**FEATURES OF C++**

**Variables**

|  |
| --- |
| a = 5;  b = 2;  a = a + 1;  result = a - b; |

We can define a variable as a portion of memory to store a determined value.

Each variable needs an identifier that distinguishes it from the others, for example, in the previous code the variable identifiers were a, b and result, but we could have called the variables any names we wanted to invent, as long as they were valid identifiers.

**Fundamental data types**

Table of Fundamental Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Size\*** | **Range\*** |
| Char | Character or small integer. | 1 byte | signed: -128 to 127 unsigned: 0 to 255 |
| Short int (short) | Short Integer. | 2 bytes | signed: -32768 to 32767 unsigned: 0 to 65535 |
| Int | Integer. | 4 bytes | signed: -2147483648 to 2147483647 unsigned: 0 - 4294967295 |
| Long int (long) | Long integer. | 4 bytes | signed: -2147483648 to 2147483647 unsigned: 0 - 4294967295 |
| Float | Floating point number. | 4 bytes | 3.4e +/- 38 (7 digits) |
| Double | Double precision floating point number. | 8 bytes | 1.7e +/- 308 (15 digits) |
| Long double | Long double precision floating point number. | 8 bytes | 1.7e +/- 308 (15 digits) |

**Identifiers**

A valid identifier is a sequence of one or more letters, digits or underscore characters (\_).Only letters, digits and single underscore characters are valid. They can also begin with an underline character (\_ ), but in some cases these may be reserved for compiler specific keywords or external identifiers, as well as identifiers containing two successive underscore characters anywhere. In no case they can begin with a digit.

Another rule that you have to consider when inventing your own identifiers is that they cannot match any keyword of the C++ language nor your compiler's specific ones, which are reserved keywords.

The standard reserved keywords are: asm, auto, bool, break, case, catch, char, class, const, const\_cast, continue, default, delete, do, double, dynamic\_cast, else, enum, explicit, export, extern, false, float, for, friend, goto, if, inline, int, long, mutable, namespace, new, operator, private, protected, public, register, reinterpret\_cast, return, short, signed, sizeof, static, static\_cast, struct, switch, template, this, throw, true, try, typedef, typeid, typename, union, unsigned, using, virtual, void, volatile, wchar\_t, while

**Control Structure**

A program is usually not limited to a linear sequence of instructions. During its process it may bifurcate, repeat code or take decisions. For that purpose, C++ provides control structures that serve to specify what has to be done by our program, when and under which circumstances.

Conditional structure: if and else

The if keyword is used to execute a statement or block only if a condition is fulfilled.

Its form is:

if (condition)

statement;

If it is false, statement is ignored (not executed) and the program continues right after this conditional structure.

We can additionally specify what we want to happen if the condition is not fulfilled by using the keyword else. Its form used in conjunction with if is:

if (condition)

statement1;

else

statement2;

The if + else structures can be concatenated with the intention of verifying a range of values.

**Iteration structures (loops)**

Loops have as purpose to repeat a statement a certain number of times or while a condition is fulfilled.

The while loop - Its format is:

while (expression)

statement

and its functionality is simply to repeat statement while the condition set in expression is true.

The do-while loop- Its format is:

do

statement

while (condition);

Its functionality is exactly the same as the while loop, except that condition in the do-while loop is evaluated after the execution of statement instead of before, granting at least one execution of statement even if condition is never fulfilled.

The for loop

Its format is:

for (initialization; condition; increment/decrement)

statement;

and its main function is to repeat statement while condition remains true, like the while loop. But in addition, the for loop provides specific locations to contain an initialization statement and an increase statement. So this loop is specially designed to perform a repetitive action with a counter which is initialized and increased on each iteration**.**

The break statement

Using break we can leave a loop even if the condition for its end is not fulfilled. It can be used to end an infinite loop, or to force it to end before its natural end.

The continue statement

The continue statement causes the program to skip the rest of the loop in the current iteration as if the end of the statement block had been reached, causing it to jump to the start of the following iteration.

The exit function

exit is a function defined in the cstdlib library.

The purpose of exit is to terminate the current program with a specific exit code. Its prototype is:

|  |
| --- |
| void exit (int exitcode); |

The exitcode is used by some operating systems and may be used by calling programs. By convention, an exit code of 0 means that the program finished normally and any other value means that some error or unexpected results happened.

#### The selective structure: switch

The syntax of the switch statement is a bit peculiar. Its objective is to check several possible constant values for an expression. Something similar to what we did at the beginning of this section with the concatenation of several if and else if instructions. Its form is the following:

switch (expression)

{

case constant1: group of statements 1;

break;

case constant2: group of statements 2;

break;

.

.

.

default: default group of statements

}

It works in the following way: switch evaluates expression and checks if it is equivalent to constant1, if it is, it executes group of statements 1 until it finds the break statement. When it finds this break statement the program jumps to the end of the switch selective structure.

If it is equal to this, it will execute group of statements 2 until a break keyword is found, and then will jump to the end of the switch selective structure.

Finally, if the value of expression did not match any of the previously specified constants (you can include as many case labels as values you want to check), the program will execute the statements included after the default: label, if it exists (since it is optional).

**Functions**

Using functions we can structure our programs in a more modular way, accessing all the potential that structured programming can offer to us in C++. A function is a group of statements that is executed when it is called from some point of the program. The following is its format:

type name ( parameter1, parameter2, ...)

{

statements

}

where:

type is the data type specifier of the data returned by the function.

name is the identifier by which it will be possible to call the function.

parameters (as many as needed): Each parameter consists of a data type specifier followed by an identifier, like any regular variable declaration (for example: int x) and which acts within the function as a regular local variable. They allow to pass arguments to the function when it is called. The different parameters are separated by commas.

statements is the function's body. It is a block of statements surrounded by braces { }.

**Arrays**

An array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier.

That means that, for example, we can store 5 values of type int in an array without having to declare 5 different variables, each one with a different identifier. Instead of that, using an array we can store 5 different values of the same type, int for example, with a unique identifier.

A typical declaration for an array in C++ is:

type name [elements];

**Pointers**

A pointer is a variable that holds a memory address,usually the location of another variable in memory.

The declaration of pointers follows this format:

type \* name;

where type is the data type of the value that the pointer is intended to point to.

**Classes & Objects**

A class is a way to blind the data describing an entity and its associated functions together.An object represents an identifiable entity with some characteristics and behaviour.A single class named ‘Directory’ has been used in the code. It consist of private data members and public functions.A single global object has been used.

**Advantages of C++**

C++ provides following advantages over C:

* It is fast, stronger and you are free to create your world.
* As it is described C++'s main advantage is its speed.
* You can create your own formulations and use your own mathematical technics to penetrate these complexities.
* Using Classes and Templates makes coding easier.

**PROJECT OVERVIEW**

**HEADER FILES USED**

1. iostream.h
2. conio.h
3. math.h
4. process.h

**FUNCTIONS: USER DEFINED**

* + - VOID MAIN()

Displays the Welcome Screen of MATHLAB.

* + - VOID WEL()

Displays the Main Menu of MATHLAB.

* + - VOID TRIGO()

Displays the Menu listing the important functions and identities used in Trigonometry.

* + - VOID ALGO()

Displays the Menu listing the important operators and concepts used in Algebra.

* + - VOID COGEO()

Displays the Menu listing the important concepts of Co-ordinate Geometry.

* + - VOID PERMS(INT,INT)

Function to perform the calculations regarding Permutation.

* + - VOID COMBO(INT,INT)

Function to perform the calculations regarding Combination.

* + - VOID DIVISION()

Function to perform the calculations regarding Division.

* + - VOID REMAIN()

Function to perform the calculations regarding Remainder.

* + - VOID SS()

Function to perform calculations under further sub categories of Algebraic Progression and Geometric Progression.

* + - VOID AREA()

Function to find out the area of different geometrical figures.

* + - VOID POINT()

Function to perform calculations based on the concept of point.

* + - VOID LINE()

Function to perform calculations based on the concept of line.

* + - VOID CIRCLE()

Function to perform calculations based on the concept of circle.

* + - VOID PARABOLA()

Function to perform calculations based on the concept of parabola.

* + - VOID ELLIPSE()

Function to perform calculations based on the concept of ellipse.

* + - VOID HYPERBOLA()

Function to perform calculations based on the concept of hyperbola.

**FUNCTIONS: BUILT-IN**

* + - CLRSCR() [CONIO.H] :

Clears the current text window and places the cursor in the upper left-hand corner at position (1,1).

* + - GETCH() [CONIO.H] :

Reads a single character directly from the keyboard.

* + - EXIT() [PROCESS.H] :

Terminates the program.

* + - BREAK() [IOSTREAM.H] :

Causes the enclosing [for](http://cppreference.com/wiki/language/for), [while](http://cppreference.com/wiki/language/while), [do-while](http://cppreference.com/wiki/language/do) loop or [switch statement](http://cppreference.com/wiki/language/switch) to terminate. Used when it is otherwise awkward to terminate the loop using the condition expression and conditional statements.

* COS () [MATH.H] :

It returns the cosine of the argument. The value argument must be in radians.

* SIN () [MATH.H] :

It returns the sine of the argument. The value argument must be in radians.

* TAN () [MATH.H] :

It returns the tangent of the argument. The value argument must be in radians.

* POW () [MATH.H] :

It returns base raised to exponent power i.e. base^exponent.

A domain error occurs if base = 0, exp <= 0, base <0 or exponent is not integer.

**STATEMENTS USED:**

* SWITCH-CASE: Used to select from different options in all the menu based functions.
* IF-ELSE: It tests an expression and depending upon the defined condition executes one of the two sets-of-action.

