even;
            avg_odd=sum_odd/odd;
            printf("\n\n\t\tYour Total Even No is %d",even);
```

```
                printf("\n\n\t\tYour Total Odd No. is %d",odd);
                printf("\n\n\t\tYour Avg value of Even is %d",avg_even);
                printf("\n\n\t\tYour Avg value of Odd is %d",avg_odd);
        getch();
}
```

Output:



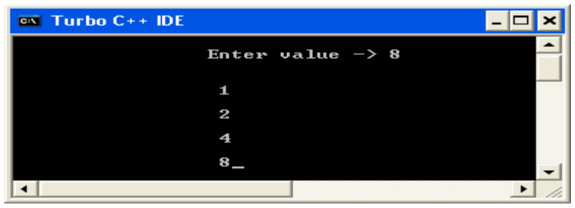
```
Enter value -> 4

Enter value -> 5

Enter value -> 6

Enter value -> 2

Enter value -> 0

Your Total Even No is 3
Your Total Odd No. is 1
Your Avg value of Even is 4
Your Avg value of Odd is 5
```

Ex 4: Write an interactive program to generate the divisors of a given integers.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int val,i;
        clrscr();
        printf("\n\n\t\tEnter value ->");
        scanf("%d",&val);
        for(i=1; i<=val; i++)
        {
                if(val%i==0)
                {
                        printf("\n\n\t\t %d",i);
                }
        }
        getch();
}
```

Output:



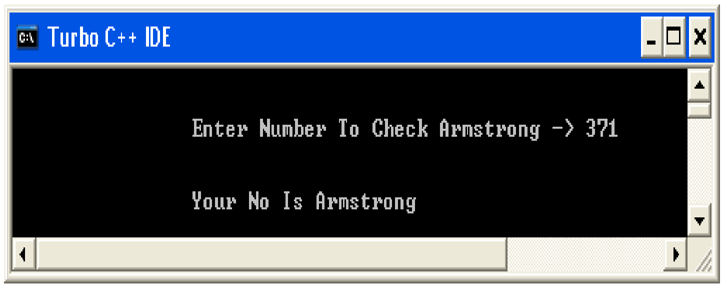
```
Enter value -> 8

 1
 2
 4
 8_
```

Session 2

Ex 5: Write a program to find all Armstrong Number in the range of 0 and 999.
Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
	int val,i,j=0,val1;
	clrscr();
	printf("\n\n\t\tEnter Number To Check Armstrong -> ");
	scanf("%d",&val);
	val1=val;
	while(val>=1)
	{
		i=val%10;
		i=i*i*i;
		j=i+j;
		val=val/10;
	}
		if(val1==j)
		{
			printf("\n\n\t\tYour No Is Armstrong");
		}
		else
		{
			printf("\n\n\t\tYour No is Not Armstrong");
		}
	getch();
}
```

Output:



```
		Enter Number To Check Armstrong -> 371

		Your No Is Armstrong
```

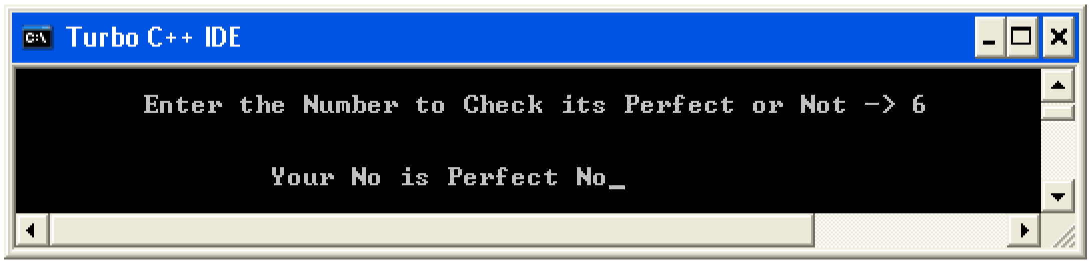
Ex 6: Write a program to check whether a given number is a perfect number or not.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
	int val,i,j=0;
	clrscr();
	printf("\n\n\tEnter the Number to Check its Perfect or Not -> ");
	scanf("%d",&val);
	for(i=1; i<val; i++)
	{
		if(val%i==0)
		{
			j=j+i;
		}
	}
	if(j==val)
```

```
        {
                printf("\n\n\t\tYour No is Perfect No");
        }
        else
        {
                printf("\n\n\t\tYour No is not a perfect no");
        }
        getch();
}
```

Output:



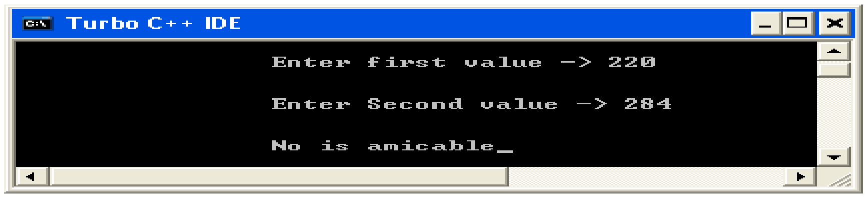
```
        Enter the Number to Check its Perfect or Not -> 6

                Your No is Perfect No_
```

Ex 7: Write a program to check whether given two numbers are amicable numbers or not.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int val,val1,i,j=0,k,l=0;
        clrscr();
        printf("\n\n\t\tEnter first value -> ");
        scanf("%d",&val);
        printf("\n\n\t\tEnter Second value -> ");
        scanf("%d",&val1);
        for(i=1; i<val; i++)
        {
                if(val%i==0)
                {
                        j=j+i;
                }
        }
        for(k=1; k<val1; k++)
        {
                if(val1%k==0)
                {
                        l=l+k;
                }
        }
        if(l==val && j==val1)
        {
                printf("\n\n\t\tNo is amicable");
        }
        else
        {
                printf("\n\n\t\tNot a Amicable");
        }
        getch();
}
```

Output:

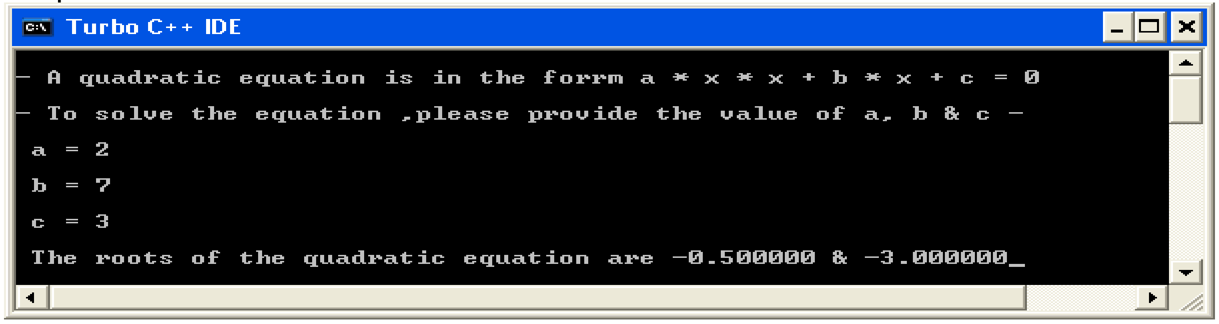


```
Turbo C++ IDE
                    Enter first value -> 220

                    Enter Second value -> 284

                    No is amicable_
```

Ex 8: Write a program to find the roots of a quadratic equation.
Code:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<process.h>
void main()
{
        double a,b,c,d,root1,root2;
        clrscr();
        printf("\n- A quadratic equation is in the forrm a * x * x + b * x + c = 0");
        printf("\n\n- To solve the equation ,please provide the value of a, b & c -");
                printf("\n\n a = ");
                scanf("%lf", &a);
        printf("\n b = ");
        scanf("%lf", &b);
                printf("\n c = ");
                scanf("%lf", &c);
                d=(b*b-4*a*c);
        if(d<0)
        {
                printf("\n Cannot claculate roots, as these would be complex numbers.\n");
                getch();
                exit(0);
        }
                root1=(-b+sqrt(d))/(2.0*a);
                root2=(-b-sqrt(d))/(2.0*a);
                printf("\n The roots of the quadratic equation are %lf & %lf", root1,root2);
        getch();
}
```

Output:



```
Turbo C++ IDE
- A quadratic equation is in the forrm a * x * x + b * x + c = 0
- To solve the equation ,please provide the value of a, b & c -
 a = 2
 b = 7
 c = 3
 The roots of the quadratic equation are -0.500000 & -3.000000_
```

Session 3

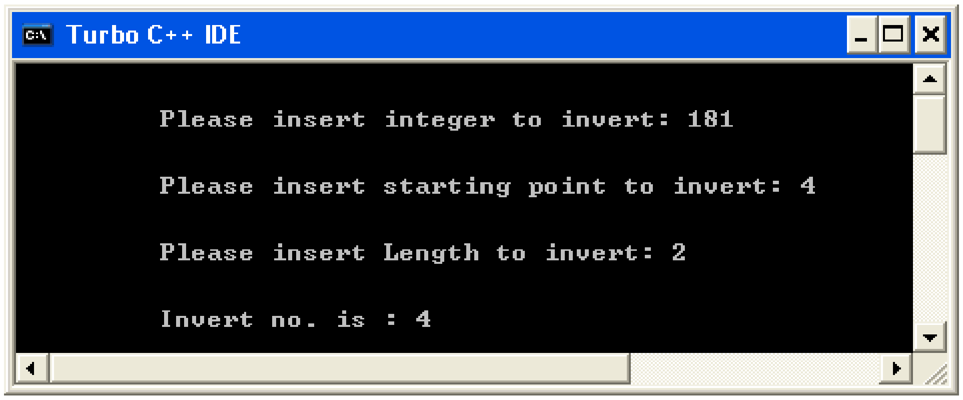
Ex 9: Write a function invert(x,p,n) that returns x with the n bits that begin at position p inverted. You can assume that x,p & n are integer variables and that the function will return an integer.

Code:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
        int intUserInput, intUserInput1, intUserInput2;
        int intCompResult;
        int invert(int, int, int);
        clrscr();
        printf("\n\n\t Please insert integer to invert: ");
        scanf("%d", &intUserInput);

        printf("\n\n\t Please insert starting point to invert: ");
        scanf("%d", &intUserInput);

        printf("\n\n\t Please insert Length to invert: ");
        scanf("%d", &intUserInput2);

        intCompResult=invert(intUserInput, intUserInput1, intUserInput2);
        printf("\n\n\t Invert no. is : %d", intCompResult);
        getch();
}
int invert(int x, int p, int n)
{
        int intbinary[8];
        int i;
        int y;
        int r=0;
        for(i=0;i<8;i++)
        {
                intbinary[i]=0;
        }
        i=0;
        y=0;
        while(x>0)
        {
                intbinary[i]=x%2;
                x=x/2;
                i++;
        }
        for(i=0;i<8;i++)
        {
                if(i==p)
                {
                        for(i=p;i>p-n;i--)
                        {
                                if(intbinary[i]==0)
```

```
                                        {
                                                intbinary[i]=1;
                                        }
                                        else
                                        {
                                        intbinary[i]=0;
                                        }
                                }
                                i=i+n;
                        }
                }
                for(i=0;i<8;i++)
                {
                        r=r+(intbinary[i]*pow(2,i));
                }
                return r;
        }
```

Output:



```
Please insert integer to invert: 181

Please insert starting point to invert: 4

Please insert Length to invert: 2

Invert no. is : 4
```

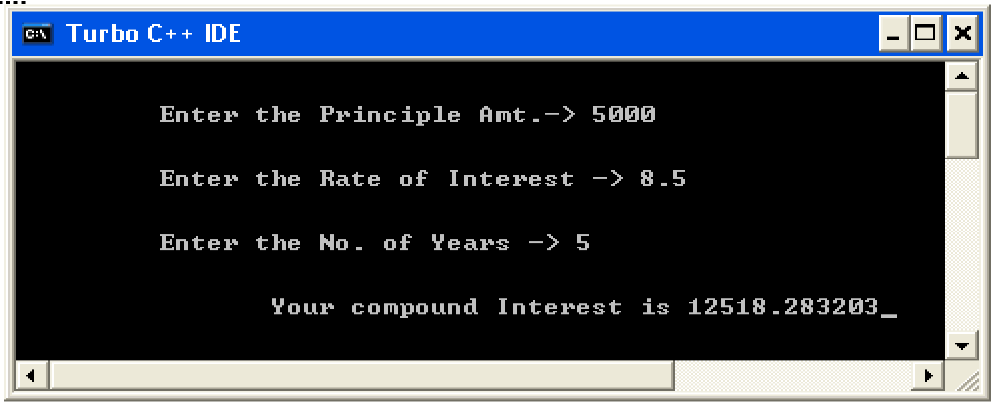
Ex 10: Write a function that calculates the compounded interest amount for a given initial amount, interest rate & no. of years. The interest is compounded annually. The return value will be the interest amount. Use the following function definition: float comp_int_calc(floatint_amt, float rate, int years); Write a program that will accept the initial amount, interest rate & the no. of years and call the function with these values to find out the interest amount and display the returned value.

Code:

```
#include<stdio.h>
#include<conio.h>
float interest(float int_amt,float rate, int year);
void main()
{
        int int_amt,year;
        float rate,amt;
        clrscr();
        printf("\n\n\t Enter the Principle Amt.-> ");
        scanf("%d",&int_amt);
        printf("\n\n\t Enter the Rate of Interest -> ");
        scanf("%f",&rate);
        printf("\n\n\t Enter the No. of Years -> ");
        scanf("%d",&year);
```

```
            amt=interest(int_amt,rate,year);
            printf("\n\n\t\tYour compound Interest is %f",amt);
            getch();
}
float interest(float int_amt,float rate, int year)
{
            float interest,amt,ci;
            float i,j=1;
            for(i=1; i<=year; i++)
            {
                        j=(rate+100)/100*j;
            }
            interest=int_amt*j;
            ci=int_amt+interest;
            return ci;
}
```

Output:



```
        Enter the Principle Amt.-> 5000

        Enter the Rate of Interest -> 8.5

        Enter the No. of Years -> 5

               Your compound Interest is 12518.283203_
```

Ex 11: Break up the program that you wrote to solve above problem into two separate source files. The main function should be in one file & the calculation function must be in another file. And modify the program so that the interest rate is a symbolic constant and is no longer input from the keyboard. And put all the C preprocessor directives into a separate header file that is included in the two program source files.

Code:

file-1.c

```
#include "header.h"
main()
{
            float amt,interest;
            int year;

            float comp_int_calc(float,float,int);

            clrscr();

            printf("Enter the initial amount: ");
            scanf("%f",&amt);

            printf("Enter the Number of years: ");
            scanf("%f",&year);
```

```
        interest=comp_int_calc(amt,roi,year);
        printf("\nThe int is %.2f",interest);
        getch();
}
```

file-2.c

```
#include "header.h"
float comp_int_calc(float x,float y,int z)
{
        float i;
        i=x*pow((1+y/100),2);
        return(i-x);
}
```

header.h

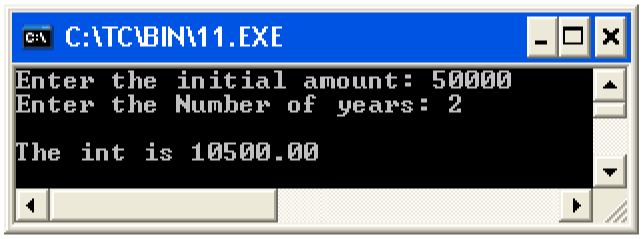
```
#include<stdio.h>
#include<math.h>
#define roi 10
```

Then press Alt+P in Turbo C and enter a project file name, e.g. Q11.prj. Create a new project file of the same name e.g. Q11.prj and enter the following in it-

```
file-1.c
file-2.c
header.h
```

Now compile the project file and the desired output will be obtained.

Output:



```
Enter the initial amount: 50000
Enter the Number of years: 2

The int is 10500.00
```

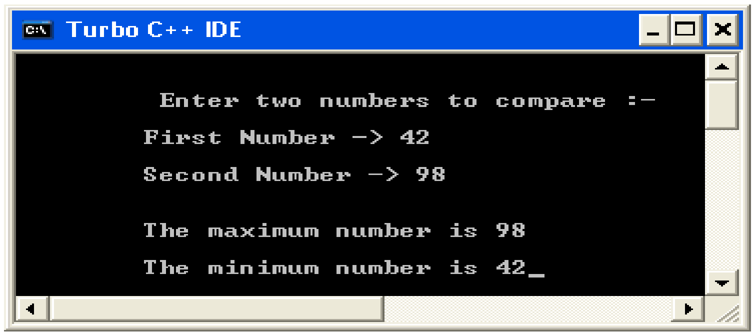
Ex 12: Define two separate macros,MIN & MAX, to find and return, respectively the minimum & maximum of two values. Write a sample program that usesthese macros.

Code:

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

#define min(x,y)(x<y ? x:y)
#define max(x,y)(x>y ? x:y)

void main()
{
        int i,j;
        clrscr();
        printf("\n\n\t Enter two numbers to compare :-");
```

```
            printf("\n\n\tFirst Number -> ");
            scanf("%d", &i);
            printf("\n\tSecond Number -> ");
            scanf("%d", &j);
            printf("\n\n\tThe maximum number is %d", max(i,j));
            printf("\n\n\tThe minimum number is %d", min(i,j));
            getch();
}
```