**PROJECT CODE**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HEADER FILES USED

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream.h>

#include<conio.h>

#include<math.h>

#include<process.h>

double pi = 3.14159; //GLOBAL VARIABLE

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION DECLARATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void trigo();

void algo();

void cogeo();

void perms(int,int);

void combo(int,int);

void division();

void remain();

void ss();

void area();

void point();

void line();

void circle();

void parabola();

void ellipse();

void hyperbola();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION FOR MAIN MENU

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void wel()

{

clrscr();

cout<<"\n\n\n";

cout<<"\t\t WELCOME TO MATH LAB.......!! \n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n";

cout<<"\t\tMAIN MENU \n\n";

cout<<"\t\t<1>TRIGONOMETRY \n\n";

cout<<"\t\t<2>ALGEBRA \n\n";

cout<<"\t\t<3>CO-ORDINATE GEOMETRY \n\n";

cout<<"\t\t<4>EXIT \n\n";

cout<<"\t\tENTER YOUR CHOICE : ";

int x;

cin>>x;

switch(x)

{

case 1: trigo();

break;

case 2: algo();

break;

case 3: cogeo();

break;

case 4: cout<<"\n\n \t\t THANK YOU FOR USING MATH LAB...!!";

getch();

exit(0);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION FOR TRIGONOMETRY

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void trigo()

{

clrscr();

cout<<"\n\n\n";

cout<<"\t\t MAIN MENU FOR TRIGONOMETRY \n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n";

cout<<"\t\tMENU \n\n";

cout<<"\t\t<1>VALUE OF SINE \n";

cout<<"\t\t<2>VALUE OF COSINE \n";

cout<<"\t\t<3>VALUE OF TANGENT \n";

cout<<"\t\t<4>VALUE OF COSECANT \n";

cout<<"\t\t<5>VALUE OF SECANT \n";

cout<<"\t\t<6>VALUE OF COTANGENT \n";

cout<<"\t\t<7>SOME TRIGONOMETRIC IDENTITIES \n";

cout<<"\t\t<8>RETURN TO MAIN MENU \n\n";

cout<<"\t\tENTER YOUR CHOICE : ";

int x;

cin>>x;

switch(x)

{

case 1 : clrscr();

double x,y;

cout<<"\n\n\n \t\t Enter the angle in degrees :";

cin>>x;

y = x \* (pi/180);

cout<<"\n\n\n \t\t The value of sin("<<x<<") is :"<< sin(y);

break;

case 2 : clrscr();

cout<<"\n\n\n \t\t Enter the angle in degrees :";

cin>>x;

y = x \* (pi/180);

cout<<"\n\n\n \t\t The value of cos("<<x<<") is :"<< cos(y);

break;

case 3 : clrscr();

cout<<"\n\n\n \t\t Enter the angle in degrees :";

cin>>x;

y = x \* (pi/180);

cout<<"\n\n\n \t\t The value of tan("<<x<<") is :"<< tan(y);

break;

case 4 : clrscr();

cout<<"\n\n\n \t\t Enter the angle in degrees :";

cin>>x;

y = x \* (pi/180);

cout<<"\n\n\n \t\t The value of cosec("<<x<<") is :"<< 1/(sin(y));

break;

case 5 : clrscr();

cout<<"\n\n\n \t\t Enter the angle in degrees :";

cin>>x;

y = x \* (pi/180);

cout<<"\n\n\n \t\t The value of sec("<<x<<") is :"<< 1/(cos(y));

break;

case 6 : clrscr();

cout<<"\n\n\n \t\t Enter the angle in degrees :";

cin>>x;

y = x \* (pi/180);

cout<<"\n\n\n \t\t The value of cot("<<x<<") is :"<< 1/(tan(y));

break;

case 7 : clrscr();

cout<<"\n\n\n \t\t SOME GENEREAL IDENTITTIES OF TRIGONOMETRY ARE : "<<endl;

cout<<"\t\t<1> sinx.sinx + cosx.cosx = 1";

cout<<"\n\t\t<2> 1 + cotx.cotx = cosecx.cosecx";

cout<<"\n\t\t<3> 1 + tanx.tanx = cotx.cotx";

cout<<"\n\t\t<4> sin(x + y) = sinx.cosy + siny.cosx";

cout<<"\n\t\t<5> cos(x + y) = cosx.cosy + sinx.siny";

cout<<"\n\t\t<6> tan(x + y) = (tanx + tan y)/(1 - tanx.tany)";

cout<<"\n\t\t<7> sin(3x) = 3sinx - 4.sinx.sinx.sinx";

cout<<"\n\t\t<8> cos(3x) = 4.cosx.cosx.cosx - 3.cosx";

cout<<"\n\t\t<9> tan(3x) = (3.tanx - tanx.tanx.tanx)/(1 - 3.tanx.tanx)";

cout<<"\n\t\t<10> sinx.sin(60 - x).sin(60 + x) = (sin3x)/4";

cout<<"\n\t\t<11> cosnx.cos(60 - x).cos(60 + x) = (cos3x)/4";

cout<<"\n\t\t<12> tanx.tan(60 - x).tan(60 + x) = tan3x";

break;

case 8 : wel();

break;

}

cout<<"\n\n\n \t\tDo you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

trigo();

else

wel();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION FOR ALGEBRA

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void algo()

{

clrscr();

cout<<"\n\n\n";

cout<<"\t\t MAIN MENU FOR ALGEBRA \n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n";

cout<<"\t\tMENU \n\n";

cout<<"\t\t<1>ADDITION \n";

cout<<"\t\t<2>SUBTRACTION \n";

cout<<"\t\t<3>MULTIPLICATION \n";

cout<<"\t\t<4>DIVISION \n";

cout<<"\t\t<5>REMAINDER \n";

cout<<"\t\t<6>PERMUTATIONS \n";

cout<<"\t\t<7>COMBINATIONS \n";

cout<<"\t\t<8>FACTORIAL \n";

cout<<"\t\t<9>SEQUENCE AND SERIES \n";

cout<<"\t\t<10>RETURN TO MAIN MENU \n\n";

cout<<"\t\tENTER YOUR CHOICE : ";

int x;

cin>>x;

switch(x)

{

case 1 : clrscr();

double x,y,z;

cout<<"\n\n\n \t\t ADDITION";

cout<<"\n\n\n \t\t Form : A + B";

cout<<"\n\n\n \t\t Enter the first number :";

cin>>x;

cout<<"\n\n\n \t\t Enter the second number :";

cin>>y;

z = x+y;

cout<<"\n\n\n \t\t The sum of "<<x<<" and "<<y<<" is : "<<z;

break;

case 2 : clrscr();

cout<<"\n\n\n \t\t SUBTRACTION";

cout<<"\n\n\n \t\t Form : A - B";

cout<<"\n\n\n \t\t Enter the first number :";

cin>>x;

cout<<"\n\n\n \t\t Enter the second number :";

cin>>y;

z = x-y;

cout<<"\n\n\n \t\t The difference between "<<x<<" and "<<y<<" is : "<<z;

break;

case 3 : clrscr();

cout<<"\n\n\n \t\t MULTIPLICATION";

cout<<"\n\n\n \t\t Form : A x B";

cout<<"\n\n\n \t\t Enter the first number :";

cin>>x;

cout<<"\n\n\n \t\t Enter the second number :";

cin>>y;

z = x\*y;

cout<<"\n\n\n \t\t The product of "<<x<<" and "<<y<<" is : "<<z;

break;

case 4 : division();

break;

case 5 : remain();

break;

case 6 : clrscr();

cout<<"\n\n\n \t\t PERMUTATION ";

cout<<"\n\tEnter Total Number of objects : ";

cin>>x;

cout<<"\n\tObject are to be arranged in how many places? : ";

cin>>y;

perms(x,y);

getch();

break;

case 7 : clrscr();

cout<<"\n\n\n \t\t COMBINATION";

cout<<"\n\tEnter Total Number of objects : ";

cin>>x;

cout<<"\n\tHow many objects are to be selected? : ";

cin>>y;

combo(x,y);

getch();

break;

case 8 : clrscr();

cout<<"\n\n\n \t\t FACTORIAL";

cout<<"\n\n\n \t\t Form : n!";

cout<<"\n\n\n \t\t Enter the value of n :";

cin>>x;

y=1;

for(int i=x; i>0; --i)

y=y\*i;

cout<<"\n\n\n \t\t The value of "<<x<<"! is : "<<y;

break;

case 9 : ss();

break;

case 10 : wel();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

algo();

else

wel();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION FOR CO-ORDINATE GEOMETRY

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void cogeo()

{

clrscr();

cout<<"\n\n\n";

cout<<"\t\t MAIN MENU FOR CO-ORDONATE GEOMETRY \n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n";

cout<<"\t\tMENU \n\n";

cout<<"\t\t<1>AREA \n";

cout<<"\t\t<2>POINT \n";

cout<<"\t\t<3>STRAIGHT LINES \n";

cout<<"\t\t<4>CIRCLE \n";

cout<<"\t\t<5>PARABOLA \n";

cout<<"\t\t<6>ELLIPSE \n";

cout<<"\t\t<7>HYPERBOLA \n";

cout<<"\t\t<8>RETURN TO MAIN MENU \n\n";

cout<<"\t\tENTER YOUR CHOICE : ";

int x;

cin>>x;

switch(x)

{

case 1: area();

break;

case 2: point();

break;

case 3: line();

break;

case 4: circle();

break;

case 5: parabola();

break;

case 6: ellipse();

break;

case 7: hyperbola();

break;

case 8: wel();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

cogeo();

else

wel();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION FOR WELCOME SCREEN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

clrscr();

cout<<"\n\n\n\n";

cout<<" \t \t \t MAHARAJA AGRASEN VIDYALYA";

cout<<"\n\n\n";

cout<<"\t \t \t \tWELCOME TO MATH LAB";

cout<<"\n\n\n\n";

cout<<"Math Lab will change your experience with mathematics in a "

<<"unique way with the help of Turbo C++. With Math Lab you can "

<<"get answers to various problems of algebra, co-ordinate geometry "

<<"and trigonometry. Enjoy....!!";

cout<<"\n\n Solve this simple problem to Enter MATH LAB : ";

cout<<"\n 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = ";

int x;

cin>>x;

if(x==55)

wel();

else

{

cout<<"\n\n \t Sorry! Wrong answer! Try again!";

getch();

main();

}

getch();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTIONS USED IN THE PROGRAM

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void perms(int c,int d) //For Permutation

{

double x=1,y=1,z;

if(c<d)

{

cout<<"\n\tPlaces can not be greater than total objects.";

getch();

exit(0);

}

for(int t=c;t>0;--t)

x=x\*t;

for(int r=c-d;r>0;--r)

y=y\*r;

if(c==d)

{

z=x;

cout<<"\n\tThe objects can be arranged in "<<z<<" ways..!!!";

}

else

{

z=x/y;

cout<<"\n"<<"\t"<<c<<" objects can be arranged in "<<d<<" places in these many ways: "<<z;

}

}

void combo(int c,int d) // For Combination

{

double x=1,y=1,z=1,f;

if(c<d)

{

cout<<"\n\tSelection can not be greater than total objects.";

getch();

exit(0);

}

for(int t=c;t>0;--t)

x=x\*t;

for(int r=c-d;r>0;--r)

y=y\*r;

for(int s=d;s>0;--s)

z=z\*s;

if(c==d)

{

cout<<"\n\tThe object can be selected in 1 way..!!!";

}

else

{

f=x/(y\*z);

cout<<"\n"<<"\t"<<d<<" objects can be selected from "<<c<<" objects in these many ways: "<<f;

}

}

void division() // For Division

{

clrscr();

double x,y,z;

cout<<"\n\n\n \t\t DIVISION";

cout<<"\n\n\n \t\t Form : A / B";

cout<<"\n\n\n \t\t Enter the first number :";

cin>>x;

cout<<"\n\n\n \t\t Enter the second number :";

cin>>y;

if(y==0)

{

cout<<"\n\n\n \t\t Divisor (ie. B) can not be zero...!!";

cout<<"\n\n\n \t\t Try Again...!!";

getch();

division();

}

else

{

z = x/y;

cout<<"\n\n\n \t\t The quoteint of "<<x<<" and "<<y<<" is : "<<z;

}

}

void remain() // For Remainder

{

clrscr();

int x,y,z;

cout<<"\n\n\n \t\t REMAINDER";

cout<<"\n\n\n \t\t Form : A / B";

cout<<"\n\n\n \t\t Enter the first number :";

cin>>x;

cout<<"\n\n\n \t\t Enter the second number :";

cin>>y;

if(y==0)

{

cout<<"\n\n\n \t\t Divisor (ie. B) can not be zero...!!";

cout<<"\n\n\n \t\t Try Again...!!";

getch();

remain();

}

else

{

z = x%y;

cout<<"\n\n\n \t\t The remainder of "<<x<<" and "<<y<<" is : "<<z;

}

}

void ss() // For Sequence and Series

{

clrscr();

int n;

cout<<"\n\n\n \t\t SEQUENCE AND SERIES";

cout<<endl<<endl;

cout<<"\n\n\n \t\t SELECT YOUR CHOICE FROM BELOW MENU";

cout<<"\n \t\t <1> ARITHMETIC PROGRESSION";

cout<<"\n \t\t <2> GEOMETRIC PROGRESSION";

cout<<"\n\n \t\t ENTER YOUR CHOICE :";

cin>>n;

switch(n)

{

case 1 : clrscr();

double w,x,y,z;

cout<<"\n\n\n \t\t ARITHMETIC PROGRESSION";

cout<<"\n\n\n \t\t FINDING THE SUM OF AN A.P.";

cout<<"\n\n \t\t Form : Sum = (n/2)(2a + (n-1)d)";

cout<<"\n\n \t\t Enter the First Term (a) :";

cin>>w;

cout<<"\n \t\t Enter the Common Difference (d) :";

cin>>x;

cout<<"\n \t\t Enter the Number of Terms (n) :";

cin>>y;

z=(y/2)\*((2\*w) + (y-1)\*x);

cout<<"\n\n \t\t The Sum of The A.P. is :"<<z;

getch();

break;

case 2 : clrscr();

cout<<"\n\n\n \t\t GEOMATRIC PROGRESSION";

cout<<"\n\n\n \t\t FINDING THE SUM OF AN G.P.";

cout<<"\n\n \t\t Form : Sum = [(a)(r^n - 1)]/(r-1)";

cout<<"\n\n \t\t Enter the First Term (a) : ";

cin>>w;

cout<<"\n \t\t Enter the Common Ratio (r) : ";

cin>>x;

cout<<"\n \t\t Enter the Number of Terms (n) : ";

cin>>y;

if(x==1)

z=y\*w;

else

z=(w\*(pow(x,y) - 1))/(x-1);

cout<<"\n\n \t\t The Sum of The G.P. is : "<<z;

getch();

break;

}

cout<<"\n\n \t\t Do you want to continue with Seqyence and Series? (Y/N) :";

char s;

cin>>s;

if(s=='Y'||s=='y')

ss();

else

algo();

getch();

}

void area() // For area

{

clrscr();

int n;

double w,x,y,z;

cout<<" \n\n\n \t\t AREA OF DIFFERENT GEOMETRICAL FIGURES"

<<" \n\n\n \t\t <1> SQUARE "

<<" \n \t\t <2> RECTANGLE "

<<" \n \t\t <3> CIRCLE "

<<" \n \t\t <4> TRIANGLE "

<<" \n \t\t <5> ELLIPSE "

<<" \n \t\t <6> TRAPEZIUM "

<<" \n \t\t <7> RETURN TO PREVIOUS MENU "

<<" \n\n \t\t ENTER YOUR CHOICE : ";

cin>>n;

switch(n)

{

case 1 : clrscr();

cout<<"\n\n\n \t\t\t AREA OF A SQUARE"

<<"\n\n \t\t Formula : (Side)^2 "

<<" \n\n \t\t Enter length of side of square : ";

cin>>x;

z = x \* x ;

cout<<"\n\n \t\t Area of the square of side "<<x<<" is : "<<z;

break;

case 2 : clrscr();

cout<<"\n\n\n \t\t\t AREA OF A RECTANGLE"

<<"\n\n \t\t Formula : BASE \* HEIGHT "

<<"\n\n \t\t Enter length of base of rectangle : ";

cin>>x;

cout<<"\n\n \t\t Enter height of the rectangle : ";

cin>>y;

z = x \* y ;

cout<<"\n\n \t\t Area of the rectangle is : "<<z;

break;

case 3 : clrscr();

cout<<"\n\n\n \t\t\t AREA OF A CIRCLE"

<<"\n\n \t\t Formula : (22/7)r^2 "

<<"\n\n \t\t Enter the radius of the circle : ";

cin>>x;

z = pi \* x \* x ;

cout<<"\n\n \t\t Area of the circle of radius "<<x<<" is : "<<z;

break;

case 4 : clrscr();

cout<<"\n\n\n \t\t\t AREA OF A TRIANGLE"

<<"\n\n \t\t Formula : (1/2)(base)(height) "

<<"\n\n \t\t Enter length of base of triangle : ";

cin>>x;

cout<<"\n\n \t\t Enter height of the triangle : ";

cin>>y;

z=0.5\*x\*y ;

cout<<"\n\n \t\t Area of the triangle of base "<<x<<" and height "<<y<<" is :"<<z;

break;

case 5 : clrscr();

cout<<"\n\n\n \t\t\t AREA OF A ELLIPSE"

<<"\n\n \t\t Formula : (22/7)A^B "

<<"\n\n \t\t Enter the length of semi major axis : ";

cin>>x;

cout<<"\n\n \t\t Enter the length of semi minor axis : ";

cin>>y;

z = pi \* x \* y ;

cout<<"\n\n \t\t Area of the ellipse is : "<<z;

break;

case 6 : clrscr();

cout<<"\n\n\n \t\t\t AREA OF A TRAPEZIUM"

<<"\n\n \t\t Formula : (1/2)(sum of length of parallel sides)(height)"

<<"\n\n \t\t Enter lengths of parallel sides : ";

cin>>x>>y;

cout<<"\n\n \t\t Enter height of the trapezium : ";

cin>>w;

z = 0.5\*(x + y)\*w ;

cout<<"\n\n \t\t Area of the trapezium is : "<<z;

break;

case 7 : cogeo();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

area();

else

cogeo();

}

void point() // For point

{

clrscr();

int n;

double w,x,y,z;

cout<<" \n\n\n \t\t\t POINT"

<<" \n\n\n \t\t <1> DISTANCE BETWEEN TWO POINTS "

<<" \n \t\t <2> SHIFTING OF ORIGIN "

<<" \n \t\t <3> CENROID "

<<" \n \t\t <4> AREA OF TRIANGLE "

<<" \n \t\t <5> RETURN TO PREVIOUS MENU "

<<" \n\n \t\t ENTER YOUR CHOICE : ";

cin>>n;

switch(n)

{

case 1 : clrscr();

float x1,x2,y1,y2;

double z,x;

cout<<"\n\n\n \t\t\t DISTANCE BETWEEN TWO POINTS"

<<"\n\n \t\t Formula : ((x1 - x2)^2 + (y1 - y2)^2)^(1/2)"

<<" \n\n \t\t Enter co-ordinates of first point (x1 y1) : ";

cin>>x1 >>y1;

cout<<" \n\n \t\t Enter co-ordinates of second point (x2 y2) : ";

cin>>x2 >>y2;

z = pow(pow(x1-x2,2) + pow(y1-y2,2),0.5);

cout<<"\n\n \t\t DISTANCE BETWEEN ("<<x1<<","<<y1<<") and ("<<x2<<","<<y2<<") is : "<<z;

break;

case 2 : clrscr();

cout<<"\n\n\n \t\t\t SHIFTING OF ORIGIN"

<<"\n\n \t\t Form : X = x - h"

<<"\n\n \t\t Form : Y = y - k"

<<"\n\n \t\t Enter co-ordinates of the point (x y): ";

cin>>x1 >>y1;

cout<<"\n\n \t\t Enter co-ordinates of shifted origin (h k): ";

cin>>x2 >>y2;

cout<<"\n\n \t\t Co-ordinates of the point along the shifted origin are :"<<"("<<x1-x2<<","<<y1-y2<<")";

break;

case 3 : clrscr();

float x3, y3;

cout<<"\n\n\n \t\t\t CENTROID"

<<"\n\n \t\t Form : X = (x1+x2+x3)/3"

<<"\n\n \t\t Form : Y = (y1+y2+y3)/3"

<<"\n\n \t\t Enter co-ordinates of first point (x1 y1) : ";

cin>>x1 >>y1;

cout<<" \n\n \t\t Enter co-ordinates of second point (x2 y2) : ";

cin>>x2 >>y2;

cout<<" \n\n \t\t Enter co-ordinates of third point (x3 y3) : ";

cin>>x3 >>y3;

cout<<" \n\n \t\t The co-ordinates of centroid are :"<<"("<<(x1+x2+x3)/3<<","<<(y1+y2+y3)/3<<")";

break;

case 4 : clrscr();

cout<<"\n\n\n \t\t\t AREA OF TRIANGLE"

<<"\n\n \t\t Form : AREA = (1/2)\*( x1\*(y2-y3) + x2\*(y3-y1) + x3\*(y1-y2) )"

<<"\n\n \t\t Enter co-ordinates of first point (x1 y1) : ";

cin>>x1 >>y1;

cout<<" \n\n \t\t Enter co-ordinates of second point (x2 y2) : ";

cin>>x2 >>y2;

cout<<" \n\n \t\t Enter co-ordinates of third point (x3 y3) : ";

cin>>x3 >>y3;

z=((x1\*(y2-y3)) + (x2\*(y3-y1)) + (x3\*(y1-y2)))/2;

cout<<" \n\n \t\t Area of the triangle is : "<<z;

break;

case 5 : cogeo();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

point();

else

cogeo();

}

void line() // For line

{

clrscr();

int n,x1,x2,y1,y2;

double z;

cout<<"\n\n\n \t\t STRAIGHT LINE ";

cout<<"\n\n\n \t\t <1> DISTANCE BETWEEN TWO POINTS"

<<"\n\t\t <2> SLOPE"

<<"\n\t\t <3> RETURN TO PREVIOUS MENU"

<<"\n\n \t\t ENTER YOUR CHOICE : ";

cin>>n;

switch(n)

{

case 1 :clrscr();

cout<<"\n\n\n \t\tDISTANCE BETWEEN TWO POINTS"