Output:



```
         Enter two numbers to compare :-
        First Number -> 42
        Second Number -> 98

        The maximum number is 98
        The minimum number is 42_
```

Session 4

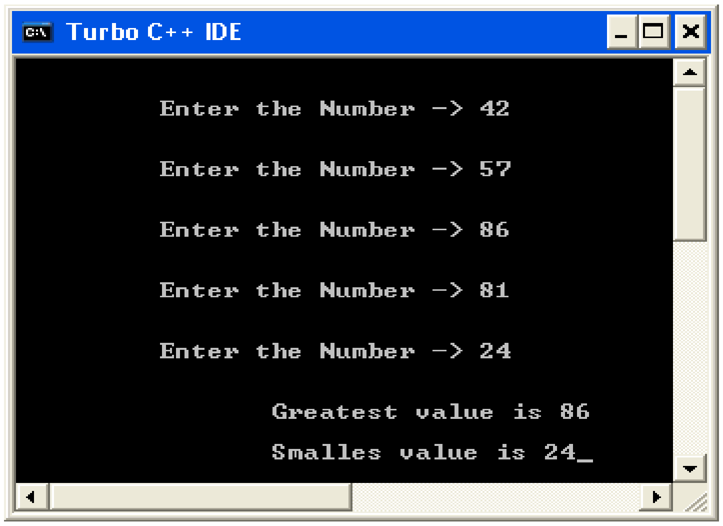
Ex 13: Write a program that will take as input a set of integers and find and display the largest and the smallest values within the input data values.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
            int arr[5],i,j,k,ii;
            clrscr();
            for(i=0; i<5; i++)
            {
                        printf("\n\n\t Enter the Number -> ");
                        scanf("%d",&arr[i]);
            }
            i=arr[0];
            ii=arr[0];
            for(j=0; j<5; j++)
            {
                        if(arr[j]<ii)
                        {
                                    ii=arr[j];
                        }
                        if(arr[j]>i)
                        {
                                    i=arr[j];
                        }
            }
            printf("\n\n\t\tGreatest value is %d",i);
            printf("\n\n\t\tSmalles value is %d",ii);
```

```
        getch();
}
```

Output:



```
        Enter the Number -> 42

        Enter the Number -> 57

        Enter the Number -> 86

        Enter the Number -> 81

        Enter the Number -> 24

                Greatest value is 86
                Smalles value is 24_
```

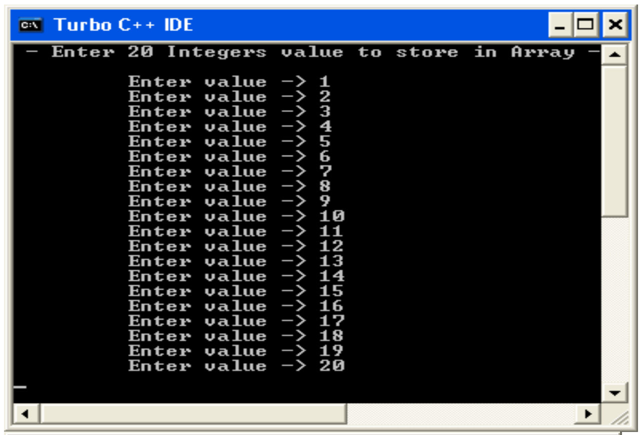
Ex 14: Write an interactive program that will take as input a set of 20 integers and store them in an array and using a temporary array of equal length, reverse the order of the integers & display the values.

Code:

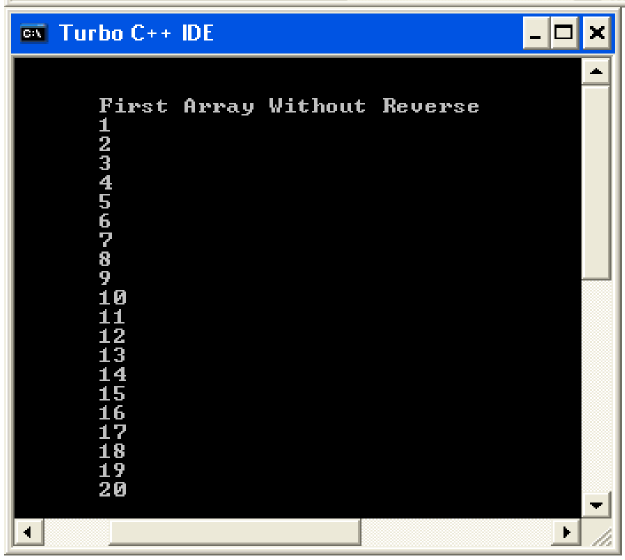
```
#include<stdio.h>
#include<conio.h>
void main()
{
        int arr[20],arr1[20],i,j=0;
        clrscr();
        //--------Input In First Array-------
        printf(" - Enter 20 Integers value to store in Array - \n\n");

        for(i=0; i<20; i++)
        {
                printf("\t Enter value -> ");
                scanf("%d",&arr[i]);
        }

        //-------For Reverse Order--------

        for(i=19; i>=0; i--)
        {
                arr1[j]=arr[i];
                j++;
        }
        getch();
        clrscr();

        //--------Print First Array------------

        printf("\n\n\t\tFirst Array Without Reverse");
```

```
		for(i=0; i<20; i++)
		{
			printf("\n\t\t%d",arr[i]);
		}
		getch();
		clrscr();

		//-------Print Second Array-------------

		printf("\tArray In Reverse Order");
		for(i=0;i<19; i++)
		{
			printf("\n\t\t%d",arr1[i]);
		}

		getch();
}
```

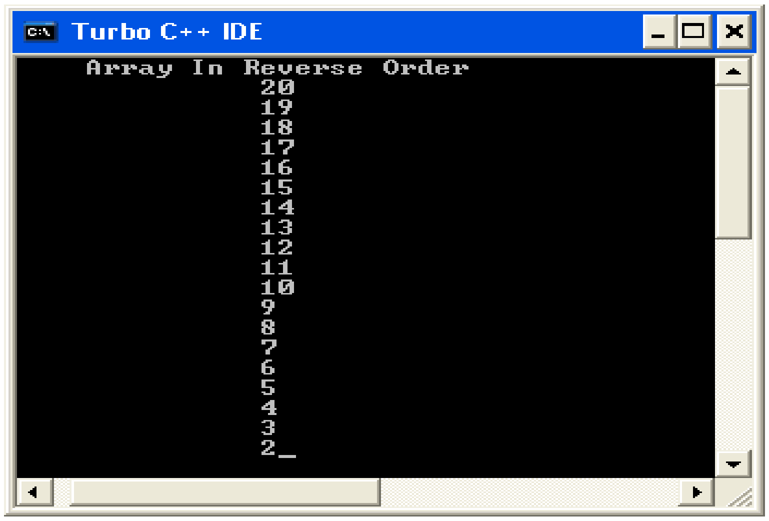
Output:



```
 - Enter 20 Integers value to store in Array -

           Enter value -> 1
           Enter value -> 2
           Enter value -> 3
           Enter value -> 4
           Enter value -> 5
           Enter value -> 6
           Enter value -> 7
           Enter value -> 8
           Enter value -> 9
           Enter value -> 10
           Enter value -> 11
           Enter value -> 12
           Enter value -> 13
           Enter value -> 14
           Enter value -> 15
           Enter value -> 16
           Enter value -> 17
           Enter value -> 18
           Enter value -> 19
           Enter value -> 20
_
```



```

       First Array Without Reverse
       1
       2
       3
       4
       5
       6
       7
       8
       9
       10
       11
       12
       13
       14
       15
       16
       17
       18
       19
       20
```



```
        Array In Reverse Order
                    20
                    19
                    18
                    17
                    16
                    15
                    14
                    13
                    12
                    11
                    10
                    9
                    8
                    7
                    6
                    5
                    4
                    3
                    2_
```

Ex 15: Write an interactive program to do the following computation by providing the option using the switch statement:

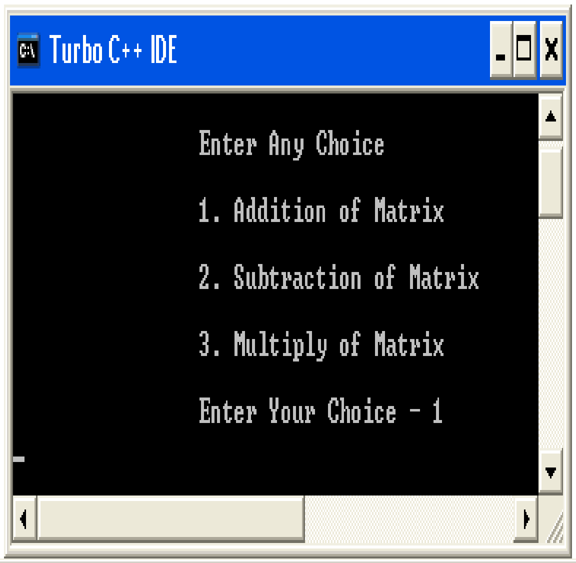
(i) Add two matrices.
(ii) Subtract two matrices.
(iii) Multiply two matrices.

Code:

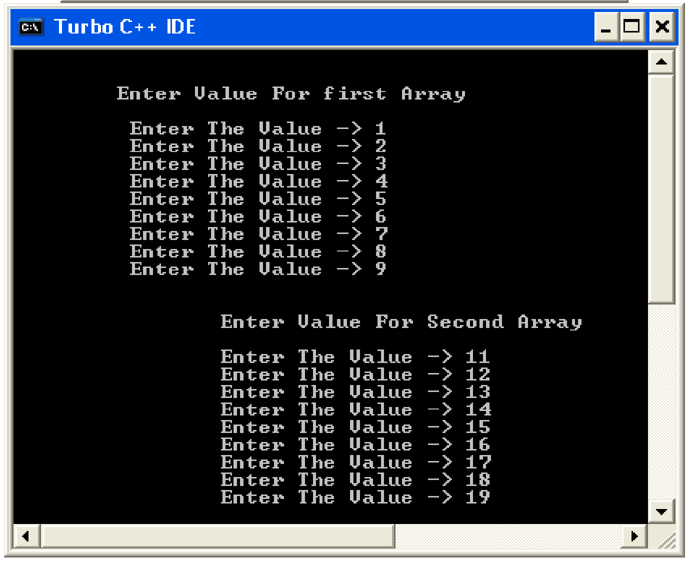
```
#include<stdio.h>
#include<conio.h>
int arr[3][3],arr1[3][3],arr2[3][3],i,j,sum,k;
void input()
{
        printf("\n\n\tEnter Value For first Array\n\n");
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        printf("\t Enter The Value -> ");
                        scanf("%d",&arr[i][j]);
                }
        }
        printf("\n\n\t\tEnter Value For Second Array\n\n");
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        printf("\t\tEnter The Value -> ");
                        scanf("%d",&arr1[i][j]);
                }
        }
}
void display()
{
        getch();
        clrscr();
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        printf("\t%d",arr2[i][j]);
                }
                printf("\n\n");
        }
    }
    void addition()
    {
        input();
    //int i,j;
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            arr2[i][j]=arr[i][j]+arr1[i][j];
        }
    }
}
void sub()
{
    //int i,j;
    input();
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            arr2[i][j]=arr[i][j]-arr1[i][j];
        }
    }
}
void multiply()
{
    input();
    //int i,j,k;
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            arr2[i][j]=0;
            for(k=0; k<3; k++)
            {
            arr2[i][j]=arr2[i][j]+arr[i][k]*arr1[k][j];
            }
        }
    }
}
void main()
{
    int ch;
    clrscr();
    printf("\n\n\t\tEnter Any Choice");
    printf("\n\n\t\t1. Addition of Matrix");
    printf("\n\n\t\t2. Subtraction of Matrix");
    printf("\n\n\t\t3. Multiply of Matrix");
    printf("\n\n\t\tEnter Your Choice - ");
    scanf(" %d",&ch);
```

```
        getch();
        clrscr();
        switch(ch)
        {
                case 1:
                {
                addition();
                display();
                break;
                }
                case 2:
                {
                sub();
                display();
                break;
                }
                case 3:
                {
                multiply();
                display();
                break;
                }
                default:
                {
                printf("\n\n\t\tWrong choice");
                }
        }
        getch();
}
```

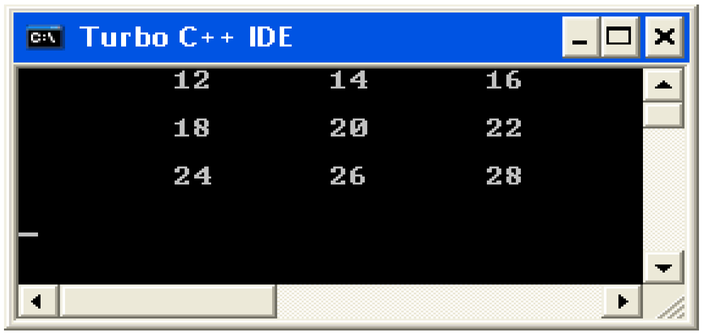
Output:



```
Enter Any Choice

1. Addition of Matrix

2. Subtraction of Matrix

3. Multiply of Matrix

Enter Your Choice - 1
```



```
Enter Value For first Array

 Enter The Value -> 1
 Enter The Value -> 2
 Enter The Value -> 3
 Enter The Value -> 4
 Enter The Value -> 5
 Enter The Value -> 6
 Enter The Value -> 7
 Enter The Value -> 8
 Enter The Value -> 9

        Enter Value For Second Array

        Enter The Value -> 11
        Enter The Value -> 12
        Enter The Value -> 13
        Enter The Value -> 14
        Enter The Value -> 15
        Enter The Value -> 16
        Enter The Value -> 17
        Enter The Value -> 18
        Enter The Value -> 19
```



```
     12     14     16

     18     20     22

     24     26     28
```

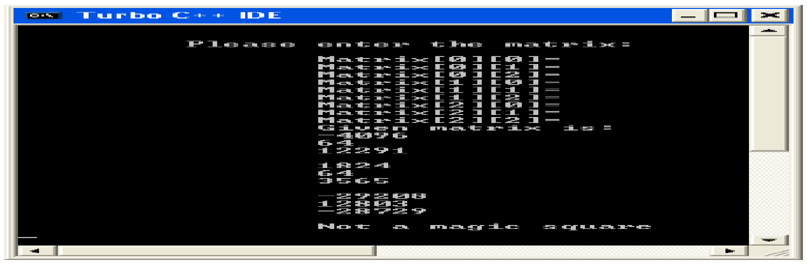
Session 5

Ex 16: Write a program to check if the given matrix is square or not.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
		int intAUserMatrix[3][3];
		int i,j;
		void check_msquare(int a[3][3]);
		clrscr();
		printf("\n\n\t Please enter the matrix:\n");
		for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
			printf("\n\t\tMatrix[%d][%d]=", i,j);
			}
		}
		printf("\n\t\tGiven matrix is: ");
		for(i=0;i<3;i++)
		{
			for(j=0;j<3;j++)
			{
			printf("\n\t\t%d", intAUserMatrix[i][j]);
			}
		printf("\n");
		}
		check_msquare(intAUserMatrix);

		getch();

	}
	void check_msquare(int matrix[3][3])
	{
		int row,col,sumRow[3],sumCol[3];
		for(row=0;row<3;row++)
		{
			for(col=0;col<3;col++)
			{
				sumCol[col]=sumCol[col]+matrix[row][col];
			}
		}
		for(col=0;col<3;col++)
		{
			for(row=0;row<3;row++)
			{
				sumRow[row]=sumCol[row]+matrix[row][col];
			}
		}

		if(sumCol[0]==sumCol[1] && sumCol[0]==sumCol[2] && sumCol[0]==sumRow[0]
	&& sumRow[0]==sumRow[1] && sumRow[0]==sumRow[2])
		{
			printf("\n\n\t\tMagic Square");
		}
		else
		{
			printf("\n\t\tNot a magic square\n");
		}
	}
```

Output:



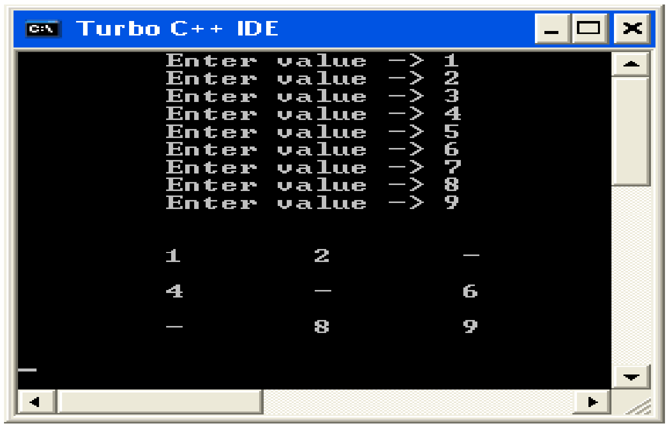
```
                    Please  enter  the  matrix:
                            Matrix[0][0]=
                            Matrix[0][1]=
                            Matrix[0][2]=
                            Matrix[1][0]=
                            Matrix[1][1]=
                            Matrix[1][2]=
                            Matrix[2][0]=
                            Matrix[2][1]=
                            Matrix[2][2]=
                            Given  matrix  is:
                            -4096
                            64
                            12291

                            1824
                            64
                            3565

                            -27208
                            12803
                            -28729

                            Not  a  magic  square
_
```

Ex 17: Write a program to print the upper and lower triangle of matrix.

Code:

```
        #include<stdio.h>
        #include<conio.h>
        void main()
        {
                int arr[3][3],i,j,k=2;
                clrscr();
                for(i=0; i<3; i++)
                {
                        for(j=0; j<3; j++)
                        {
                                printf("\tEnter value -> ");
                                scanf("%d",&arr[i][j]);
                        }
                }
                printf("\n\n");
                for(i=0; i<3; i++)
                {
                        for(j=0; j<=2; j++)
                        {
                                if(arr[i]==arr[k])
                                {
                                printf("\t-");
                                k--;
                                }
                                else
                                {
                                printf("\t%d",arr[i][j]);
                                k--;
                                }
                        }
                        printf("\n\n");
                        k=2;
                }
                getch();
                }
```

Output:

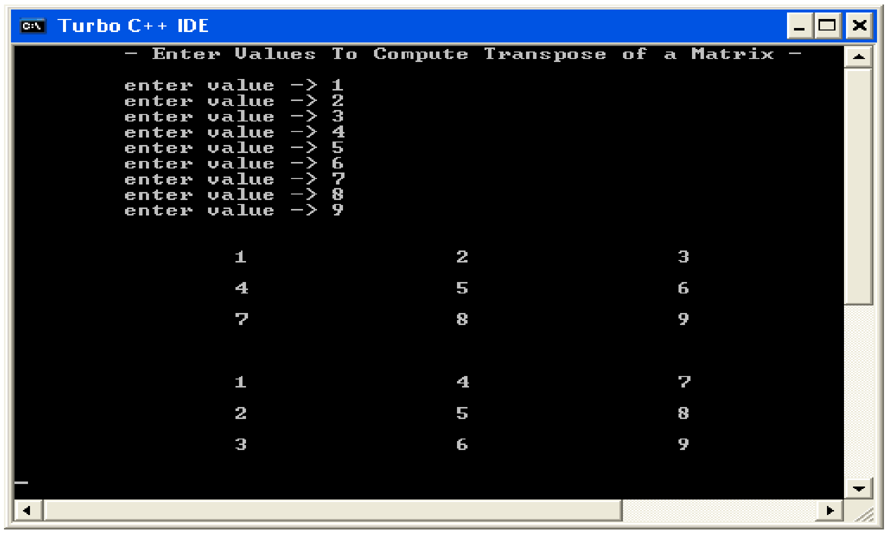


```
Enter value -> 1
Enter value -> 2
Enter value -> 3
Enter value -> 4
Enter value -> 5
Enter value -> 6
Enter value -> 7
Enter value -> 8
Enter value -> 9


1          2          –

4          –          6

–          8          9

_
```

Ex 18: Write a program To compute transpose of a matrix.
Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int arr[3][3],i,j;
        clrscr();
        printf("\t- Enter Values To Compute Transpose of a Matrix - \n\n");
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        printf("\tenter value -> ");
                        scanf("%d",&arr[i][j]);
                }
        }
        printf("\n\n");
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        printf("\t\t%d",arr[i][j]);
                }
        printf("\n\n");
        }
        printf("\n\n");
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        printf("\t\t%d",arr[j][i]);
                }
        printf("\n\n");
        }

        getch();
}
```

Output:



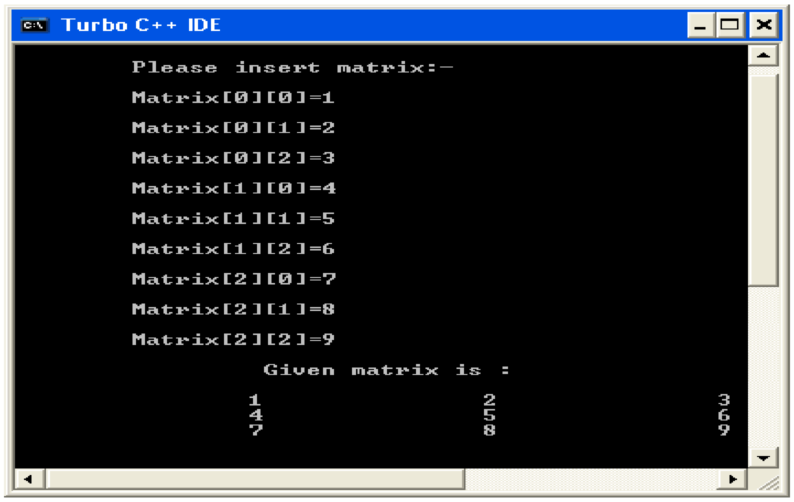
```
      - Enter Values To Compute Transpose of a Matrix -

        enter value -> 1
        enter value -> 2
        enter value -> 3
        enter value -> 4
        enter value -> 5
        enter value -> 6
        enter value -> 7
        enter value -> 8
        enter value -> 9

                1               2               3

                4               5               6

                7               8               9


                1               4               7

                2               5               8

                3               6               9

_
```

Ex 19: Write a program to find the inverse of a Matrix.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int AUserMatrix[3][3];
        int i,j;
        clrscr();
        printf("\n\n\tPlease insert matrix:-\n");
        for(i=0;i<3;i++)
        {
                for(j=0;j<3;j++)
                {
                        printf("\n\tMatrix[%d][%d]=", i,j);
                        scanf("%d", &AUserMatrix[i][j]);
                }
        }
        printf("\n\t\t Given matrix is :\n\n");

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

Output:



```
          Please insert matrix:-
          Matrix[0][0]=1
          Matrix[0][1]=2
          Matrix[0][2]=3
          Matrix[1][0]=4
          Matrix[1][1]=5
          Matrix[1][2]=6
          Matrix[2][0]=7
          Matrix[2][1]=8
          Matrix[2][2]=9
                      Given matrix is :

                      1                   2                   3
                      4                   5                   6
                      7                   8                   9
```

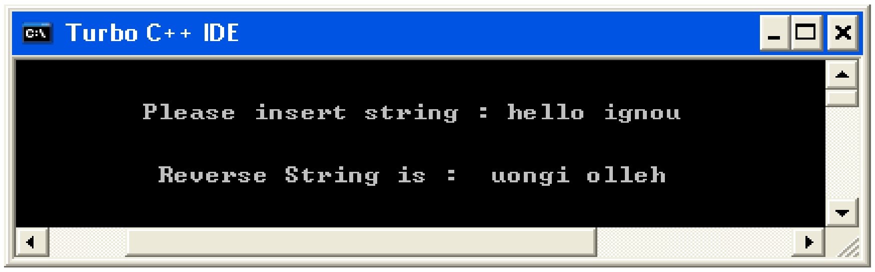
Session 6

Ex 20: Using Recursion, Reverse ‘n’ Characters.

Code:

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

        void reverse(char chrAParam[],int intParamLen)
        {

                if(intParamLen>-1)
                {
                        printf("%c", chrAParam[intParamLen]);
                        intParamLen=intParamLen-1;
                        reverse(chrAParam, intParamLen);
                }
        }

        void main()
        {
                char chrAUserInput[50];
                clrscr();
                printf("\n\n\t\tPlease insert string : ");
                gets(chrAUserInput);
                printf("\n\n\t\t Reverse String is : ");
                reverse(chrAUserInput,strlen(chrAUserInput));
                getch();
        }
```

Output:

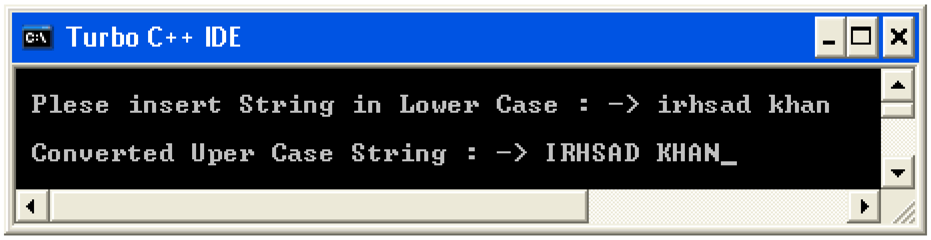

```
Please insert string : hello ignou

Reverse String is :  uongi olleh
```

Session 7

Ex 21: Write a program to convert a given lowercase string to upper case string without using the inbuilt string function.
Code:

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

void main ()
{
        char ChrUserLcase[100];
        int i ;
        clrscr();
        printf ("\t\n Plese insert String in Lower Case : -> ");
        gets(ChrUserLcase) ;
        printf ("\t\n Converted Uper Case String : -> ");
        for (i = 0 ; i <=strlen(ChrUserLcase)-1 ; i++)
        {
        if (ChrUserLcase[i]>=97 && ChrUserLcase[i]<=122 )
        {
        printf ("%c", ChrUserLcase[i]-32) ;
        }
        else
        {
        printf ("%c", ChrUserLcase[i]);
        }
        }
        getch() ;
}
```

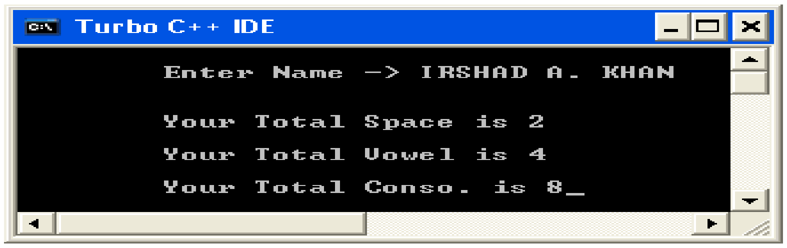
Output:


```
Plese insert String in Lower Case : -> irhsad khan
Converted Uper Case String : -> IRHSAD KHAN_
```

Ex 22: Write a program to count number of vowels, consonants & spaces in a given string.
Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
	char name[20];
	int i,space=0,vowel=0,conso=0;
	clrscr();
	printf("\n\tEnter Name -> ");
	gets(name);
	for(i=0; name[i]!=NULL; i++)
	{
		if(name[i]==' ')
		{
			space++;
		}
		else if(name[i]=='a'|| name[i]=='A'|| name[i]=='e'|| name[i]
			=='E'|| name[i]=='i'|| name[i]=='I'|| name[i]=='o'
			|| name[i]=='O' || name[i]=='u' || name[i]=='U')
		{
			vowel++;
		}
		else
		{
			conso++;
		}
	}
	printf("\n\n\tYour Total Space is %d",space);
	printf("\n\n\tYour Total Vowel is %d",vowel);
	printf("\n\n\tYour Total Conso. is %d",conso);
	getch();
}
```

Output:



```
        Enter Name -> IRSHAD A. KHAN

        Your Total Space is 2
        Your Total Vowel is 4
        Your Total Conso. is 8_
```

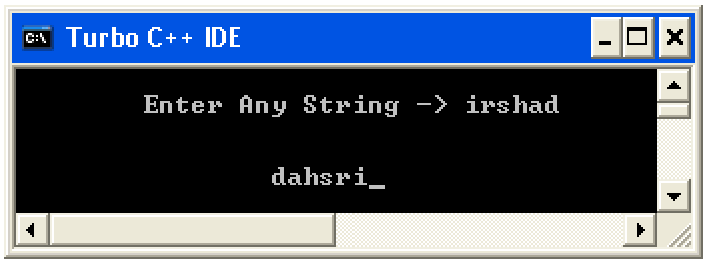
Ex 23: Write a program to input a string and output the reversed string, i.e. if "USF" is input, the program has to output "FSU". You are not to use array notation to access the characters, instead please use pointer notation.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
	char name[20],*p;
	int i,j=0,k;

	clrscr();
		printf("\n\tEnter Any String -> ");
	gets(name);
```

```
        for(i=0; name[i]!=NULL; i++)
        {
                j++;
        }
        p=&name[j-1];
        printf("\n\n\t\t");
        for(i=j-1; i>=0; i--)
        {
                printf("%c",*p);
                p--;
        }
        getch();
}
```

Output:



```
        Enter Any String -> irshad

                dahsri_
```

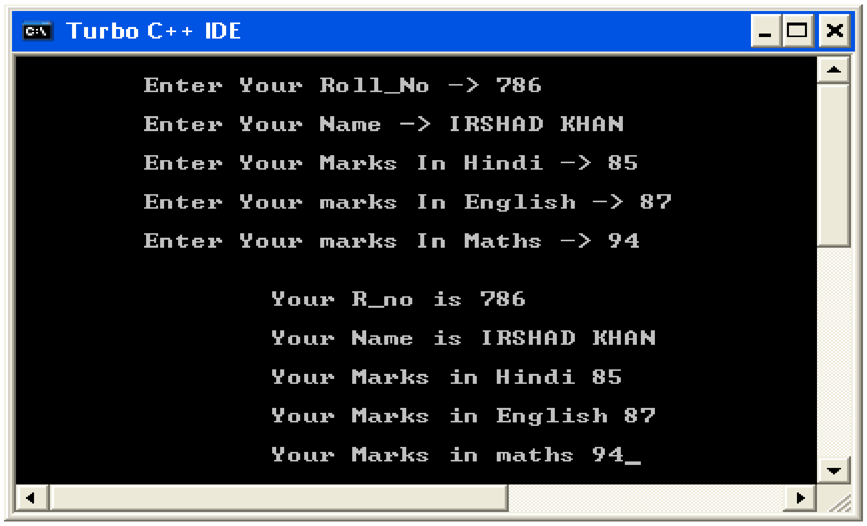
Session 8

Ex 24: Write a program to process the students-evolution records using structures.

Code:

```
        #include<stdio.h>
        #include<conio.h>
        struct student
        {
                char r_no[10],name[20];
                int h,e,m;
        }i;
        void main()
        {
                clrscr();
                printf("\n\tEnter Your Roll_No -> ");
                gets(i.r_no);
                printf("\n\tEnter Your Name -> ");
                gets(i.name);
                printf("\n\tEnter Your Marks In Hindi -> ");
                scanf("%d",&i.h);
                printf("\n\tEnter Your marks In English -> ");
                scanf("%d",&i.e);
                printf("\n\tEnter Your marks In Maths -> ");
                scanf("%d",&i.m);


                printf("\n\n\t\tYour R_no is %s",i.r_no);
                printf("\n\n\t\tYour Name is %s",i.name);
                printf("\n\n\t\tYour Marks in Hindi %d",i.h);
                printf("\n\n\t\tYour Marks in English %d",i.e);
                printf("\n\n\t\tYour Marks in maths %d",i.m);
                getch();
        }
```

Output:



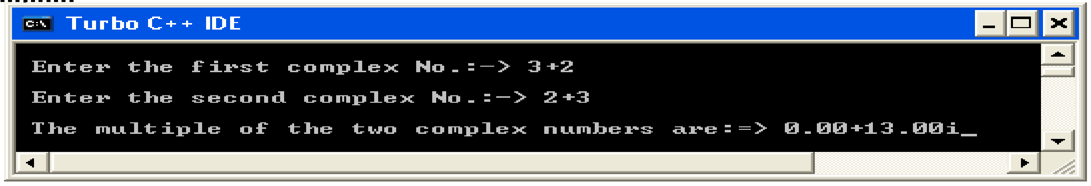
```
        Enter Your Roll_No -> 786
        Enter Your Name -> IRSHAD KHAN
        Enter Your Marks In Hindi -> 85
        Enter Your marks In English -> 87
        Enter Your marks In Maths -> 94

                Your R_no is 786
                Your Name is IRSHAD KHAN
                Your Marks in Hindi 85
                Your Marks in English 87
                Your Marks in maths 94_
```

Ex 25: Define a structure that will hold the data for a complex number. Using this structure, please write a program that will input two complex numbers and output the multiple of two complex numbers. Use double variables to represent complex number components.

Code:

```
#include<stdio.h>
#include<conio.h>
        typedef struct
        {
                double rl;
                double im;
        }
complex;

void main()
{
        complex a,b,c;
        clrscr();
        printf("\n\n Enter the first complex No.:-> ");
        scanf("%lf%lf",&a.rl,&a.im);
        printf("\n Enter the second complex No.:-> ");
        scanf("%lf%lf",&b.rl,&b.im);
        c.rl=(a.rl*b.rl)-(a.im*b.im);
        c.im=(a.rl*b.im)+(b.rl*a.im);
        printf("\n The multiple of the two complex numbers are:=> %.2lf\+%.2lfi",c.rl,c.im);
}
```

Output:



```
Enter the first complex No.:-> 3+2
Enter the second complex No.:-> 2+3
The multiple of the two complex numbers are:=> 0.00+13.00i_
```

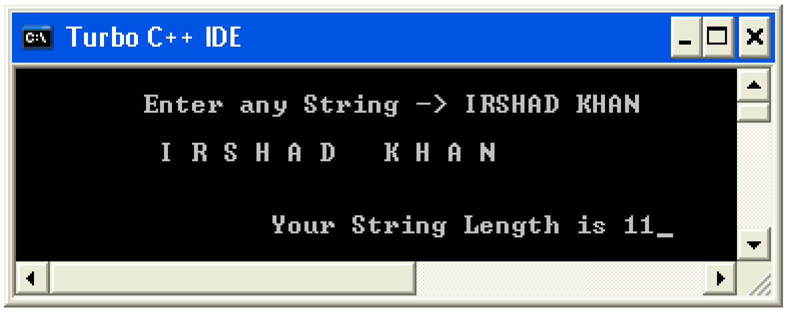
Session 9

Ex 27: Write a function that will return the length of a character string. You are not allowed to use the strlen C library function.

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        char name[20],*p;
        int i,j=0,k;
        clrscr();
        printf("\n\tEnter any String -> ");
        gets(name);
        p=&name[0];
        printf("\n\t");
        for(i=0; name[i]!=NULL; i++)
        {
                printf(" %c",*p);
                p++;
                j++;

        }
        printf("\n\n\n\t\tYour String Length is %d",j);
        getch();
}
```

Output:



```
        Enter any String -> IRSHAD KHAN
         I R S H A D   K H A N

                Your String Length is 11_
```

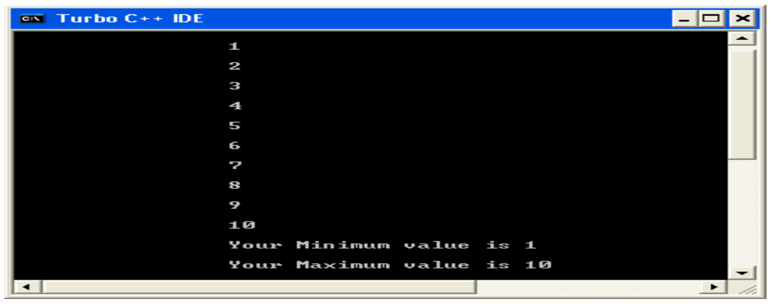
Ex 29: Write a sample program that uses this function to find the display the minimum and the maximum values of an array of integers. Use an array of 10 integers. You can either use scanf to input the values into that array or initialize the array with values in the program itself.

Code:

```
#include<stdio.h>
#include<conio.h>
int arr[10],i,val,val1;
void input()
{
        for(i=0; i<10; i++)
        {
                printf("\n\tEnter value -> ");
                scanf("%d",&arr[i]);
        }
```

```
		clrscr();
		getch();
	}
	void display()
	{
		for(i=0; i<10; i++)
		{
			printf("\n\n\t\t%d",arr[i]);
		}

	}
	void min()
	{
		val=arr[0];
		for(i=0; i<10; i++)
		{
			if(val>arr[i])
			{
				val=arr[i];
			}
		}
		printf("\n\n\t\tYour Minimum value is %d",val);
	}
	void max()
	{
		val1=arr[0];
		for(i=0; i<10; i++)
		{
			if(arr[i]>val1)
			{
				val1=arr[i];
			}
		}
		printf("\n\n\t\tYour Maximum value is %d",val1);
	}
	void main()
	{
		clrscr();
		input();
		display();
		min();
		max();
		getch();
	}
```

Output:



```
		1
		2
		3
		4
		5
		6
		7
		8
		9
		10
		Your Minimum value is 1
		Your Maximum value is 10
```

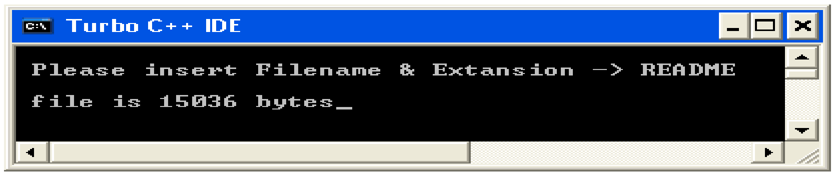
Session 10

Ex 30: Write a program that prompts the user the name of a file and then counts and displays the number of bytes in the file. And create a duplicate file with the word '.backup' appended to the file name. Please check whether file was successfully opened, and display an error message,if not.

Code:

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<string.h>
void main()
{
        int IntFileSize=0,i;
        char chUserFile[20],c;
        FILE*F1User,*F1Bak;
        clrscr();
        printf("\n Please insert Filename & Extansion -> ");
        scanf("%s", chUserFile);

        F1User=fopen(chUserFile,"r");
        if(F1User==NULL)
        {
                printf("File does not exist or File I/O Error");
                getch();
                exit(0);
        }
        strcat(chUserFile,".backup");
        F1Bak=fopen(chUserFile,"w");
        if(F1User==NULL)
        {
                printf("File I/O Error!");
                getch();
                exit(0);
        }
        while((c=getc(F1User))!=EOF)
        {
                putc(c,F1Bak);
                IntFileSize++;
        }

        printf("\n file is %d bytes", IntFileSize);
        fclose(F1User);
        fclose(F1Bak);
        getch();
}
```

Output:



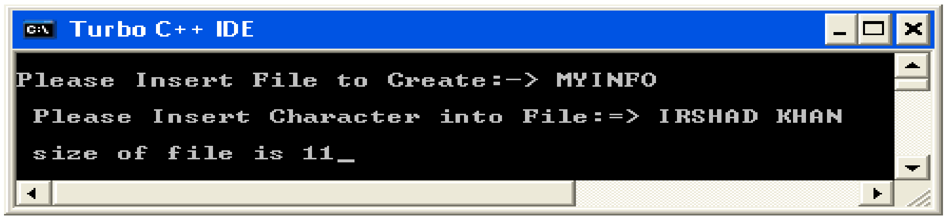
```
Please insert Filename & Extansion -> README
file is 15036 bytes_
```

Ex 31: Write a program to create a file, open it, type-in some character and count the no. of char. in file.

Code:

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<string.h>
void main()
{
	int IntFileSize=0;
	FILE *FIUser;
	char chUserFile[20];
	char chUserInput[100];
	clrscr();
	printf("\nPlease Insert File to Create:-> ");
	gets(chUserFile);

	FIUser=fopen(chUserFile,"w");

	if(FIUser==NULL)
	{
		printf("File Creation Error");
		getch();
		exit(0);
	}
	printf("\n Please Insert Character into File:=> ");
	gets(chUserInput);
	fputs(chUserInput,FIUser);
	fclose(FIUser);
	FIUser=fopen(chUserFile,"r");
	while(getc(FIUser)!=EOF)
	{
		IntFileSize++;
	}
	printf("\n size of file is %d",IntFileSize);
	fclose(FIUser);
	getch();
}
```

Output:



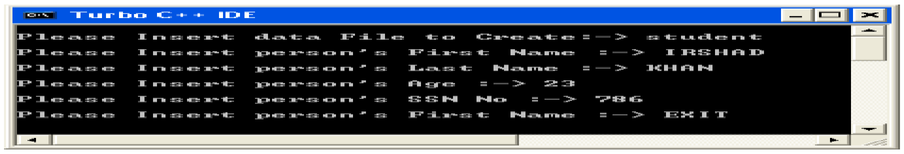
```
Please Insert File to Create:-> MYINFO
 Please Insert Character into File:=> IRSHAD KHAN
 size of file is 11_
```

Ex 32: Write a program that will input a person's first name,last name, SSN number and age and write the information to a data file. One person's information should be in a single line. Use the function fprintf to write to the data file. Accept the information & write the data within a loop. Your program should exit the loop when the word 'EXIT' is entered for the first name. Remember to close the file before terminating the program.

Code:

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <string.h>
struct Datastru{
char fname[20] ;
char lname[20] ;
int SSNno ;
int age ;
}

main ()
{
struct Datastru StPerson ;

FILE *FlUser;
char chUserFile[20] ;
char chUserInput[100] ;

int i;
        printf("\nPlease Insert data File to Create:-> ") ;
        scanf("%s",chUserFile);

FlUser = fopen(chUserFile,"w");
if(FlUser == NULL)
{
        printf("File Creation Error");
        getch() ;
        exit(0);
}

while (strcmp(StPerson.fname,"EXIT") != 0 )
{
        printf("\nPlease Insert person's First Name :-> ") ;
        scanf("%s",&StPerson.fname);
        if(strcmp(StPerson.fname,"EXIT") == 0 )
        {
        fclose(FlUser) ;
        exit(0);
        }
        fprintf(FlUser,"%s",&StPerson.fname) ;
        printf("\nPlease Insert person's Last Name :-> ") ;
        scanf("%s",&StPerson.lname);
        fprintf(FlUser,"%s",&StPerson.lname) ;
        printf("\nPlease Insert person's Age :-> ") ;
        scanf("%d",&StPerson.age);
        fprintf(FlUser,"%d",StPerson.age) ;
        printf("\nPlease Insert person's SSN No :-> ") ;
        scanf("%d",&StPerson.SSNno);
        fprintf(FlUser,"%d",StPerson.SSNno) ;
        fprintf(FlUser,"\n") ;
}
fclose(FlUser) ;
}
```

Output:



```
Please Insert data File to Create:—> student
Please Insert person's First Name :—> IRSHAD
Please Insert person's Last Name :—> KHAN
Please Insert person's Age :—> 23
Please Insert person's SSN No :—> 786
Please Insert person's First Name :—> EXIT
```

SECTION – 2

DIGITAL LOGIC CIRCUITS

Session 1

Ex 1: Design and implement the Exclusive-OR gate using AND, OR and NOT gates.

Ans:

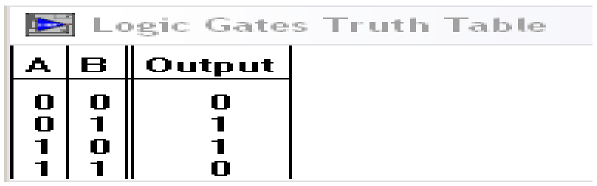
Step 1:

Circuit specification:-

Exclusive or is a combinational circuit the Forms the ex-or operation on the two input values x and y.

Input: Two bits (A, B)
Output: Output= A (+) B

Step 2: Truth Table



Logic Gates Truth Table

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Step 3: Minterm t= F1 (1, 2)

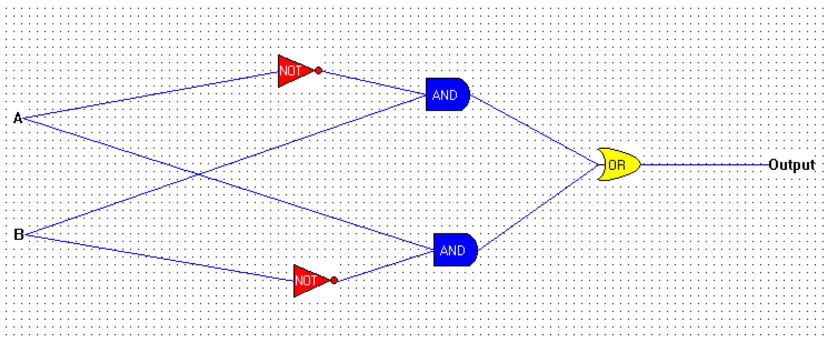
Step 4: Karnaugh maps

0	1
1	

Step5: expression

Output = A (+) B

Step 6: Circuit



Ex 2: Design an "Alarm circuit" using only OR gate in which, if 'doors' OR 'windows' Or 'Fire alarm' is activated and then alarm sound should start.

Ans:

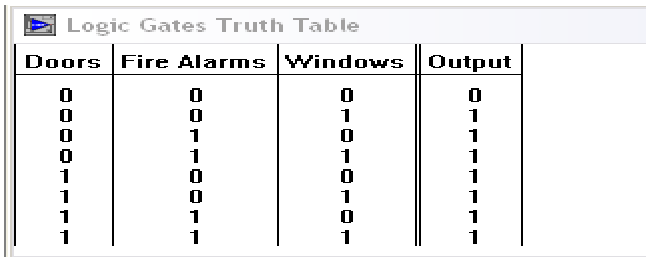
Step1: Specification

Alarm circuit is a combination circuit that forms output a if 'doors' OR 'windows' Or 'Fire alarm' are activated by setting the corresponding bit 1.

Input: 3 input bits ('d','w','f')

Output: 1 bit

Step2: Truth table


Logic Gates Truth Table

Doors	Fire Alarms	Windows	Output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Step 3: identifying Minterms

Output= f+d+w

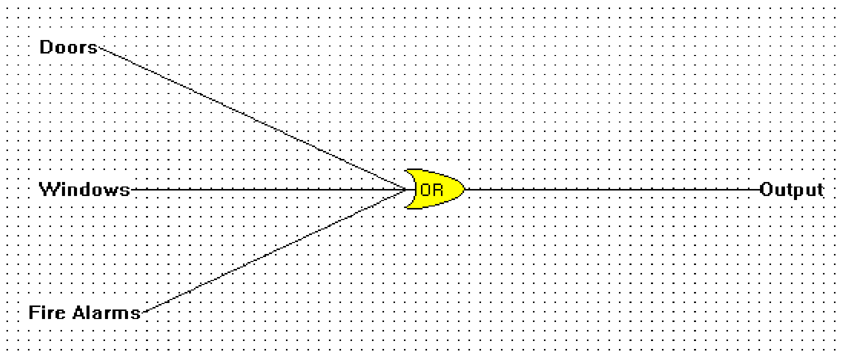
Step 4: K-map

0	1	1	1
1	1	1	1

Step5: Expression

Output= f+d+w

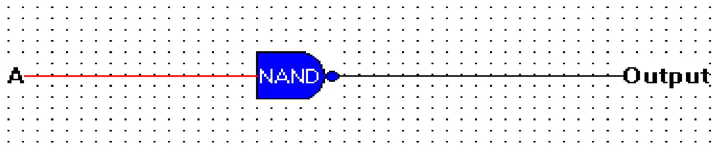
Step6: Circuit



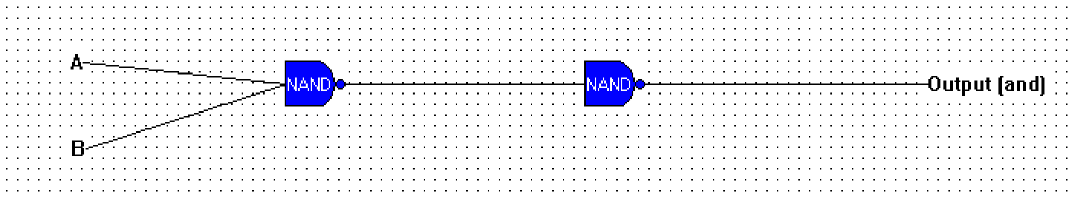
Ex 3: We know NAND gate is universal gate but we need proof, so Design other gates like OR, NOR, AND and NOT using only NAND gates.

Ans:

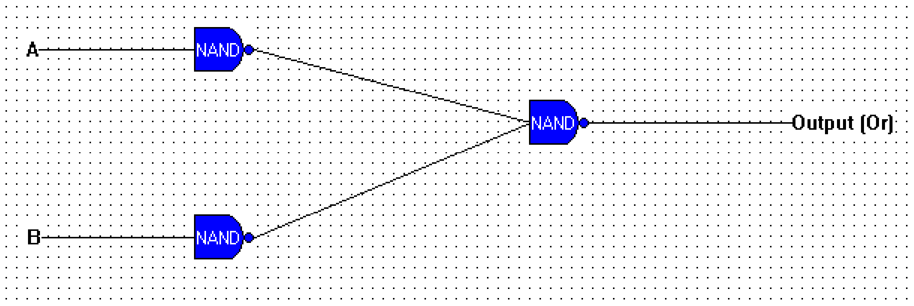
(A) NOT gate using NAND



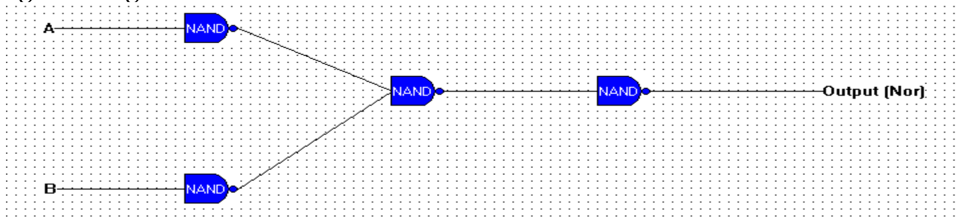
(B) AND gate using NAND



(C) OR gate using NAND



(D) NOR gate using NAND



Ex 4: Design a digital circuit whose output is equal to 1 if the majority of inputs are 1's . The output is 0 otherwise.

Ans:

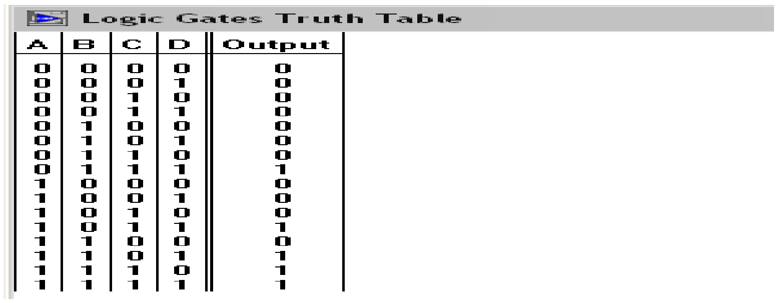
Step1: Specification

Digital circuit whose output is equal to 1 if the majority of inputs are 1's The output is 0 otherwise.

Inputs: 4 bits (a, b, c, d)

Output: 1 bit

Step2: Truth Table



Logic Gates Truth Table

A	B	C	D	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Step 3: Minterms

Output = F (13, 14, 15)

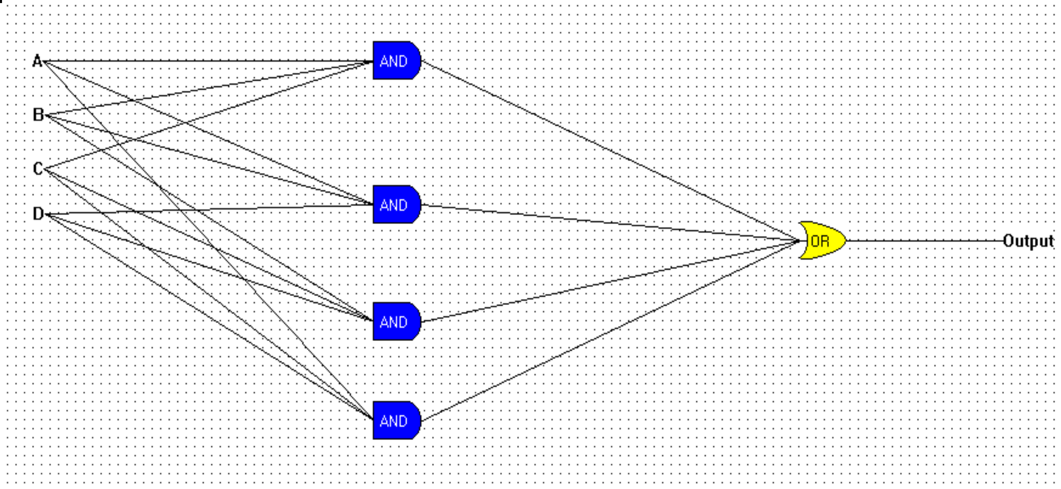
Step4: K-map

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

Step 5: expression

Output = abc+abd+bcd+acd

Step6: Circuit



Ex 5: Design the following digital circuit

Ans:

1) Half adder

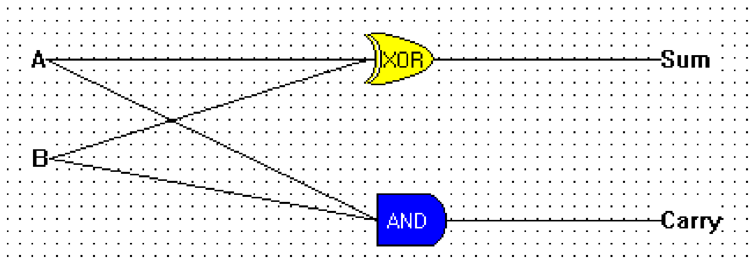
A half adder circuit takes 2 binary input and gives its sum. The input is 2 bits are a and b the outputs are its sum and carry.

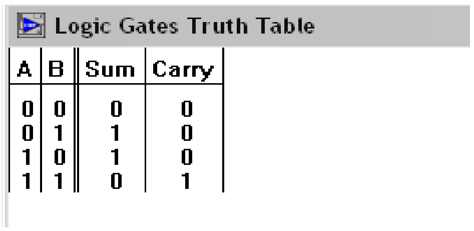
Step 1: Specification

Inputs: 2 bits

Outputs: Sum and Carry

Step2: Circuit and Truth table





Logic Gates Truth Table

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

2) Half Subtractor

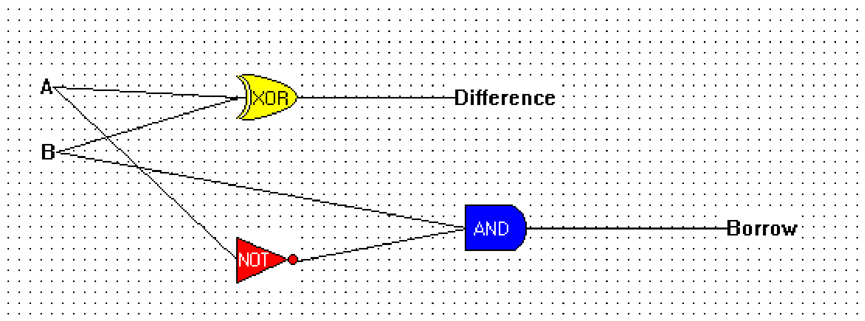
A half subtractor circuit takes 2 binary input and gives its difference. The input is 2 bits are a and b the outputs are its difference and borrow

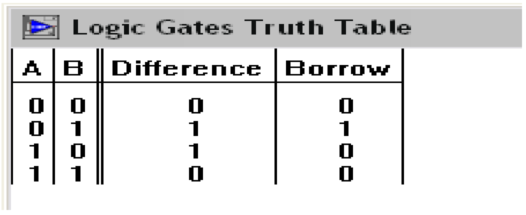
Step 1: Specification

Input: 2 bits

Output: Difference and Borrow

Step 2: Circuit and Truth table





Logic Gates Truth Table

A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

3) Full Subtractor

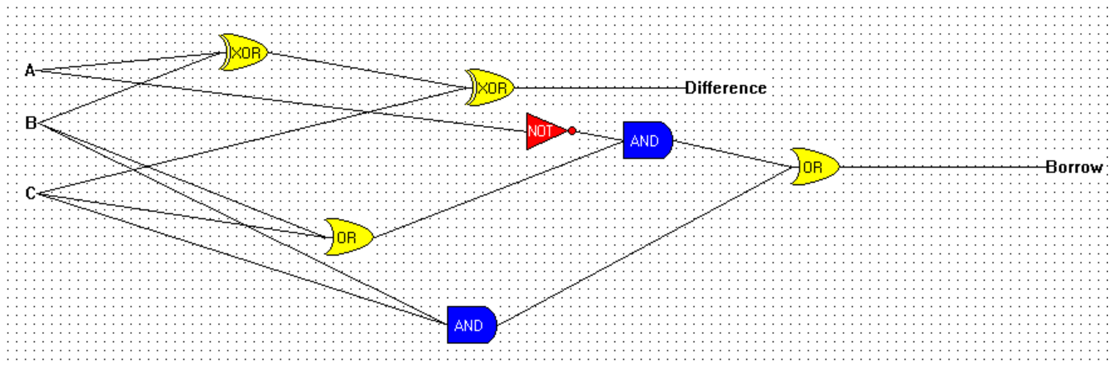
A full subtractor is a combinational circuit that performs a subtraction between two bits taking into account that a one may be borrowed by a lower significant bit, the circuit has 3 inputs A, B and C. and 2 outputs Difference and Borrow.

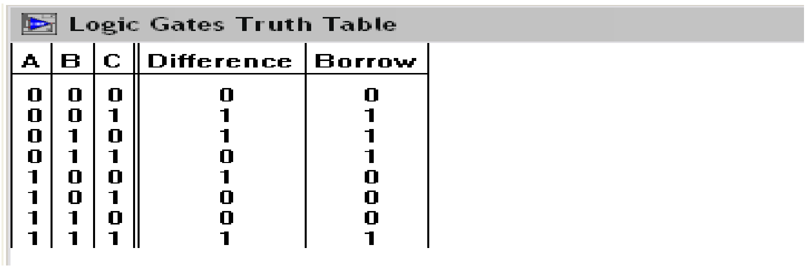
Step1. Specification

Inputs: A, B and C

Outputs: Difference and Borrow

Step2: Circuit and truth table





Logic Gates Truth Table

A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Ex 7: Design a combinational circuit that takes 3-bit number and the output of that circuit should be the square of the input.

Ans:

Step1: Specification

Square of the number is a combinational circuit that can be obtained by taking 3 bits inputs and 6 bit outputs.

Step 2: Truth table

A	B	C	O1	O2	O3	O4	O5	O6
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

Step 3: Identifying Minterms

O1=F1 (6, 7)

O2=F2 (4, 5, 7)

O3=F3 (3, 5)

O4=F4 (2, 6)

O5=0

O6=F6 (1, 3, 5, 7)

The Boolean functions for the three inputs and 6 outputs are derived as follows:-

For F1 (6, 7)
O1=AB

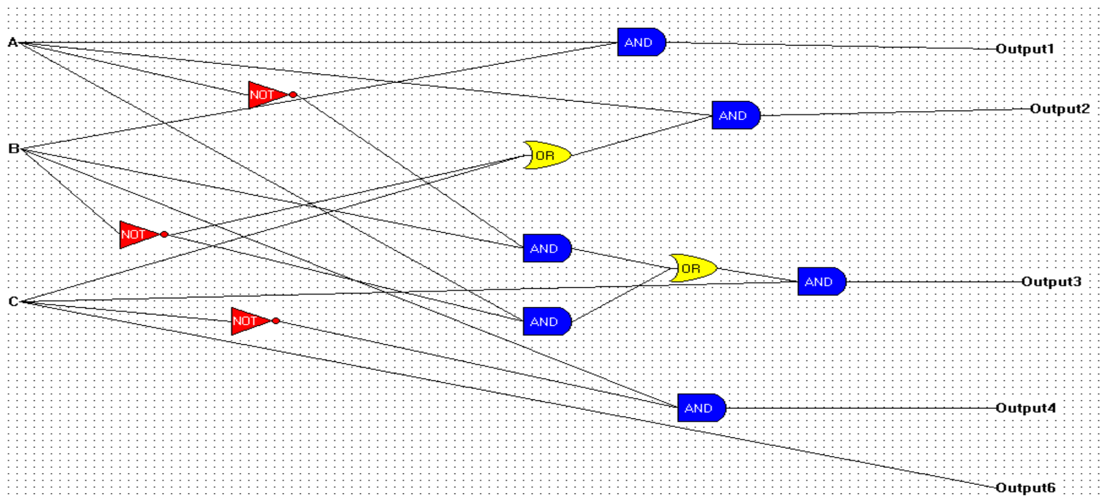
For F2 (4, 5, 7)
O2=AB`+AC

For F3 (3, 7)
O3=A`BC+AB`C

For F4 (2, 6)
O4=BC`

For F6 (1, 3, 5, 7)
O6=C

Step4: Circuit



Ex 8: Design a combinational circuit where input is a 4 bit number and output is it's 2's complement.

Ans: Step1:

Inputs= A, B, C, D
Output=Q1, Q2, Q3, Q4 (2's complement)

Step2: Truth table

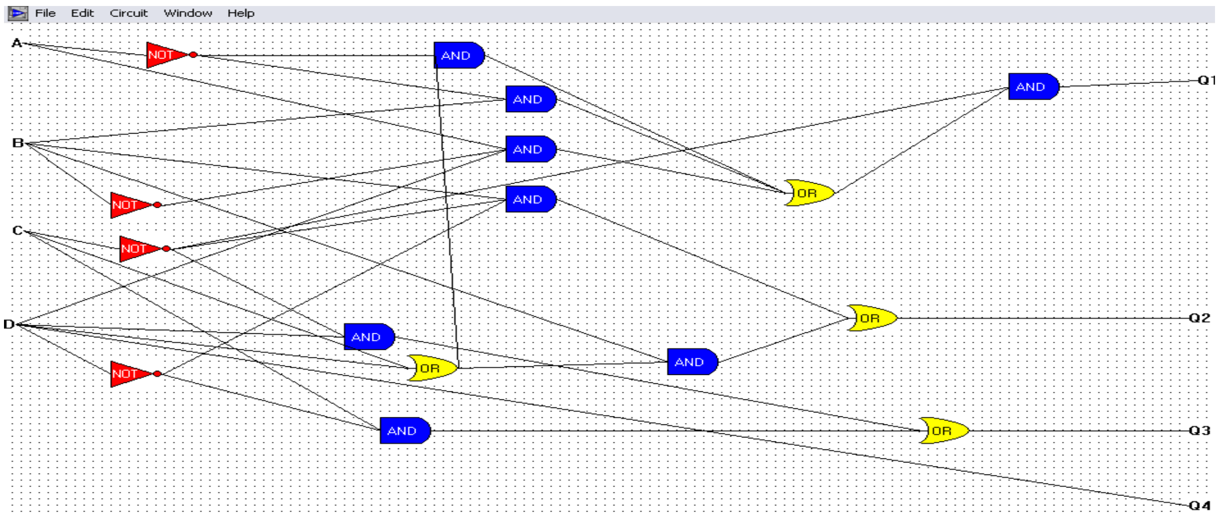
A	B	C	D	Q1	Q2	Q3	Q4
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	1	0	0	0	1	0	0

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

Step3: K-map

Q1=A`D+A`C+A`BC`+AB`C`D`
Q2=BC`D`+B`D+B`C
Q3=C`D+CD`
Q4=D

Step 4: Circuit



Ex 9: Design an encoder circuit, which will convert decimal number to binary.

Ans:

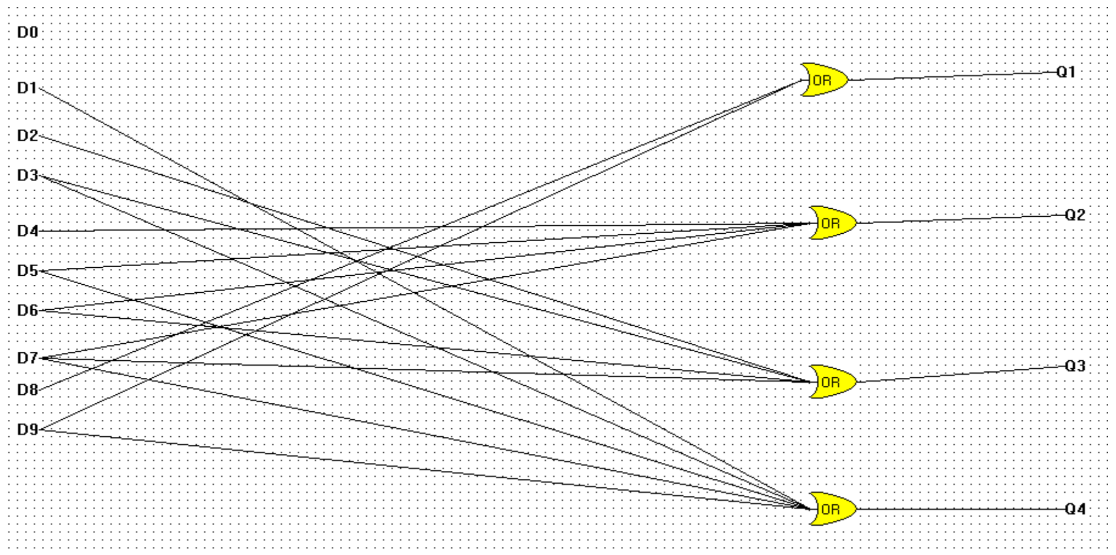
An encoder is a circuit that encodes a particular input to a different format.
A Decimal to binary encoder constructed below

Truth table:

D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	O1	O2	O3	O4
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

Q1=D8+D9
Q2=D4+D5+D6+D7
Q3=D2+D3+D6+D7
Q4=D1+D3+D5+D7+D9

Circuit:



Session 2

Ex 10: Design Sequential Circuit of clocked RS flipflop with 4 NAND gates.

Ans:

The circuit has R and S inputs and a clock input. This latch flip flop is activated by a positive level on the clock input.

If clock = 0, Output Q, Q`= Hold (nochange)
If clock = 1, R=0, S=1,Q=1 State = Set
If clock =1, R=1, S=0, Q`=1 State= Reset
If clock =1, R=0, S=0, State = Hold (no change)

Ex 11: Design Sequential Circuit of Clocked D flip flop with AND and NOR gates.

Ans:

A D-type latch is shown below.

The advantage of this is the single D input.
The flip flop takes the value at its D input whenever the clock pulse input is high it will effectively "track' the input levels as long as the clock input is high.
If the clock input is zero, the state will be that of the last state the flipflop was when it was high.

Ex 13: Design Linear Feed-back Shift Register.

Ans:

A shift register with feedback consists of four flip-flops connected in a shift register configuration and feedback from these four flip-flops to the flip-flop's inputs. This particular counter is started by setting 1 in X1 and 0s in X2, X3 and X4. The sequence of states is then

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

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

Notice that this sequence contains 15 of 16 possible 4 bit numbers that might be taken by that might be taken by this circuit. This is a widely used sequence which occurs in many instruments and has many uses in radar system, sonar system, coding encryption boxes, etc.

X1	X2	X3	X4
1	0	0	0
0	1	0	0
0	0	1	0
1	0	0	1
1	1	0	0
0	1	1	0
1	0	1	1
0	1	0	1
1	0	1	0
1	1	0	1
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1

In the counter table, the flip – flop names are first listed, flowed by the starting states. Then the successive states taken are listed in order, and the final line contains the state preceding the starting state.

Ex 14: Design a logical circuit that will calculate the less than for 2 bits...

Ans:

Step: 1 specification

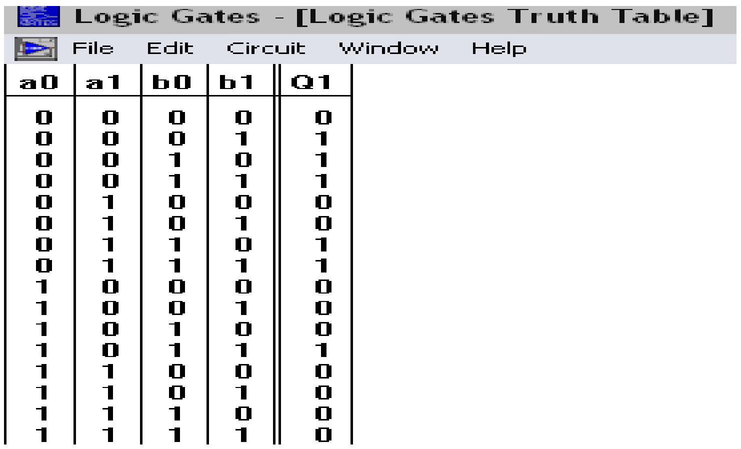
This circuit compares two inputs of size 2- bits i.e. its range is (0-3) the output will be 1 if A<B else0

Input: 2 input bits

1 bit for A0
1 bit for A1
1 bit for B0
1 bit for B1

Output: 1 bit (either 0 or 1)

Step2: Truth table

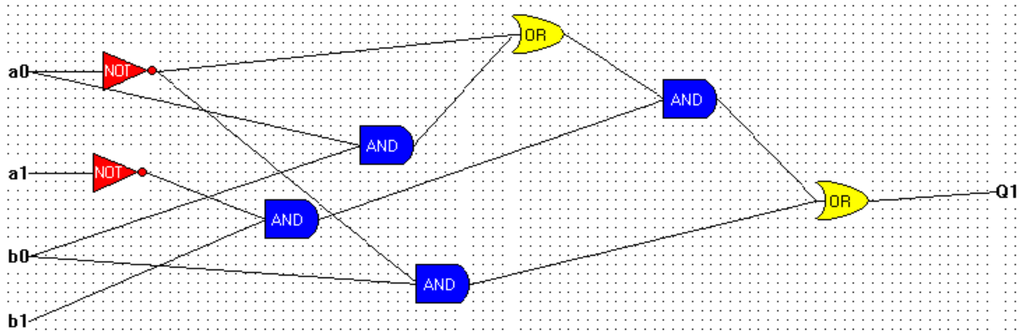

Logic Gates - [Logic Gates Truth Table]
File
Edit
Circuit
Window
Help

a0	a1	b0	b1	Q1
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Step 3: Minterms and K-map

O1=F1 (1, 2, 3, 6, 7, 11)

Circuit:

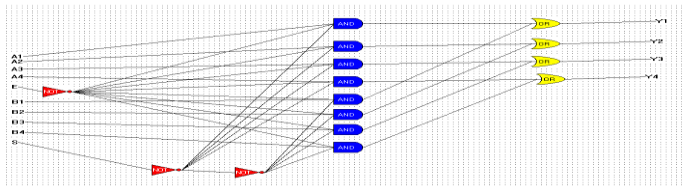


Ex 15: Design a multiplexer circuit that accepts N inputs and Outputs the value of one of those outputs.

Ans:

Multiplexing means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is a combination circuit that selects binary info from one of the many input lines and directs it to a single output line.

Circuit:



Ex 16: Design a decoder that has m inputs and 2^m outputs.

Ans: A decoder has the characteristic that for each possible 2^n input which can be taken by the n input cells, the matrix will have a unique one of its 2^n output lines selected.

SECTION – 3

ASSEMBLY LANGUAGE PROGRAMMING

Session 3 & 4 – Simple Assembly Programs

Ex 1: Write a program to add two numbers present in two consecutive memory locations and store the result in next memory location.

Ans:
```
Prg(add2num.asm)
Title add two numbers in consecutive memory location
dosseg
.model small
.stack
.data
msg1 db 13,10,"Sum of two numbers stored in memory:$"
num1 db 20h
num2 db 15h
sum db ?
res db 20 DUP('$')
.code
main proc
mov ax,@data
mov ds,ax
mov al,num1
add al,num2
mov sum,al
lea dx,msg1
mov ah,09h
int 21h
mov dl,summov ah,02hint 21h
mov ax,4c00h
int 21h
main endp
end
```
Output:
```
Sum of two numbers stored in memory:5
```

Ex 2: Develop program to read a character from console and echo it.

Ans:
```
Prg(rdecho.asm)
Title read a character from console and echo it.
dosseg
.model small
.stack
.data
msg1 db 13,10,"Enter a character:$"
msg2 db 13,10,"Read a character from console and echo:$"
.code
```