<<"\n\n \t\tFormula : [(x1-x2)^2 + (y1-y2)^2]^(1/2)";

cout<<"\n\n \t\t Enter co-ordinates of first point(x1 y1) : ";

cin>>x1>>y1;

cout<<"\n\n \t\t Enter co-ordinates of second point(x2 y2) : ";

cin>>x2>>y2;

z=pow((pow(x1-x2,2) + pow(y1-y2,2)),(0.5));

cout<<"\n\n \t\t Distance between the two points is : "<<z;

break;

case 2 :clrscr();

cout<<"\n\n\n \t\t SLOPE OF A LINE"

<<"\n\n\n \t\t Formula : (y1-y2)/(x1-x2)";

cout<<"\n\n \t\t Enter co-ordinates of first point(x1, y1) : ";

cin>>x1>>y1;

cout<<"\n\n \t\t Enter co-ordinates of second point(x2, y2) : ";

cin>>x2>>y2;

if(x1==x2)

{

cout<<"Error";

line( );

}

else

{

z=(y1-y2)/(x1-x2);

cout<<"\n\n \t\t Slope of the line is : "<<z;

}

break;

case 3 :cogeo();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

line();

else

cogeo();

}

void circle() //For circle

{

clrscr();

int n,x1,y1,f,g,c;

double r;

cout<<"\n\n\n \t\t CIRCLE"

<<"\n\n\n \t\t <1> EQUATION OF CIRCLE"

<<"\n \t\t <2> RADIUS AND CENTER"

<<"\n \t\t <3> TANGENT AND NORMAL"

<<"\n \t\t <4> DIRECTOR CIRCLE"

<<"\n \t\t <5> RETURN TO PREVIOUS MENU"

<<"\n\n \t\t ENTER YOUR CHOICE : ";

cin>>n;

switch(n)

{

case 1 :clrscr();

cout<<"\n\n\n \t\t EQUATION OF CIRCLE";

cout<<"\n \t\t Form : x^2 + y^2 = r^2";

cout<<"\n\n\n \t\t Enter radius of circle : ";

cin>>r;

cout<<"\n\n\n \t\t Equation of circle is : x^2 + y^2 = "<<(r\*r);

break;

case 2 :clrscr();

cout<<"\n\n\n \t\t CENTER AND RADIUS";

cout<<"\n \t\t Form : x^2 + y^2 +2gx +2fy +c = 0";

cout<<"\n\n\n \t\t Enter g, f & c (in order) : ";

cin>>g>>f>>c;

cout<<"\n\n\n \t\t Center of circle is : ("<<-g<<" , "<<-f<<")";

r=pow((pow(g,2) + pow(f,2) - c),0.5);

cout<<"\n\n \t\t Radius of circle is : "<<r;

break;

case 3 :clrscr();

cout<<"\n\n\n \t\t TANGENT AND NORMAL";

cout<<"\n \t\t Form : x^2 + y^2 = r^2";

cout<<"\n\n\n \t\t Enter radius of the circle : ";

cin>>r;

cout<<"\n\n \t\t Enter a point (on circle) : ";

cin>>x1>>y1;

cout<<"\n\n\n \t\t Tangent to circle is : x."<<x1<<" + y."<<y1<<" = "<<r\*r;

cout<<"\n\n \t\t Normal to circle is : x."<<y1<<" - y."<<x1<<" = 0";

break;

case 4 :clrscr();

cout<<"\n\n\n \t\t DIRECTOR CIRCLE";

cout<<"\n \t\t Form : x^2 + y^2 = r^2";

cout<<"\n\n\n \t\t Enter radius of the circle : ";

cin>>r;

cout<<"\n\n\n \t\t Equation of original circle is : x^2 + y^2 = "<<(r\*r);

cout<<"\n\n\n \t\t Equation of director circle is : x^2 + y^2 = "<<(2\*(pow(r,2)));

break;

case 5 :cogeo();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

circle();

else

cogeo();

}

void parabola() //For Parabola

{

clrscr();

double x1,y1;

double a,k;

int n;

cout<<"\n\n\n \t\t PARABOLA"

<<"\n\n\n \t\t <1> EQUATION OF PARABOLA"

<<"\n \t\t <2> FOCUS AND LENGTH OF LATUS RECTUM"

<<"\n \t\t <3> TANGENT AND NORMAL"

<<"\n \t\t <4> EQUATION OF DIRECTRIX"

<<"\n \t\t <5> RETURN TO PREVIOUS MENU"

<<"\n\n \t\t ENTER YOUR CHOICE : ";

cin>>n;

switch(n)

{

case 1 :clrscr();

cout<<"\n\n\n \t\t EQUATION OF PARABOLA";

cout<<"\n \t\t Form : y^2 = 4 \* a \* x";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n\n \t\t Equation of the parabola is : y^2 = "<<4\*a<<".x";

break;

case 2 :clrscr();

cout<<"\n\n\n \t\t FOCUS AND LENGTH OF LATUS RECTUM";

cout<<"\n \t\t Form : y^2 = 4 \* a \* x";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n\n \t\t The co-ordinates of the Focus are : ("<<a<<",0)";

cout<<"\n\n\n \t\t The Length of the Latus Rectum is : "<<4\*a;

break;

case 3 :clrscr();

cout<<"\n\n\n \t\t TANGENT AND NORMAL";

cout<<"\n \t\t Form : y^2 = 4 \* a \* x";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n \t\t Enter a point (on parabola) : ";

cin>>x1>>y1;

cout<<"\n\n\n \t\t Tangent to parabola is : y."<<y1<<" = "<<2\*a<<".(x + "<<x1<<")";

k = (0 - (y1/(2\*a)));

cout<<"\n\n \t\t Normal to the parabola is : y - "<<y1<<" = "<<k<<"(x - "<<x1<<")";

break;

case 4 :clrscr();

cout<<"\n\n\n \t\t EQUATION OF DIRECTRIX";

cout<<"\n \t\t Form : y^2 = 4 \* a \* x";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n\n \t\t Equation of directrix is : x + ("<<a<<")";

break;

case 5 :cogeo();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

parabola();

else

cogeo();

}

void ellipse() //For ellipse

{

clrscr();

double x1,y1;

double a,b,e;

int n;

cout<<"\n\n\n \t\t ELLIPSE"

<<"\n\n\n \t\t <1> EQUATION OF ELLIPSE"

<<"\n \t\t <2> ECCENTRICITY OF ELLIPSE"

<<"\n \t\t <3> FOCI, LENGTH OF AXES AND LENGTH OF LATUS RECTUM"

<<"\n \t\t <4> TANGENT AND NORMAL"

<<"\n \t\t <5> EQUATION OF DIRECTOR CIRCLE"

<<"\n \t\t <6> RETURN TO PREVIOUS MENU"

<<"\n\n \t\t ENTER YOUR CHOICE : ";

cin>>n;

switch(n)

{

case 1 :clrscr();

cout<<"\n\n\n \t\t EQUATION OF ELLIPSE";

cout<<"\n \t\t Form : (x/a)^2 + (y/b)^2 = 1";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

cout<<"\n\n\n \t\t Equation of the ellipse is : (x^2/"<<a\*a<<") + (y^2/"<<b\*b<<") = 1";

break;

case 2 :clrscr();

cout<<"\n\n\n \t\t ECCENTRICITY OF ELLIPSE";

cout<<"\n \t\t Form : (x/a)^2 + (y/b)^2 = 1";

cout<<"\n \t\t Eccentricity (e) = (1 - (b\*b/a\*a))^0.5 ( if a>b)";

cout<<"\n \t\t Eccentricity (e) = (1 - (a\*a/b\*b))^0.5 ( if a<b)";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

if(a>b)

e = pow(1 - ((b\*b)/(a\*a)),0.5);

if(b>a)

e = pow(1 - ((a\*a)/(b\*b)),0.5);

cout<<"\n\n\n \t\t Eccentricity of the ellipse is : "<<e;

break;

case 3 :clrscr();

cout<<"\n\n\n \t\t FOCI, LENGTH OF AXES AND LENGTH OF LATUS RECTUM";

cout<<"\n \t\t Form : (x/a)^2 + (y/b)^2 = 1";

cout<<"\n \t\t Foci : (a\*e,0) and (-a\*e,0) (if a>b)";

cout<<"\n \t\t Foci : (0,b\*e) and (0,-b\*e) (if a<b)";

cout<<"\n \t\t Length of Major Axis and Minor Axis : 2\*a and 2\*b ( if a>b)";

cout<<"\n \t\t Length of Major Axis and Minor Axis : 2\*b and 2\*a ( if a<b)";

cout<<"\n \t\t Length of Latus Rectum : (2\*b\*b)/a ( if a>b)";

cout<<"\n \t\t Length of Latus Rectum : (2\*a\*a)/b ( if a<b)";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

if(a>b)

{

e = pow(1 - ((b\*b)/(a\*a)),0.5);

cout<<"\n\n \t\t The Foci of the ellipse are : ("<<a\*e<<",0) and ("<< 0 - a\*e<<",0)";

cout<<"\n \t\t The length of Major Axis is : "<<2\*a;

cout<<"\n \t\t The length of Minor Axis is : "<<2\*b;

cout<<"\n \t\t The length of Latus Rectum is : "<< (2\*b\*b)/a;

}

if(b>a)

{

e = pow(1 - ((a\*a)/(b\*b)),0.5);

cout<<"\n\n \t\t The Foci of the ellipse are : (0,"<<b\*e<<") and (0,"<< 0 - b\*e<<")";

cout<<"\n \t\t The length of Major Axis is : "<<2\*b;

cout<<"\n \t\t The length of Minor Axis is : "<<2\*a;

cout<<"\n \t\t The length of Latus Rectum is : "<< (2\*a\*a)/b;

}

break;

case 4 :clrscr();

cout<<"\n\n\n \t\t TANGENT AND NORMAL";

cout<<"\n \t\t Form : (x/a)^2 + (y/b)^2 = 1";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

cout<<"\n\n \t\t Enter a point (on ellipse) : ";

cin>>x1>>y1;

cout<<"\n\n\n \t\t Tangent to ellipse is : (x."<<x1<<")/"<<a\*a<<" + (y."<<y1<<")/"<<b\*b<<" = 1";

cout<<"\n\n \t\t Normal to the ellipse is : "<<a\*a<<".x/"<<x1<<" - "<<b\*b<<".y/"<<y1<<" = "<<(a\*a) - (b\*b);

break;

case 5 :clrscr();

cout<<"\n\n\n \t\t EQUATION OF DIRECTOR CIRCLE";

cout<<"\n \t\t Form : (x/a)^2 + (y/b)^2 = 1";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

cout<<"\n\n\n \t\t Equation of Director Circle is : x^2 + y^2 = "<<(a\*a) + (b\*b);

break;

case 6 :cogeo();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

ellipse();

else

cogeo();

}

void hyperbola() //For hyperbola

{

clrscr();

double x1,y1;

double a,b,e;

int n;

cout<<"\n\n\n \t\t HYPERBOLA"

<<"\n\n\n \t\t <1> EQUATION OF HYPERBOLA"

<<"\n \t\t <2> ECCENTRICITY OF HYPERBOLA"

<<"\n \t\t <3> FOCI, LENGTH OF AXES AND LENGTH OF LATUS RECTUM"

<<"\n \t\t <4> TANGENT AND NORMAL"

<<"\n \t\t <5> EQUATION OF DIRECTOR CIRCLE"

<<"\n \t\t <6> RETURN TO PREVIOUS MENU"

<<"\n\n \t\t ENTER YOUR CHOICE : ";

cin>>n;

switch(n)

{

case 1 :clrscr();

cout<<"\n\n\n \t\t EQUATION OF HYPERBOLA";

cout<<"\n \t\t Form : (x/a)^2 - (y/b)^2 = 1 (if a>b)";

cout<<"\n \t\t Form : (y/b)^2 - (x/a)^2 = 1 (if a<b)";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n\n \t\t Enter the value of b (b not equal to a): ";

cin>>b;

if(a>b)

cout<<"\n\n\n \t\t Equation of the hyperbola is : (x^2/"<<a\*a<<") - (y^2/"<<b\*b<<") = 1";

if(a<b)

cout<<"\n\n\n \t\t Equation of the hyperbola is : (y^2/"<<b\*b<<") - (x^2/"<<a\*a<<") = 1";

break;

case 2 :clrscr();

cout<<"\n\n\n \t\t ECCENTRICITY OF HYPERBOLA";

cout<<"\n \t\t Form : (x/a)^2 - (y/b)^2 = 1 (if a>b)";

cout<<"\n \t\t Form : (y/b)^2 - (x/a)^2 = 1 (if a<b)";

cout<<"\n \t\t Eccentricity (e) = (1 + (b\*b/a\*a))^0.5 ( if a>b)";

cout<<"\n \t\t Eccentricity (e) = (1 + (a\*a/b\*b))^0.5 ( if a<b)";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

if(a>b)

e = pow(1 + ((b\*b)/(a\*a)),0.5);

if(b>a)

e = pow(1 + ((a\*a)/(b\*b)),0.5);

cout<<"\n\n\n \t\t Eccentricity of the hyperbola is : "<<e;

break;

case 3 :clrscr();

cout<<"\n\n\n \t\t FOCI, LENGTH OF AXES AND LENGTH OF LATUS RECTUM";

cout<<"\n \t\t Form : (x/a)^2 - (y/b)^2 = 1 (if a>b)";

cout<<"\n \t\t Form : (y/b)^2 - (x/a)^2 = 1 (if a<b)";

cout<<"\n \t\t Foci : (a\*e,0) and (-a\*e,0) (if a>b)";

cout<<"\n \t\t Foci : (0,b\*e) and (0,-b\*e) (if a<b)";

cout<<"\n \t\t Length of Transverse Axis and Conjugate Axis : 2\*a and 2\*b ( if a>b)";

cout<<"\n \t\t Length of Transverse Axis and Conjugate Axis : 2\*b and 2\*a ( if a<b)";

cout<<"\n \t\t Length of Latus Rectum : (2\*b\*b)/a ( if a>b)";

cout<<"\n \t\t Length of Latus Rectum : (2\*a\*a)/b ( if a<b)";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

if(a>b)

{

e = pow(1 + ((b\*b)/(a\*a)),0.5);

cout<<"\n\n \t\t The Foci of the hyperbola are : ("<<a\*e<<",0) and ("<< 0 - a\*e<<",0)";

cout<<"\n \t\t The length of Transverse Axis is : "<<2\*a;

cout<<"\n \t\t The length of Conjugate Axis is : "<<2\*b;

cout<<"\n \t\t The length of Latus Rectum is : "<< (2\*b\*b)/a;

}

if(b>a)

{

e = pow(1 + ((a\*a)/(b\*b)),0.5);

cout<<"\n\n \t\t The Foci of the hyperbola are : (0,"<<b\*e<<") and (0,"<< 0 - b\*e<<")";

cout<<"\n \t\t The length of Transverse Axis is : "<<2\*b;

cout<<"\n \t\t The length of Conjugate Axis is : "<<2\*a;

cout<<"\n \t\t The length of Latus Rectum is : "<< (2\*a\*a)/b;

}

break;

case 4 :clrscr();

cout<<"\n\n\n \t\t TANGENT AND NORMAL";

cout<<"\n \t\t Form : (x/a)^2 - (y/b)^2 = 1";

cout<<"\n \t\t Form : (y/b)^2 - (x/a)^2 = 1";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

cout<<"\n\n \t\t Enter a point (on hyperbola) : ";

cin>>x1>>y1;

if(a>b)

{

cout<<"\n\n\n \t\t Tangent to hyperbola is : (x."<<x1<<")/"<<a\*a<<" - (y."<<y1<<")/"<<b\*b<<" = 1";

cout<<"\n\n \t\t Normal to the hyperbola is : "<<a\*a<<".x/"<<x1<<" + "<<b\*b<<".y/"<<y1<<" = "<<(a\*a) + (b\*b);

}

if(a<b)

{

cout<<"\n\n\n \t\t Tangent to hyperbola is : (y."<<y1<<")/"<<b\*b<<" - (x."<<x1<<")/"<<a\*a<<" = 1";

cout<<"\n\n \t\t Normal to the hyperbola is : "<<a\*a<<".x/"<<x1<<" + "<<b\*b<<".y/"<<y1<<" = "<<(a\*a) + (b\*b);

}

break;

case 5 :clrscr();

cout<<"\n\n\n \t\t EQUATION OF DIRECTOR CIRCLE";

cout<<"\n \t\t Form : (x/a)^2 - (y/b)^2 = 1 (if a>b)";

cout<<"\n \t\t Form : (y/b)^2 - (x/a)^2 = 1 (if a<b)";

cout<<"\n\n\n \t\t Enter the value of a : ";

cin>>a;

cout<<"\n\n \t\t Enter the value of b (b not equal to a) : ";

cin>>b;

if(a>b)

cout<<"\n\n\n \t\t Equation of Director Circle is : x^2 + y^2 = "<<(a\*a) - (b\*b);

if(a<b)

cout<<"\n\n\n \t\t Equation of Director Circle is : x^2 + y^2 = "<<(b\*b) - (a\*a);

break;

case 6 :cogeo();

break;

}

cout<<"\n\n\n \t\t Do you want to continue(y/n)?";

char cont;

cin>>cont;

if((cont=='y') || (cont=='Y'))

hyperbola();

else

cogeo();

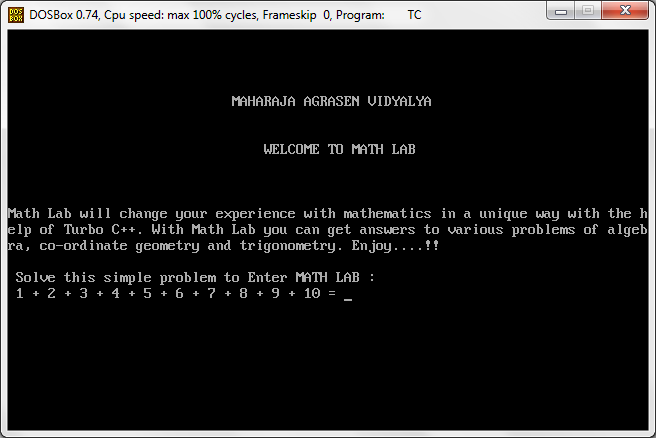
}

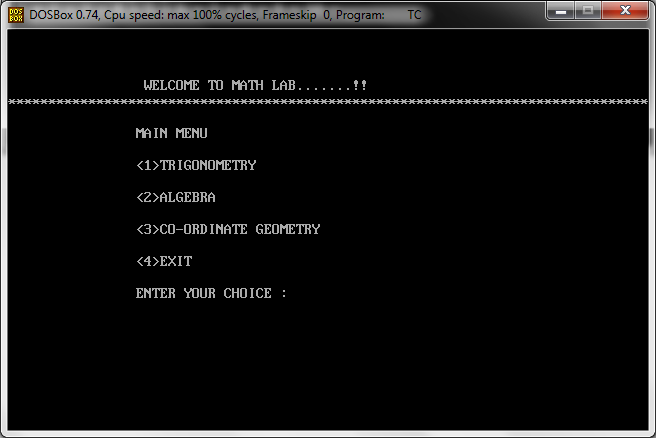
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

END OF THE PROGRAM

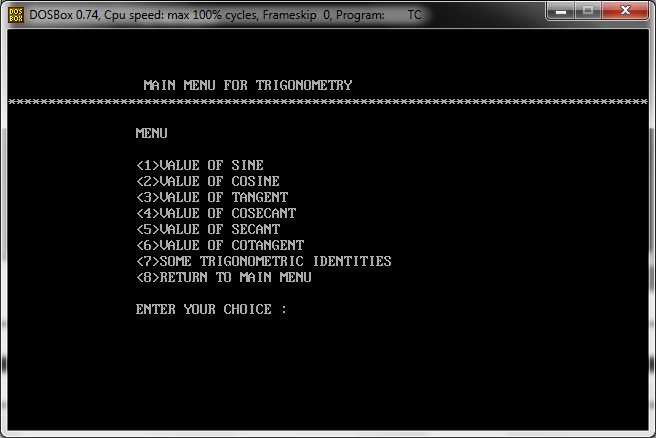
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