```
main proc
mov ax,@data
mov ds,ax
lea dx,msg1
mov ah,09h
int 21h
mov ah,01h
int 21h
mov bl,al
lea dx,msg2
mov ah,09h
int 21h
mov dl,bl
mov ah,02h
int 21h
mov ax,4c00h
int 21h
main endp
end
```

Output:

```
Enter a character:w
Read a character from console and echo:w
```

Ex 3: Develop and execute a program to read 10 chars from console.

Ans: **Prg(rd10chr.asm)**

```
Title read a 10 character from console.
dosseg
.model small
.stack
.data
msg1 db 13,10,"Enter a 10 character:$"
.code
main proc
mov ax,@data
mov ds,ax
lea dx,msg1
mov ah,09h
int 21h
mov cx,00
mov cl,10
rpt: mov ah,01h
int 21h
mov bl,al
loop rpt
mov ax,4c00h
int 21h
main endp
end
```

Output:

```
Enter a 10 character:1234567890
```

Ex 4: Write a program to exchange two memory variables using MOV and XCHG instruction. Can you do it with just XCHG?

Ans: **Prg(XCHGin.asm)**

Title to exchange two memory variables using MOV and XCHG instruction

```
dosseg
     .model small
     .stack
 .data
     msg1 db 13,10,"First value in memory:$"
     msg2 db 13,10,"Second value in memory:$"
     msg3 db 13,10,"After using XCHG instruction:$"
     msg4 db 13,10,"First value in memory:$"
     msg5 db 13,10,"Second value in memory:$"
     value1 db 35h
     value2 db 32h
     .code
     main proc
     mov ax,@data
     mov ds,ax
     lea dx,msg1
     mov ah,09h
     int 21h
     mov dl,value1
     mov ah,02h
     int 21h
     lea dx,msg2
     mov ah,09h
     int 21h
     mov dl,value2
     mov ah,02h
     int 21h
     lea dx,msg3
     mov ah,09h
     int 21h
     ;exchanging the value
     mov al,value1
     XCHG value2,al
     mov value1,al
     lea dx,msg4
     mov ah,09h
     int 21h
     mov dl,value1
     mov ah,02h
     int 21h
     lea dx,msg5
     mov ah,09h
     int 21h
     mov dl,value2
     mov ah,02h
     int 21h
     main endp
     end
```

Output:

```
First value in memory:5
Second value in memory:2
After using XCHG instruction:
First value in memory:2
Second value in memory:5
```

Ex 6: Write a program, which will read two decimal numbers, then multiply them together, and finally print out the result (in decimal).

Ans:
```
data segment
    ms1 db 13,10,"ENTER FIRST NO    :$"
    ms2 db 13,10,"ENTER SECOND NO   :$"
    ms3 db 13,10,"MULTIPLICATION IS :$"
data ends

code segment
         assume cs:code,ds:data
           start:
         mov ax,data
         mov ds,ax
         mov ah,09h
         mov dx,offset ms1
         int 21h
         mov ah,01h
         int 21h
         mov cl,al
         and cl,0fh
         mov ah,09h
         mov dx,offset ms2
         int 21h
         mov ah,01h
         int 21h
         and al,0fh
         mul cl
         aam
         mov bx,ax
         or bx,3030h

         mov ah,09h
         mov dx,offset ms3
         int 21h
         mov dl,bh
         mov ah,02h
         int 21h
         mov dl,bl
         mov ah,02h
         int 21h
         mov ah,4ch
         int 21h
code ends
    end start
output-
multiplication upto 9 * 9 = 81
```

Ex 7: Write a program to convert the ASCII code to its BCD equivalent.

Ans: **Prg(pkdbcd.asm)**

```
      Title convert the ASCII code to bcd equivalent
dosseg
      .model small
      .stack
 .data
      msg1 db 13,10,"Enter the first number:$"
      msg3 db 13,10,"Result of packed bcd:$"
      bcd db ?
      first db ?
      sec db ?
      res db 20 DUP('$')
      .code
 main proc
      mov ax,@data
      mov ds,ax
      lea dx,msg1
      mov ah,09h
      int 21h
      mov ax,00
      mov ah,01h
      int 21h
      sub al,'0'
      mov bl,al
      mov ax,00
      mov ah,01h
      int 21h
      sub al,'0'
      and bl,0Fh
      and al,0Fh
      mov cl,04h
      rol bl,cl
      or al,bl
      mov bcd,al
      lea dx,msg3
      mov ah,09h
      int 21h
      mov dx,00
      mov dl,bcd
      mov ah,02h
      int 21h
      mov ax,4c00h
      int 21h
      main endp
      end

      OUTPUT:
      Enter first number:35
      Result of packed bcd:05
```

Ex 8: Write a program, which will read in two decimal inputs and print out their sum, in decimal.

Ans: **Prg(desum.asm)**

Title read 2 decimal number and print there sum

```
dosseg
      .model small
      .stack
 .data
      msg1 db 13,10,"Enter first number:$"
      msg2 db 13,10,"Enter second number:$"
      msg3 db 13,10,"Sum in decimal number:$"
      num1 db ?
      sum db ?
      res db 20 DUP('$')
      .code
      main proc
      mov ax,@data
      mov ds,ax
      lea dx,msg1
      mov ah,09h
      int 21h
      mov ah,01h
      int 21h
      sub al,'0'
      mov num1,al
      lea dx,msg2
      mov ah,09h
      int 21h
      mov ah,01h
      int 21h
      sub al,'0'
      add al,num1
      mov sum,al
      lea dx,msg3
      mov ah,09h
      int 21h
      mov si,offset res
      mov ax,00
      mov al,sum
      call hex2asc
      lea dx,res
      mov ah,09h
      int 21h
      mov ax,4c00h
      int 21h
      main endp
      hex2asc proc near
      push ax
      push bx
      push cx
      push dx
      push si
      mov cx,00h
      mov bx,0Ah
      rpt1: mov dx,00
      div bx
      add dl,'0'
      push dx
      inc cx
      cmp ax,0Ah
      jge rpt1
      add al,'0'
      mov [si],al
  rpt2: pop ax
      inc si
      mov [si],al
      loop rpt2
      inc si
      mov al,'$'
      mov [si],al
      pop si
      pop dx
      pop cx
      pop bx
      pop ax
      ret
      hex2asc endp
      end
```

```
OUTPUT:
Enter first number:2
Enter second number:3
Sum in decimal number:05
Enter first number:5
Enter second number:6
Sum in decimal number:11
```

Ex 9: Write a program, which will read in two decimal inputs and print out the smaller of the two, in decimal.

Ans: **Prg(desmall.asm)**

```
    Title read in two decimal inputs and print out the smaller of the two, in
    decimal
dosseg
    .model small
    .stack
 .data
    msg1 db 13,10,"Enter the first number:$"
    msg2 db 13,10,"Enter the second number:$"
    msg3 db 13,10,"Smaller of two in decimal:$"
    num1 db ?
    small db ?
    res db 20 DUP('$')
    .code
 main proc
    mov ax,@data
    mov ds,ax
    lea dx,msg1
    mov ah,09h
    int 21h
    mov ah,01h
    int 21h
    sub al,'0'
    mov num1,al
```

```
        lea dx,msg2
        mov ah,09h
        int 21h
        mov ah,01h
        int 21h
        sub al,'0'
        cmp al,num1
        jb sma
        mov bl,num1
        mov small,bl
        jmp prin
        sma :mov small,al
        prin:lea dx,msg3
        mov ah,09h
        int 21h
        mov si,offset res
        mov ax,00
        mov al,small
        call hex2asc
        lea dx,res
        mov ah,09h
        int 21h
        mov ax,4c00h
        int 21h
        main endp
        hex2asc proc near
        push ax
        push bx
        push cx
        push dx
        push si
        mov cx,00h
        mov bx,0Ah
        rpt1: mov dx,00
        div bx
        add dl,'0'
        push dx
        inc cx
        cmp ax,0Ah
        jge rpt1
        add al,'0'
        mov [si],al
    rpt2: pop ax
        inc si
        mov [si],al
        loop rpt2
        inc si
        mov al,'$'
        mov [si],al
        pop si
        pop dx
        pop cx
        pop bx
        pop ax
        ret
```

```
hex2asc endp
end
OUTPUT:
Enter the first number:5
Enter the second number:2
Smaller of two in decimal:02
Enter the first number:8
Enter the second number:9
Smaller of two in decimal:08
```

Ex 10: Write a program to calculate the average of three given numbers stored in memory.

Ans: **Prg(avgthree.asm)**

```
    Title calculate average of three given numbers stored in memory
dosseg
    .model small
    .stack
 .data
    msg1 db 13,10,"Sum of three numbers stored in memory:$"
    msg2 db 13,10,"Average of three numbers stored in memory:$"
    num1 db 10h
    num2 db 10h
    num3 db 10h
    sum db ?
    avg db ?
    res db 20 DUP('$')
    .code
 main proc
    mov ax,@data
    mov ds,ax
    mov al,num1
    add al,num2
    add al,num3
    mov sum,al
    lea dx,msg1
    mov ah,09h
    int 21h
    mov dl,summov ah,02hint 21h
    mov al,sum
    mov ah,00h
    mov bl,03
    div bl
    mov avg,al
    lea dx,msg2
    mov ah,09h
    int 21h
    mov dl,avg
    mov ah,02h
    int 21h
    mov ax,4c00h
    int 21h
    main endp
    end
    OUTPUT:
    Sum of three numbers stored in memory:0
    Average of three numbers stored in memory:►
```

Ex 11: Write a program in 8086 assembly language to find the volume of sphere using following formula:

$$V = 4/3\pi\ r^3$$

Ans: **Prg(volsph.asm)**

```
       Title volume of sphere:
 dosseg
       .model small
       .stack
  .data
       msg1 db 13,10,"Enter the radius:$"
       msg2 db 13,10,"Volume of sphere is:$"
       num db ?
       rad dw ?
       pi dw ?
       result dw ?
       res db 10 DUP('$')
       .code
  main proc
       mov ax,@data
       mov ds,ax
       lea dx,msg1
       mov ah,09h
       int 21h
       call readnum
       mov cx,2
       mov ax,00
       mov al,num
       mov bx,00
       mov bl,num
  rpt: mov dx,00
       mul bl
       loop rpt
       mov rad,ax
       mov ax,00
       mov ax,22
       mov bx,00
       mov bx,7
       cwd
       mov dx,00
       div bx
       mov pi,ax
       mov ax,00
       mov ax,rad
       mov bx,00
       mov bx,4
       mov dx,00
       mul bx
       mov result,ax
       mov ax,00
       mov ax,result
       mov bx,pi
       mov dx,00
```