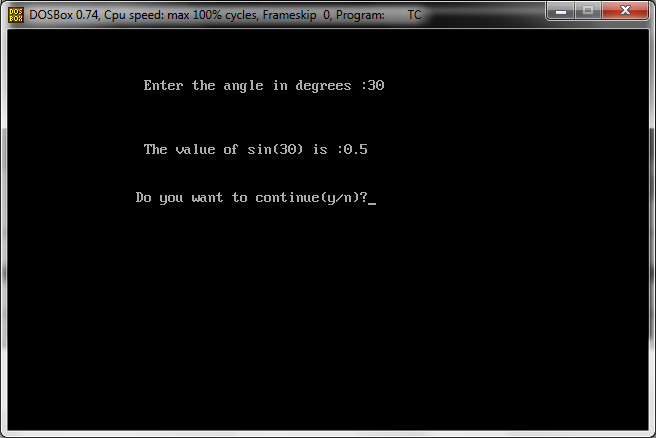
**OUTPUT**

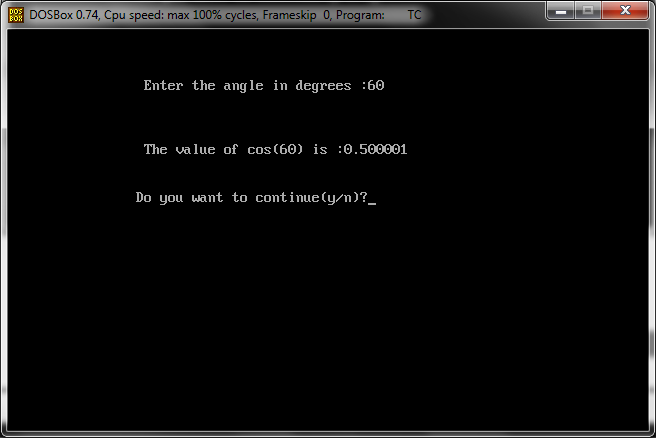
****

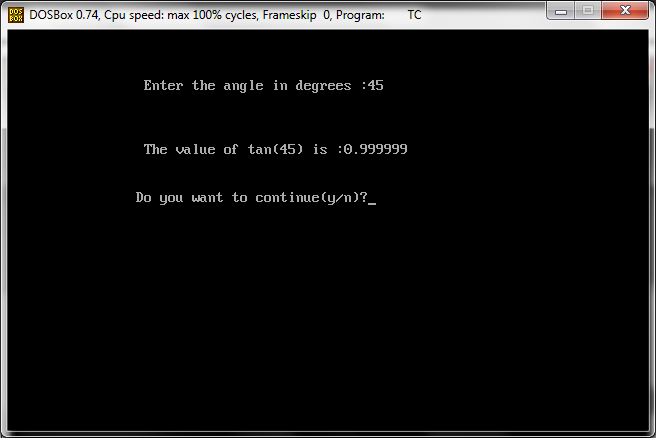
****

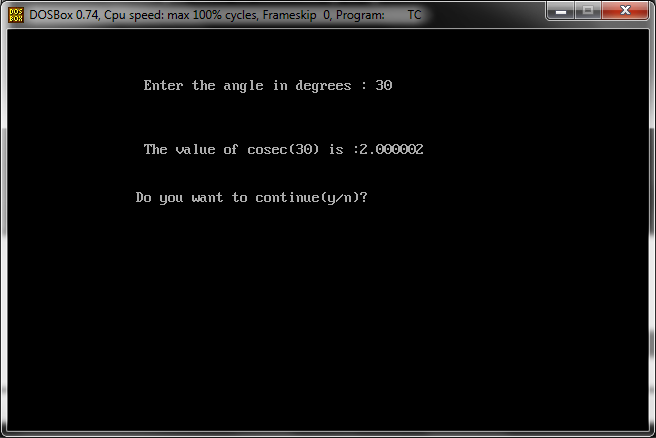
**SECTION:TRIGONOMETRY**

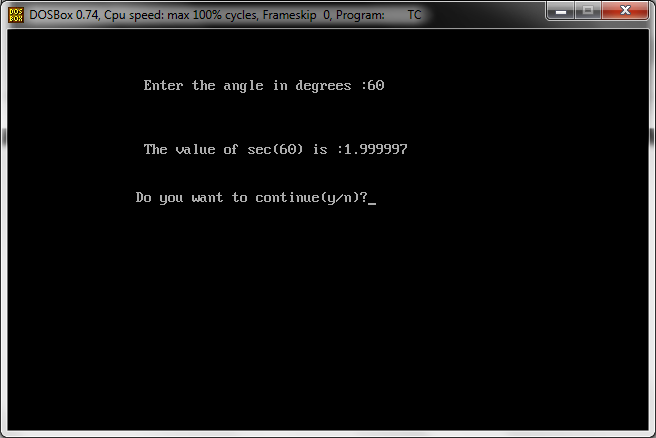
****

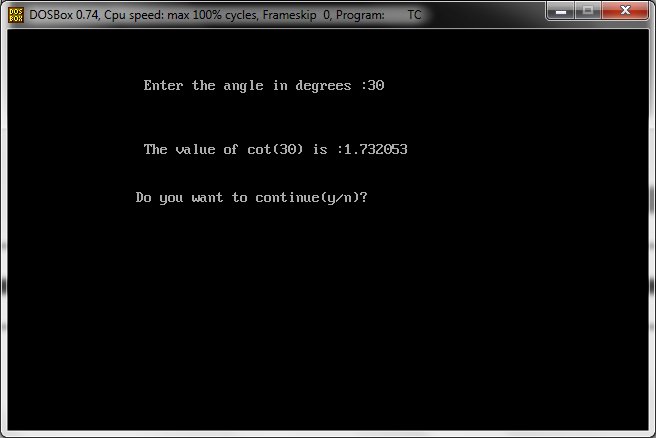
****

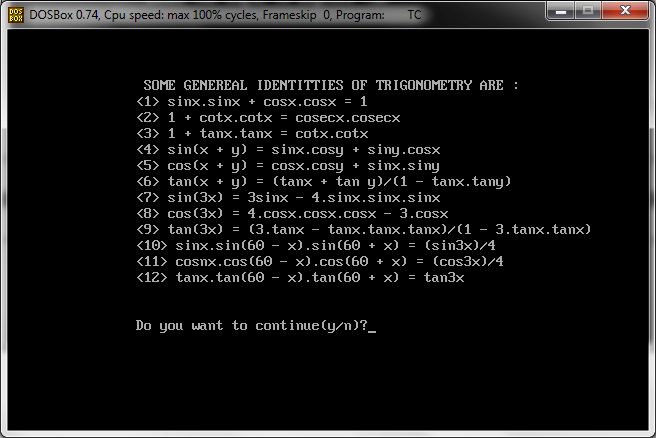
****

****

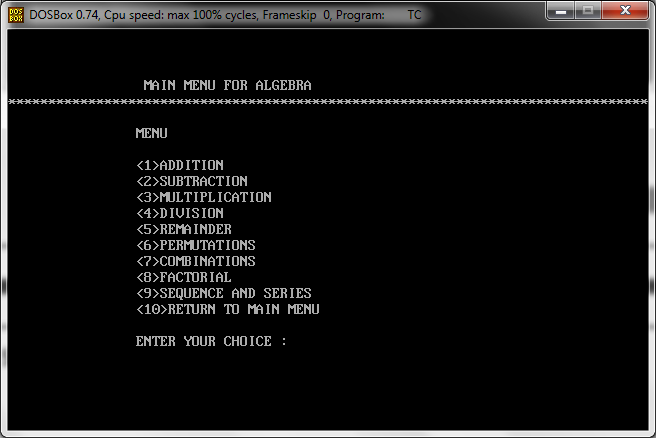
****

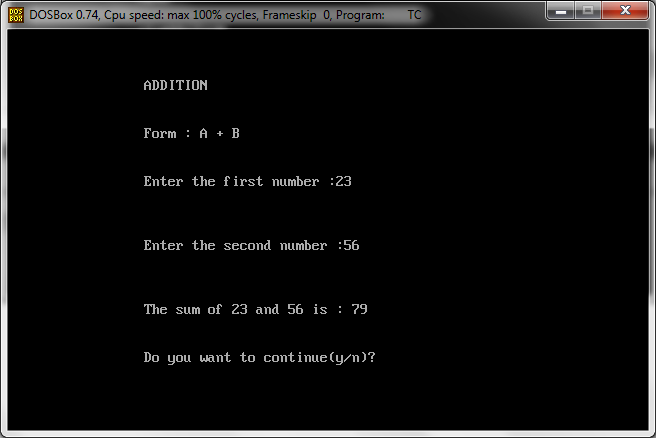
****

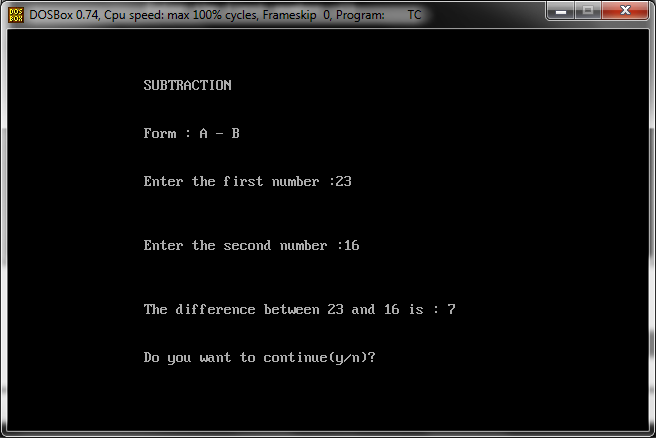
****

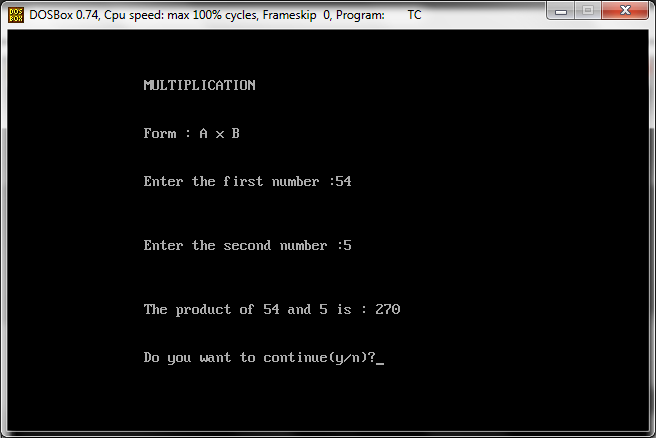
****

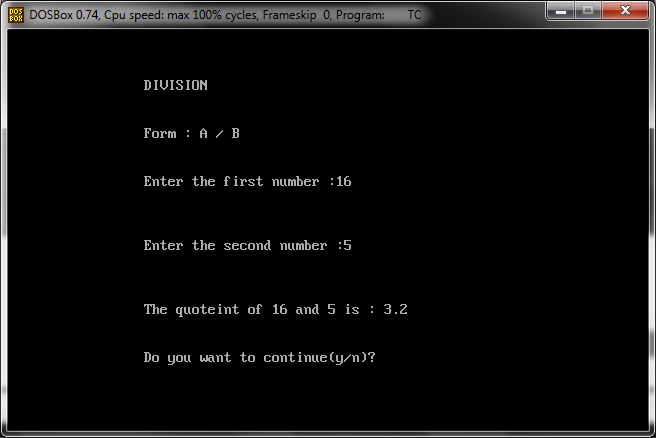
**SECTION: ALGEBRA**

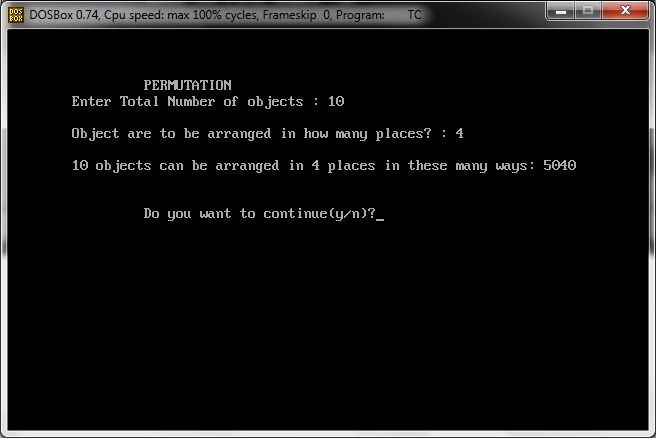
****

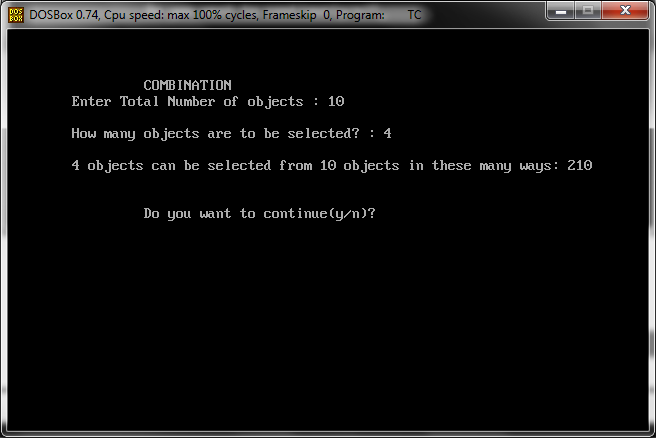
****

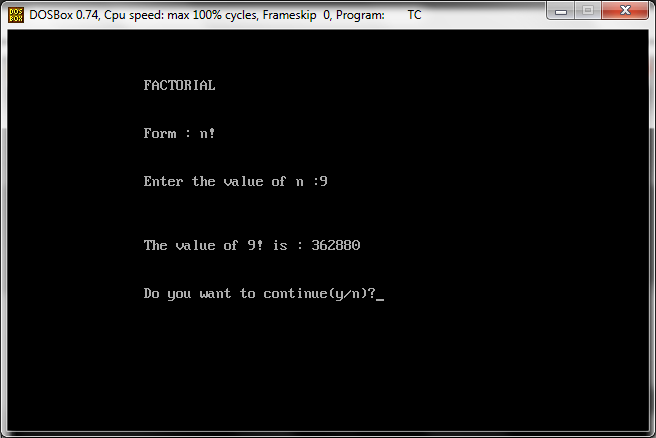
****

****

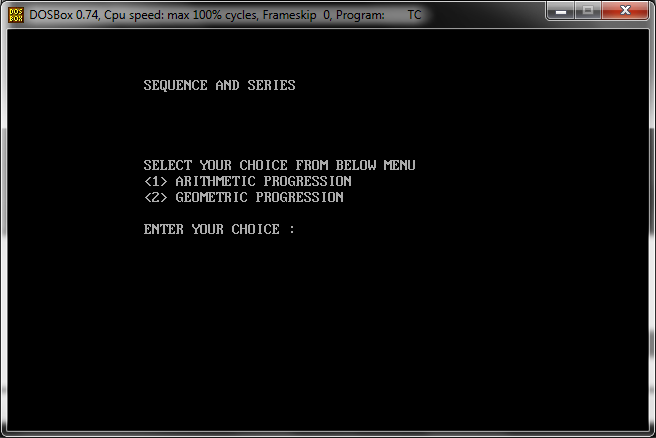
****

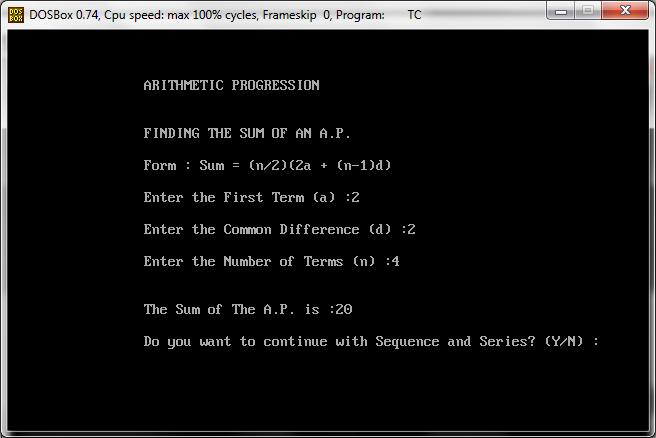
****

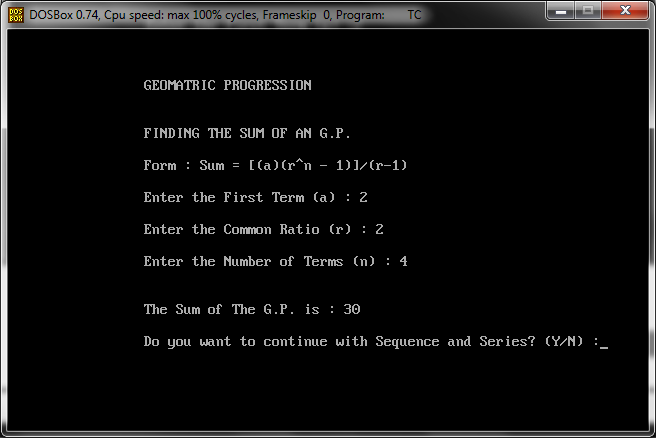
****

****

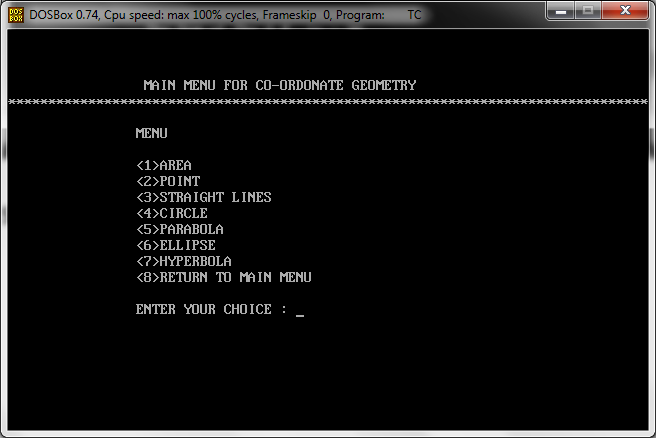
**SECTION: SEQUENCE AND SERIES**

****

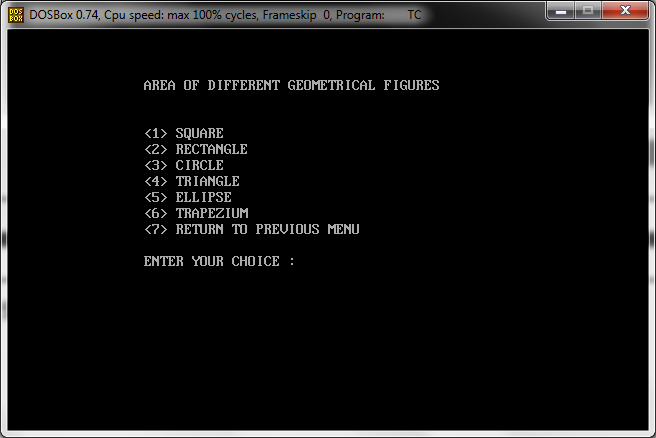
****

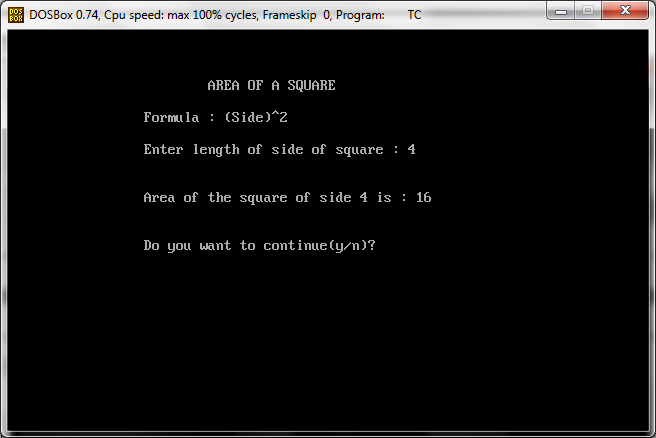
****

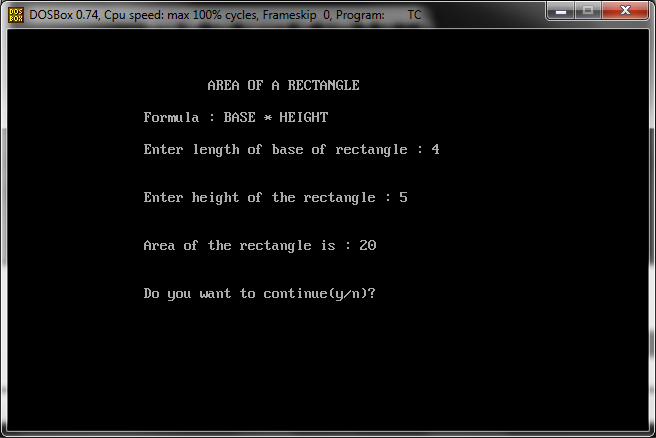
**SECTION: CO-ORDINATE GEOMETRY**

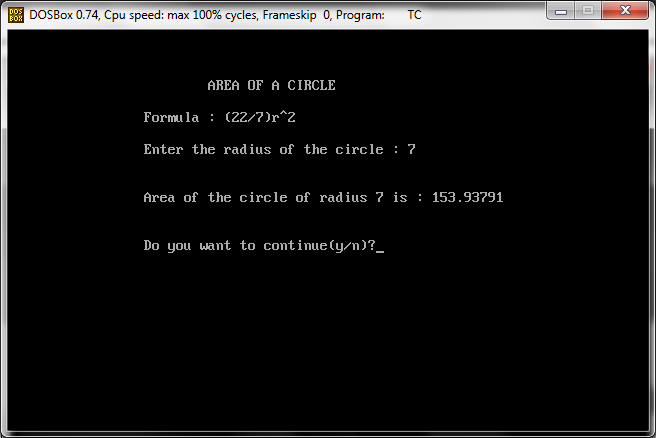
****

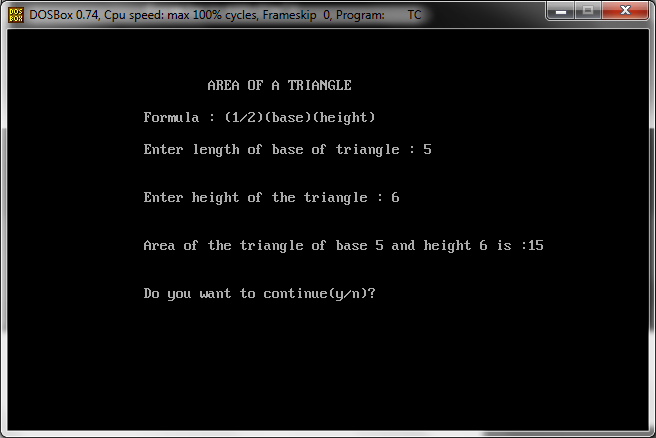