```
          mul bx
          mov result,ax
          mov bx,00
          mov bx,3
          cwd
          mov ax,00
          mov ax,result
          mov dx,00
          div bx
          mov result,ax
          mov si,offset res
          call hex2asc
          lea dx,msg2
          mov ah,09h
          int 21h
          lea dx,res
          mov ah,09h
          int 21h
          mov ax,4c00h
          int 21h
          main endp
readnum proc near
          mov ah,01h
          int 21h
          sub al,'0'
          mov bh,0Ah
          mul bh
          mov num,al
          mov ah,01h
          int 21h
          sub al,'0'
          add num,al
          ret
          readnum endp
          hex2asc proc near
          push ax
          push bx
          push cx
          push dx
          push si
          mov cx,00h
          mov bx,0Ah
          rpt1: mov dx,00
          div bx
          add dl,'0'
          push dx
          inc cx
          cmp ax,0Ah
          jge rpt1
          add al,'0'
          mov [si],al
          rpt2: pop ax
          inc si
          mov [si],al
          loop rpt2
```

```
inc si
mov al,'$'
mov [si],al
pop si
pop dx
pop cx
pop bx
pop ax
ret
hex2asc endp
end
```

Output:

```
Enter the radius:02
Volume of sphere is:32
Enter the radius:04
Volume of sphere is:256
```

Ex 13: Write a program to convert Centigrade (Celsius) to Fahrenheit temperature measuring scales. Using formula: Celsius = (Fahrenheit - 32) * 5 / 9

Ans: **Prg(farcel.asm)**

```
      Title convert temperature celsius to Farenheit:
dosseg
      .model small
      .stack
 .data
      msg1 db 13,10,"Enter a number to find fahrenheit temperature:$"
      msg2 db 13,10,"Fahrenheit Temperature is:$"
      num db ?
      res db 10 DUP('$')
      .code
 main proc
      mov ax,@data
      mov ds,ax
      lea dx,msg1
      mov ah,09h
      int 21h
      call readnum
      mov bx,00
      mov bx,9
      mov ax,00
      mov al,num
      mov dx,00
      mul bx
      mov bx,5
      cwd
      div bx
      add ax,32
      mov si,offset res
      call hex2asc
      lea dx,msg2
      mov ah,09h
      int 21h
      lea dx,res
      mov ah,09h
```

```
          int 21h
          mov ax,4c00h
          int 21h
          main endp
readnum proc near
          mov ah,01h
          int 21h
          sub al,'0'
          mov bh,0Ah
          mul bh
          mov num,al
          mov ah,01h
          int 21h
          sub al,'0'
          add num,al
          ret
          readnum endp
          hex2asc proc near
          push ax
          push bx
          push cx
          push dx
          push si
          mov cx,00h
          mov bx,0Ah
          rpt1: mov dx,00
          div bx
          add dl,'0'
          push dx
          inc cx
          cmp ax,0Ah
          jge rpt1
          add al,'0'
          mov [si],al
          rpt2: pop ax
          inc si
          mov [si],al
          loop rpt2
          inc si
          mov al,'$'
          mov [si],al
          pop si
          pop dx
          pop cx
          pop bx
          pop ax
          ret
          hex2asc endp
          end
```

Output:

```
Enter a number to find fahrenheit temperature:28
Fahrenheit Temperature is:82
Enter a number to find fahrenheit temperature:40
Fahrenheit Temperature is:104
```

Ex 14: Write a Program which adds the sales tax in the Price list of items and replace the Price list with a new list.

Ans: **Prg(saltax.asm)**

```
        Title adds the sales tax in the price list of items and replace price list with a
        new list:
dosseg
        .model small
        .stack
 .data
        msg1 db 13,10,"How many numbers:$"
        msg2 db 13,10,"Enter number between 1 to 99:$"
        msg3 db 13,10,"Enter Price:$"
        msg4 db 13,10,"Sales tax 2 rupes for less then 100 rupees:$"
        msg5 db 13,10,"After add sales tax price list is:$"
        msg6 db 13,10,"Price number is:$"
        ntable db 100 DUP(0)
        num db ?
        temp db ?
        res db 20 DUP('$')
        .code
        main proc
        mov ax,@data
        mov ds,ax
        lea dx,msg1
        mov ah,09h
        int 21h
        call readnum
        lea dx,msg2
        mov ah,09h
        int 21h
        ;read all numbers
        mov si,offset ntable
        mov ch,00
        mov cl,num
nread:lea dx,msg3
        mov ah,09h
        int 21h
        call readnum1
        mov al,temp
        mov [si],al
        inc si
        loop nread
        mov si,offset ntable
        mov cx,00
        mov cl,num
 sl: mov ax,00
        mov al,[si]
        add al,2
        mov [si],al
        inc si
        loop sl
        lea dx,msg4
        mov ah,09h
        int 21h
```

```
          lea dx,msg5
          mov ah,09h
          int 21h
          mov cx,00
          mov cl,num
          mov si,offset res
          mov di,offset ntable
     rpt: mov ax,00
          mov al,[di]
          call hex2asc
          lea dx,msg6
          mov ah,09h
          int 21h
          lea dx,res
          mov ah,09h
          int 21h
          inc di
          loop rpt
          mov ax,4c00h
          int 21h
          main endp
          readnum proc near
          mov ah,01h
          int 21h
          sub al,'0'
          mov bh,0Ah
          mul bh
          mov num,al
          mov ah,01h
          int 21h
          sub al,'0'
          add num,al
          ret
          readnum endp
readnum1 proc near
          mov ah,01h
          int 21h
          sub al,'0'
          mov bh,10
          mul bh
          mov temp,al
          mov ah,01h
          int 21h
          sub al,'0'
          add temp,al
          ret
          readnum1 endp
          hex2asc proc near
          push ax
          push bx
          push cx
          push dx
          push si
          mov cx,00h
          mov bx,0Ah
          rpt1: mov dx,00
          div bx
          add dl,'0'
          push dx
          inc cx
          cmp ax,0Ah
          jge rpt1
          add al,'0'
          mov [si],al
      rpt2: pop ax
          inc si
          mov [si],al
          loop rpt2
          inc si
          mov al,'$'
          mov [si],al
          pop si
          pop dx
          pop cx
          pop bx
          pop ax
          ret
          hex2asc endp
          end
```

Output:

```
How many numbers:04
Enter number between 1 to 99:
Enter Price:11
Enter Price:22
Enter Price:33
Enter Price:44
Sales tax 2 rupes for less then 100 rupees:
After add sales tax price list is:
Price number is:13
Price number is:24
Price number is:35
Price number is:46
```

Session 5, 6 & 7 – Loop And Comparisons

Ex 1: Write a program to find the factorial of decimal number given by user.

Ans: **Prg(fact.asm)**

```
      Title factorial of a given number
  dosseg
      .model small
      .stack
   .data
      msg1 db 13,10,"Enter a number to find factorial:$"
      msg2 db 13,10,"Factorial of given number is:$"
      num db ?
      res db 10 DUP('$')
      .code
```

```
main proc
    mov ax,@data
    mov ds,ax
    lea dx,msg1
    mov ah,09h
    int 21h
    call readnum
    mov ax,01
    mov ch,00
    mov cl,num
    cmp cx,00
    je skip
rpt: mov dx,00
    mul cx
    loop rpt
    skip:mov si,offset res
    call hex2asc
    lea dx,msg2
    mov ah,09h
    int 21h
    lea dx,res
    mov ah,09h
    int 21h
    mov ax,4c00h
    int 21h
    main endp
    readnum proc near
    mov ah,01h
    int 21h
    sub al,'0'
    mov bh,0Ah
    mul bh
    mov num,al
    mov ah,01h
    int 21h
    sub al,'0'
    add num,al
    ret
    readnum endp
    hex2asc proc near
    push ax
    push bx
    push cx
    push dx
    push si
    mov cx,00h
    mov bx,0Ah
    rpt1: mov dx,00
    div bx
    add dl,'0'
    push dx
    inc cx
    cmp ax,0Ah
    jge rpt1
    add al,'0'
```

```
        mov [si],al
    rpt2: pop ax
        inc si
        mov [si],al
        loop rpt2

        inc si
        mov al,'$'
        mov [si],al
        pop si
        pop dx
        pop cx
        pop bx
        pop ax
        ret
        hex2asc endp
        end
```

Output:

```
Enter a number to find factorial:03
Factorial of given number is:06
Enter a number to find factorial:05
Factorial of given number is:120
```

Ex 4: Write a program, which will read in decimal inputs repeatedly until a zero value is read; at this point, it should print out the sum of the numbers read in so far.

Ans: **Prg(sum0.asm)**

```
        Title read decimal inputs repeatedly until a zero value is read and print sum of the
        numbers
        read in so far:
        dosseg
        .model small
        .stack
    .data
        msg1 db 13,10,"Enter number and get the sum untill 00 is read:$"
        msg2 db 13,10,"Enter number:$"
        msg3 db 13,10,"Sum is:$"
        num db ?
        temp db ?
        res db 10 DUP('$')
        .code
        main proc
        mov ax,@data
        mov ds,ax
        lea dx,msg1
        mov ah,09h
        int 21h
        ;read numbers
        mov ax,00
        mov temp,al
    read: lea dx,msg2
        mov ah,09h
        int 21h
        call readnum
        mov al,num
        cmp al,00
        je ou
        mov ax,00
        mov al,temp
        add al,num
        mov temp,al
        mov ax,00
        mov al,temp
        mov si,offset res
        call hex2asc
        lea dx,msg3
        mov ah,09h
        int 21h
        lea dx,res
        mov ah,09h
        int 21h
        mov ax,00
        mov al,temp
        jmp read
        ou: mov ax,4c00h
        int 21h
        main endp
        readnum proc near
        mov ah,01h
        int 21h
        sub al,'0'
        mov bh,0Ah
        mul bh
        mov num,al
        mov ah,01h
        int 21h
        sub al,'0'
        add num,al
        ret
        readnum endp
        hex2asc proc near
        push ax
        push bx
        push cx
        push dx
        push si
        mov cx,00h
        mov bx,0Ah
        rpt1: mov dx,00
        div bx
        add dl,'0'
        push dx
        inc cx
        cmp ax,0Ah
        jge rpt1
        add al,'0'
        mov [si],al
rpt2: pop ax
        inc si
        mov [si],al
        loop rpt2
```

```
inc si
mov al,'$'
mov [si],al
pop si
pop dx
pop cx
pop bx
pop ax
ret
hex2asc endp
end
```

Output:

```
Enter number and get the sum untill 00 is read:
Enter number:11
Sum is:11
Enter number:22
Sum is:33
Enter number:33
Sum is:66
Enter number:44
Sum is:110
Enter number:00
```

Ex 5: Develop and execute an assembly language program to find the LCM of two 16-bit unsigned integers.

Ans: **Prg(lcm16.asm)**

Title program to find lcm of two 16 bit unsigned integers.

```
dosseg
    .model small
    .stack
 .data
    cr equ 0dh
    lf equ 0ah
    msg db cr,lf,"Program for LCM of two positive Integers..:$"
    msg1 db cr,lf,"Enter numbe1:$"
    msg2 db cr,lf,"Enter number2:$"
    msg3 db cr,lf,"LCM=:$"
    num1 dw ?
    num2 dw ?
    gcd dw ?
    num3 dw ?
    lcm dw ?
    res db 10 DUP(0)
 buff db 80
    db 0
    db 80 DUP(?)
 .code
 main proc
 mov ax,@data
 mov ds,ax
 mov ah,09h
 mov dx,offset msg
 int 21h
 ;Read number1
 mov ah,09h
```

```
            mov dx,offset msg1
            int 21h
            call readinteger
            ;Read number2
            mov ah,09h
            mov dx,offset msg2
            int 21h
            call readinteger1
            ;push num1 and num2 into stack
            mov ax,num1
            push ax
            mov ax,num2
            push ax
            call findgcd
            add sp,4
            ;adjust stack pointer
            mov gcd,ax
            ;gcd = findgcd(num[i],num[i+1])
            ;LCM = (num1*num2)/gcd(num1,num2)
            mov ax,num1
            mov dx,00
            mul num2
            div gcd
            mov lcm,ax
            ;print LCM
            mov ah,09h
            mov dx,offset msg3
            int 21h
            mov ax,lcm
            mov si,offset res
            call hex2asc
            mov ah,09h
            mov dx,offset res
            int 21h
            mov ax,4c00h
            int 21h
            main endp
readinteger proc near
            push dx
            push bx
            push ax
            mov ah,0ah
            mov dx,offset buff
            int 21h
            mov bx,offset buff
            add bx,2
            push bx
            call atoi
            pop bx
            mov num1,ax
            pop ax
            pop bx
            pop dx
            ret
            readinteger endp
```

```
        readinteger1 proc near
        push dxpush bxpush ax
        mov ah,0ah
        mov dx,offset buff
        int 21h
        mov bx,offset buff
        add bx,2
        push bx
        call atoi
        pop bx
        mov num2,ax
        pop ax
        pop bx
        pop dx
        ret
        readinteger1 endp
        findgcd proc near
        push bp
        mov bp,sp
        push dx
        push bx
        rpt: mov ax,[bp+4]
        mov bx,[bp+6]
        cmp ax,bx
        jl skip
        mov [bp+6],ax
        mov [bp+6],bx
 skip: mov dx,00
        mov ax,[bp+6]
        div word ptr[bp+4]
                                        ;num2/num1
        mov [bp+6],dx
        cmp dx,00
        jne rpt
        mov ax,[bp+4]
        pop bx
        pop dx
        pop bp
        ret
        findgcd endp
        atoi proc near
        push bp
        mov bp,sp
        push si
        push dx
        push cx
        push bx
        mov si,[bp+4]
        ;finding the length of the string
        mov bx,00
nxtch: mov al,[si]
        inc bx
        inc si
        cmp al,cr
        jne nxtch
        ;cx=length of the string
        mov cx,bx
        dec cx
        ;si is pointing outside the string so adjust
        dec si
        mov dx,00
        mov bx,01
        nxt: dec si
        push dx
        ;dx:ax=digit
        xor dx,dx
        mov ah,00
        mov al,[si]
        sub al,'0'
        mul bx
        pop dx
        add dx,ax
        ;generate multiples bx=10,100,1000....
        push dx
        push cx
        xor dx,dx
        mov cx,10
        mov ax,bx
        mul cx
        mov bx,ax
        pop cx
        pop dx
        loop nxt
        mov ax,dx
        pop bx
        pop cx
        pop dx
        pop si
        pop bp
        ret
        atoi endp
        hex2asc proc near
        push ax
        push bx
        push cx
        push dx
        push si
        mov cx,00h
        mov bx,0Ah
        rpt1: mov dx,00
        div bx
        add dl,'0'
        push dx
        inc cx
        cmp ax,0Ah
        jge rpt1
        add al,'0'
        mov [si],al
rpt2: pop ax
        inc si
```

```
mov [si],al
loop rpt2
inc si
mov al,'$'
mov [si],al
pop si
pop dx
pop cx
pop bx
pop ax
ret
hex2asc endp
end
```

Output:

```
Program for LCM of two positive Integers..:
Enter numbe1:150
Enter number2:75
LCM=:150
```

Ex 7: Develop and execute a program to sort a given set of 8-bit unsigned integers into ascending order.

Ans: **Prg(ascor.asm)**

```
    Title sort(bubble sort) an given array element in ascending order
dosseg
    .model small
    .stack
 .data
    msg1 db 13,10,"How many numbers:$"
    msg2 db 13,10,"Enter number:$"
    msg3 db 13,10,"Sorted elements in ascending order are:$"
    msg4 db 13,10,"Element:$"
    ntable db 100 DUP(0)
    num db ?
    temp db ?
    count db ?
    res db 10 DUP('$')
    .code
    main proc
    mov ax,@data
    mov ds,ax
    lea dx,msg1
    mov ah,09h
    int 21h
    call readnum
    ;read all numbers
    mov si,offset ntable
    mov ch,00
    mov cl,num
nread:lea dx,msg2
    mov ah,09h
    int 21h
    call readnum1
    mov al,temp
    mov [si],al
```

```
        inc si
        loop nread
        ;sorting an array elements
        mov cx,00
        mov cl,num
        cmp cx,01
        ;if(num=01)then print array elements
        je pnxt1
        nxtps:mov dx,00
        ;flag =false
        mov bx,00
        ;j=1
   nxtj: mov al,ntable[bx]
        mov ah,ntable[bx+1]
        cmp ah,0
        je skip
        cmp al,ah
        jle skip
        mov ntable[bx],ah
        mov ntable[bx+1],al
        mov dl,01
   skip: inc bx
        cmp bx,cx
        jl nxtj
        dec cx
        jz pnxt1
        cmp dl,01h
        je nxtps
        ;print array elements
   pnxt1:mov ch,00
        mov cl,num
        mov di,offset ntable
        mov si,offset res
        lea dx,msg3
        mov ah,09h
        int 21h
   pnxt: lea dx,msg4
        mov ah,09h
        int 21h
        mov ah,00
        mov al,[di]
        call hex2asc
        lea dx,res
        mov ah,09h
        int 21h
        inc di
        loop pnxt
        mov ax,4c00h
        int 21h
        main endp
readnum proc near
        mov ah,01h
        int 21h
        sub al,'0'
        mov bh,0Ah
        mul bh
        mov num,al
        mov ah,01h
        int 21h
        sub al,'0'
        add num,al
        ret
        readnum endp
readnum1 proc near
        mov ah,01h
        int 21h
        sub al,'0'
        mov bh,0Ah
        mul bh
        mov temp,al
        mov ah,01h
        int 21h
        sub al,'0'
        add temp,al
        ret
        readnum1 endp
        hex2asc proc near
        push ax
        push bx
        push cx
        push dx
        push si
        mov cx,00h
        mov bx,0Ah
        rpt1: mov dx,00
        div bx
        add dl,'0'
        push dx
        inc cx
        cmp ax,0Ah
        jge rpt1
        add al,'0'
        mov [si],al
  rpt2: pop ax
        inc si
        mov [si],al
        loop rpt2
        inc si
        mov al,'$'
        mov [si],al
        pop si
        pop dx
        pop cx
        pop bx
        pop ax
        ret
        hex2asc endp
        end
```

Output:

```
How many numbers:04
Enter number:04
Enter number:03
Enter number:02
Enter number:01
Sorted elements in ascending order are:
Element:01
Element:02
Element:03
Element:04
```

Ex 11: Write a program to Convert ASCII number into decimal digit.

Ans: Prg(ascdec.asm)

```
        Title convert ASCII to decimal digit
        dosseg
        .model small
        .stack
    .data
        msg1 db 13,10,"Enter a number:$"
        msg2 db 13,10,"Decimal number is:$"
        num db ?
        res db 10 DUP('$')
        .code
    main proc
        mov ax,@data
        mov ds,ax
        lea dx,msg1
        mov ah,09h
        int 21h
        call readnum
    skip:mov si,offset res
        mov ax,00
        mov al,num
        call hex2asc
        lea dx,msg2
        mov ah,09h
        int 21h
        lea dx,res
        mov ah,09h
        int 21h
        mov ax,4c00h
        int 21h
        main endp
  readnum proc near
        mov ah,01h
        int 21h
        sub al,'0'
        mov bh,0Ah
        mul bh
        mov num,al
        mov ah,01h
        int 21h
        sub al,'0'
        add num,al
        ret
        readnum endp
        hex2asc proc near
        push ax
        push bx
        push cx
        push dx
        push si
        mov cx,00h
        mov bx,0Ah
        rpt1: mov dx,00
        div bx
        add dl,'0'
        push dx
        inc cx
        cmp ax,0Ah
        jge rpt1
        add al,'0'
        mov [si],al
    rpt2: pop ax
        inc si
        mov [si],al
        loop rpt2
        inc si
        mov al,'$'
        mov [si],al
        pop si
        pop dx
        pop cx
        pop bx
        pop ax
        ret
        hex2asc endp
        end
```

Output:

```
Enter a number:12
Decimal number is:12
```

Ex 16: Write a Program, which should adds two 5-byte numbers (numbers are stored in array-NUM1 & NUM2), and stores the sum in another array named RESULT.

Ans: **Prg(ad5bnm.asm)**

```
    Title add 5 byte numbers(num1 and num2 array) and stores the sum array
    named RESULT
dosseg
    .model small
    .stack
 .data
    len equ 05h
    msg db 13,10,"To calculate sum of 5 byte number stored in memory.....$"
    msg1 db 13,10,"Element in first array................................$"
    msg2 db 13,10,"Element is:$"
    msg3 db 13,10,"Element in second array...............................$"
    msg4 db 13,10,"Sum is:$"
    num1 db 31h, 32h, 33h, 34h, 35h
    num2 db 31h, 32h, 33h, 34h, 35h
```

```
      sum db 6 DUP(0)
      res db 10 DUP(0)
      .code
      main proc
      mov ax,@data
      mov ds,ax
      lea dx,msg
      mov ah,09h
      int 21h
      ;print first array element
      lea dx,msg1
      mov ah,09h
      int 21h
      mov cx,00
      mov cl,05
      mov di,00
 nxt: lea dx,msg2
      mov ah,09h
      int 21h
      mov dl,num1[di]
      mov ah,02h
      int 21h
      inc di
      loop nxt
      ;print second array element
      lea dx,msg3
      mov ah,09h
      int 21h
      mov cx,00
      mov cl,05
      mov si,00
 nxt1:lea dx,msg2
      mov ah,09h
      int 21h
      mov dl,num2[si]
      mov ah,02h
      int 21h
      inc si
      loop nxt1
      ;adding 2 array element
      mov si,00
      mov cx,00
      mov cl,05
      clc
again:mov al,num1[si]
      adc al,num2[si]
      mov sum[si],al
      inc si
      loop again
      rcl al,01h
      and al,01h
      mov sum[si],al
      ;printing array sum
      mov cx,00
      mov cl,06
      mov si,00
      lea dx,msg4
      mov ah,09h
      int 21h
 pnxt:mov dl,sum[si]
      mov ah,02h
      int 21h
      inc si
      loop pnxt
      mov ax,4c00h
      int 21h
      main endp
      end
```

Output:

```
To calculate sum of 5 byte number stored in memory.....
Element in first array..............................
Element is:1
Element is:2
Element is:3
Element is:4
Element is:5
Element in second array..............................
Element is:1
Element is:2
Element is:3
Element is:4
Element is:5
Sum is:bdfhj
```

Ex 17: Write a program which should convert 4 digits BCD number into its binary equivalent.

Ans: **Prg(bcdbin.asm)**

```
        Title convert 4 digit bcd number into its binary equivalent
dosseg
    .model small
    .stack
    .datathou equ 3E8h
                                  ;1000 =3E8h
    msg db 13,10,"To convert bcd number of 4 digit:$"
    msg1 db 13,10,"Stored in memory to binary equivalent:$"
    msg2 db 13,10,"Hex number for 10 is 0Ah:$"
    msg3 db 13,10,"Hex number for 100 is 64h:$"
    msg4 db 13,10,"Hex number for 1000 is 3E8h:$"
    msg5 db 13,10,"The number stored in memory is 4567h:$"
    msg6 db 13,10,"Its Hex number is 11D7h:$"
    msg7 db 13,10,"After converting bcd number to binary number:$"
    msg8 db 13,10,"Binary number is:$"
    bcd dw 4567h
    hex dw ?
    res db 40 DUP('$')
    .code
    main proc
    mov ax,@data
    mov ds,ax
```

```
lea dx,msg
mov ah,09h
int 21h
lea dx,msg1
mov ah,09h
int 21h
lea dx,msg2
mov ah,09h
int 21h
lea dx,msg3
mov ah,09h
int 21h
lea dx,msg4
mov ah,09h
int 21h
lea dx,msg5
mov ah,09h
int 21h
lea dx,msg6
mov ah,09h
int 21h
;converting bcd to binary
mov ax,bcd
mov bx,ax
mov al,ah
mov bh,bl
mov cl,04
ror ah,cl
ror bh,cl
and ax,0F0Fh
and bx,0F0Fh
mov cx,ax
;multiplying the number by 10,100,1000 to set to there place value
mov ax,0000h
mov al,ch
mov di,thou
mul di
mov dh,00h
mov dl,bl
add dx,ax
mov ax,0064h
mul cl
add dx,ax
mov ax,000Ah
mul bh
add dx,ax
mov hex,dx
;printing the binary number
;its hex value is stored in memory
lea dx,msg7
mov ah,09h
int 21h
lea dx,msg8
mov ah,09h
int 21h
```

```
    mov ax,00
    mov si,offset res
    mov ax,hex
    call hex2asc
    mov dx,offset res
    mov ah,09h
    int 21h
    mov ax,4c00h
    int 21h
    main endp
    hex2asc proc near
    push ax
    push bx
    push cx
    push dx
    push si
    mov cx,00h
    mov bx,0Ah
    rpt1: mov dx,00
    div bx
    add dl,'0'
    push dx
    inc cx
    cmp ax,0Ah
    jge rpt1
    add al,'0'
    mov [si],al
rpt2: pop ax
    inc si
    mov [si],al
    loop rpt2
    inc si
    mov al,'$'
    mov [si],al
    pop si
    pop dx
    pop cx
    pop bx
    pop ax
    ret
    hex2asc endp
    end
```

Output:

```
To convert bcd number of 4 digit:
Stored in memory to binary equivalent:
Hex number for 10 is 0Ah:
Hex number for 100 is 64h:
Hex number for 1000 is 3E8h:
The number stored in memory is 4567h:
Its Hex number is 11D7h:
After converting bcd number to binary number:
Binary number is:4567
```

Session 8 - Strings

Ex 1: Write a program, which takes two inputs as strings and display the Concatenated string.

Ans:
```
     Prg(strcon.asm)
               Title string concat
     dosseg
     .model small
     .stack
     .data
     msg1 db 13,10,"Enter a string with dolar symbol as a break:$"
     msg2 db 13,10,"Enter second string with dolar symbol as a break:$"
     msg3 db 13,10,"Concated string is:$"
     strg db 20 DUP(0)
     .code
 main proc
     mov ax,@data
     mov ds,ax
     lea di,strg
     lea dx,msg1
     mov ah,09h
     int 21h
 first:mov ah,01h
     int 21h
     cmp al,24h
     je next
    ; inc di
     mov [di],al
     inc di
     jmp first
 next: lea dx,msg2
     mov ah,09h
     int 21h
second:mov ah,01h
     int 21h
     cmp al,24h
     je con
     ; inc di
     mov [di],al
     inc di
     jmp second


 con : lea dx,msg3
     mov ah,09h
```

```
int 21h
lea di,strg
dis: mov al,[di]
cmp al,0
je ou
mov dl,al
mov ah,02h
int 21h
inc di
jmp dis
ou: mov ax,4c00h
int 21h
main endp
end
```

Output:

```
Enter a string with dolar symbol as a break:saint$
Enter second string with dolar symbol as a break:alosius$
Concated string is:saintalosius
```

Ex 2: Write a program, which converts string lower case characters to upper case characters and upper case characters to lower case characters.

Ans: **Prg(strul.asm)**

```
    Title convert string upper case to lower case and lower case to upper case
    dosseg
    .model small
    .stack
    .data
    msg1 db 13,10,"Enter a string with dolar symbol as a break:$"
    msg2 db 13,10,"Modified string is:$"
    buf db 80 DUP(0)
    revbuf db 80 DUP(0)
    strlen db ?
    .code
 main proc
    mov ax,@data
    mov ds,ax
    lea dx,msg1
    mov ah,09h
    int 21h
    lea si,buf
read: mov ah,01h
    int 21h
    mov [si],al
    inc si
    cmp al,24h
    je check
    jmp read
    check:lea si,buf
    lea di,revbuf
start:mov al,[si]
    cmp al,'$'
    je dis
    cmp al,60h
    jb lower
    cmp al,7Ah
```

```
        jb upper
        jmp start
   lower:cmp al,40h
        jb skip
        cmp al,5Ah
        jb up
      up:add al,20h
        mov [di],al
        inc di
        inc si
        jmp start
        upper:cmp al,60h
        ja lo
        lo: sub al,20h
        mov [di],al
        inc di
        inc si
        jmp start
   skip: mov [di],al
        inc si
        inc di
        jmp start
        dis:mov al,'$'
        mov [di],al
       lea dx,msg2
        mov ah,09h
        int 21h
        lea dx,revbuf
        mov ah,09h
        int 21h
        ou:mov ax,4c00h
        int 21h
        main endp
        end
```

Output:

```
Enter a string with dolar symbol as a break:SaiNt$
Modified string is:sAInT
```

Ex 3: Write a program for reversing a given string.

Ans:

```
        Prg(strrev.asm)
        Title reversing a string
        dosseg
        .model small
        .stack
    .data
        msg1 db 13,10,"Enter a string with dolar symbol as a break:$"
        msg2 db 13,10,"Reverse of a string is:$"
        strg db 20 DUP(0)
        restr db 20 DUP(0)
        .code
    main proc
        mov ax,@data
        mov ds,ax
        mov es,ax
```

```
      mov di,00
      lea dx,msg1
      mov ah,09h
      int 21h
 read:mov ah,01h
      int 21h
      cmp al,24h
      je next
      inc di
      mov strg[di],al
      jmp read
      next: mov si,00
start:cmp di,0
      je dmsg2
      mov al,strg[di]
      mov restr[si],al
      inc si
      dec di
      jmp start
dmsg2:lea dx,msg2
      mov ah,09h
      int 21h
  dis:mov al,restr[di]
      cmp al,0
      je ou
      mov dl,al
      mov ah,02h
      int 21h
      inc di
      jmp dis
      ou: mov ax,4c00h
      int 21h
      main endp
      end
```

Output:

```
Enter a string with dolar symbol as a break:saint$
Reverse of a string is:tnias
```

Ex 6: Write a program to determine a given string is a palindrome. If 'Yes' output the message “The given string is a palindrome”. If 'No' output the message “No, it is not a palindrome”.

Ans: **Prg(strpal.asm)**

```
      Title string palindrome
      dosseg
      .model small
      .stack
 .data
      msg1 db 13,10,"Enter a string with dolar symbol as a break:$"
      msg2 db 13,10,"Reverse of a given string is:$"
      msg3 db 13,10,"String length is:$"
      msg4 db 13,10,"Is Palindrome:$"
      msg5 db 13,10,"Not a Palindrome:$"
      buf db 80 DUP(0)
      revbuf db 80 DUP(0)
      strlen db ?
      .code
 main proc
      mov ax,@data
      mov ds,ax
      lea dx,msg1
      mov ah,09h
      int 21h
      lea si,buf
read: mov ah,01h
      int 21h
      mov [si],al
      inc si
      cmp al,24h
      je cou
      jmp read
      cou: lea si,buf
      mov bx,00
count:mov al,[si]
      inc si
      ;inc bl
      cmp al,24h
      je rev
      inc bx
      jmp count
      rev: lea di,revbuf
      lea si,buf
      add si,bx
      mov cx,00
      mov cx,bx
      dec si
revst:mov al,[si]
      mov [di],al
      dec si
      inc di
      loop revst
      lea di,revbuf
      lea si,buf
      add di,bx
      add si,bx
      mov al,[si]
      mov [di],al
  dis:lea dx,msg2
      mov ah,09h
      int 21h
      lea dx,revbuf
      mov ah,09h
      int 21h
      lea si,buf
      lea di,revbuf
      mov cx,bx
check:mov al,[si]
      cmp [di],al
      jne pal
      inc di
      inc si
      loop check
```

```
lea dx,msg4
mov ah,09h
int 21h
jmp ou
pal:lea dx,msg5
mov ah,09h
int 21h
ou:mov ax,4c00h
int 21h
main endp
end
```

Output:

```
Enter a string with dolar symbol as a break:srrs$
Reverse of a given string is:srrs
Is Palindrome:
```

Ex 7: Write a program to search for a character in a given string and calculate the number of occurrences of the character in the given string.

Ans: **Prg(strchr.asm)**

```
    Title count character occourence in a string
    dosseg
    .model small
    .stack
 .data
    msg1 db 13,10,"Enter a string with dolar symbol as a break:$"
    msg2 db 13,10,"Enter a character to count:$"
    msg3 db 13,10,"Number of times occoured in a given string:$"
    buf db 80 DUP(0)
    chr db 10 DUP('$')
    strlen db ?
    res db 10 DUP('$')
    .code
 main proc
    mov ax,@data
    mov ds,ax
    lea dx,msg1
    mov ah,09h
    int 21h
    mov si,offset buf
read: mov ah,01h
    int 21h
    mov [si],al
    inc si
    cmp al,24h
    je next
    jmp read
next: lea dx,msg2
    mov ah,09h
    int 21h
read1:mov si,offset chr
    mov ah,01h
    int 21h
    mov [si],al
    inc si
    mov al,24h
    mov [si],al
    mov bx,00
    mov si,offset buf
    mov ax,00
    mov di,offset chr
check:mov al,[si]
    cmp al,[di]
    je count
    cmp al,'$'
    je dis
    inc si
    jmp check
count:inc bl
    inc si
    jmp check
    dis:mov strlen,bl
    lea si,res
    mov ax,00
    mov al,strlen
    call hex2asc
    lea dx,msg3
    mov ah,09h
    int 21h
    lea dx,res
    mov ah,09h
    int 21h
    ou:mov ax,4c00h
    int 21h
    main endp
    hex2asc proc near
    push ax
    push bx
    push cx
    push dx
    push si
    mov cx,00h
    mov bx,0Ah
    rpt1: mov dx,00
    div bx
    add dl,'0'
    pop si
    pop dx
    pop cx
    pop bx
    pop ax
    ret
    hex2asc endp
    end
```

Output:

```
Enter a string with dolar symbol as a break:saintalosius$
Enter a character to count:a
Number of times occoured in a given string:02
```