**SECTION: AREA**

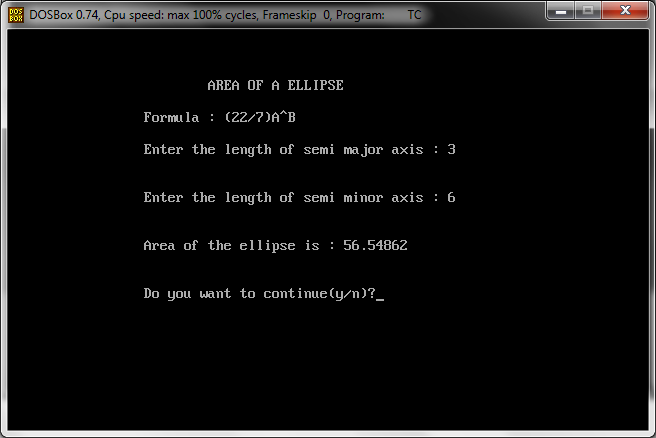
****

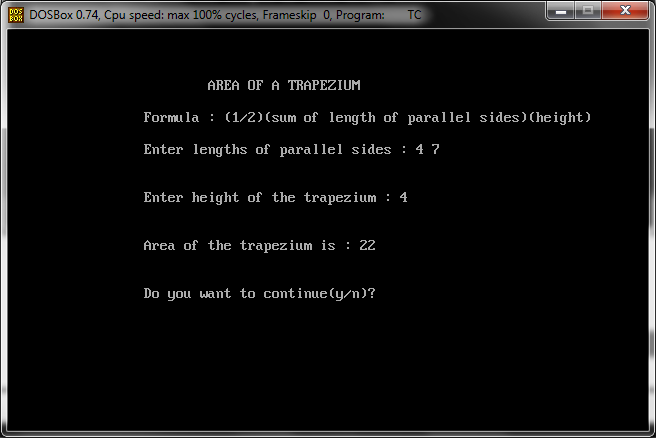
****

****

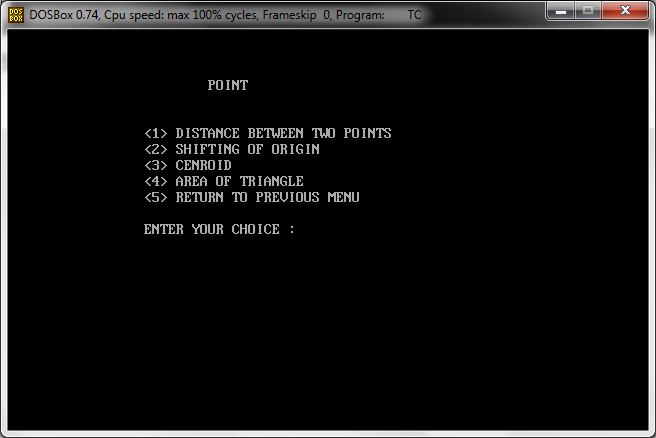
****

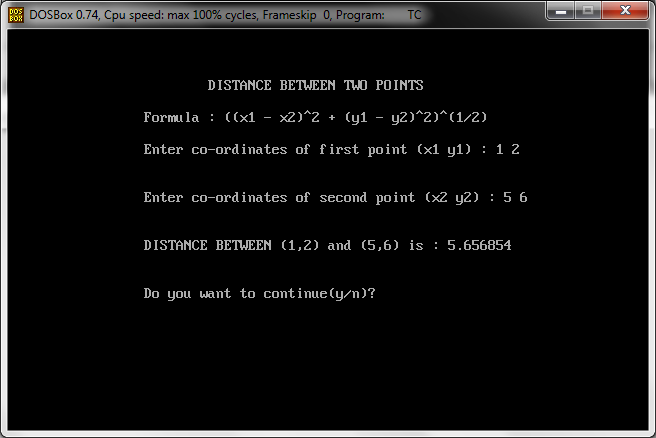
****

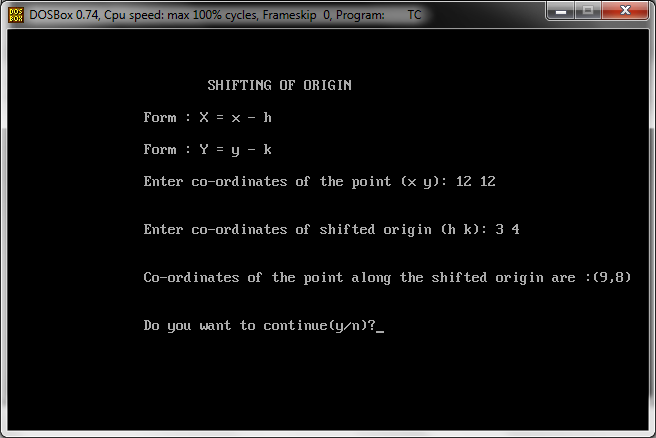
****

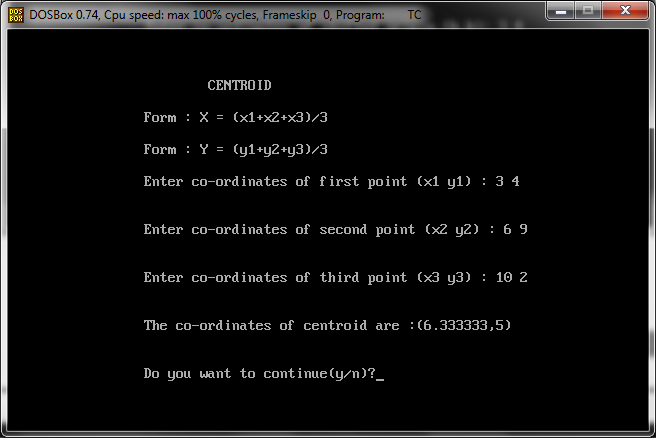
****

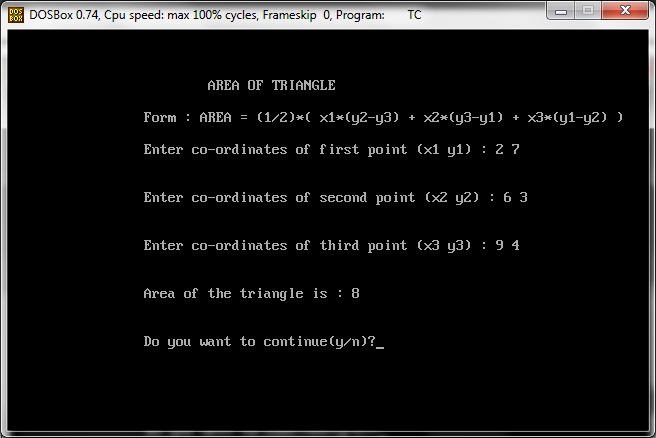
**SECTION: POINT**

****

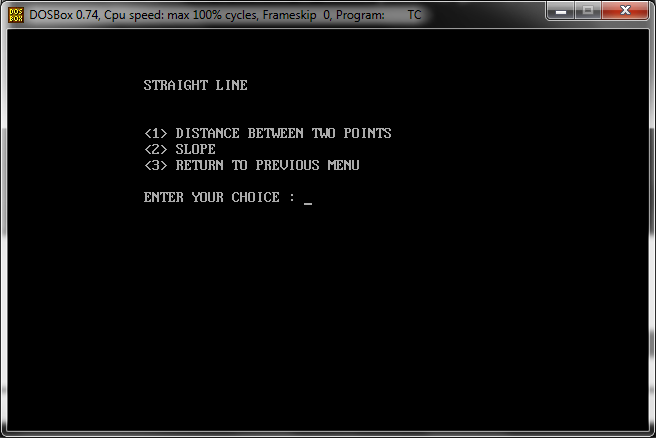
****

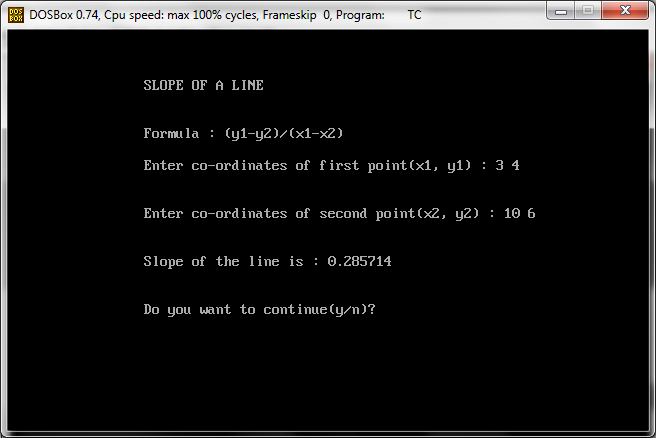
****

****

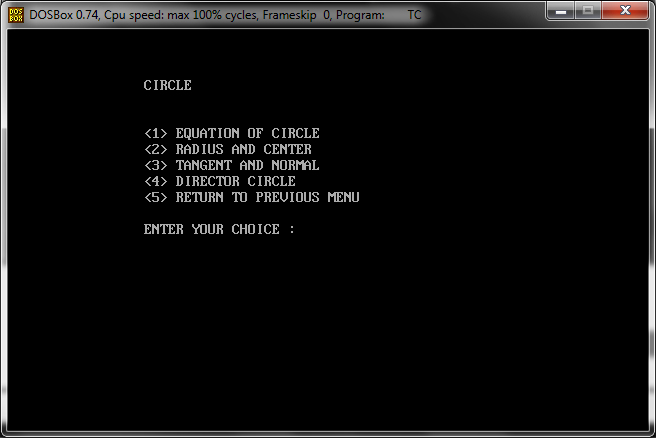
****

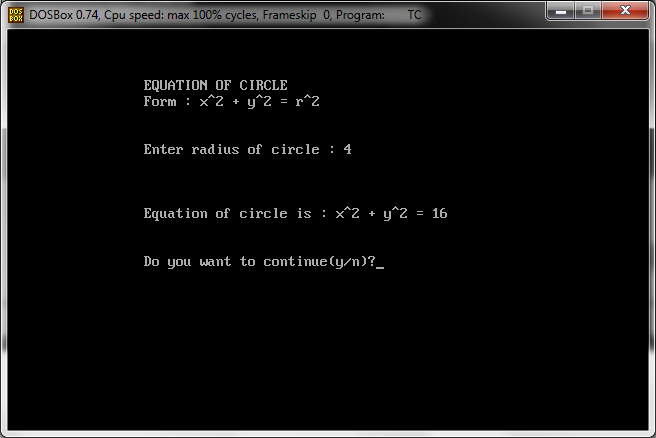
**SECTION: STRAIGHT LINE**

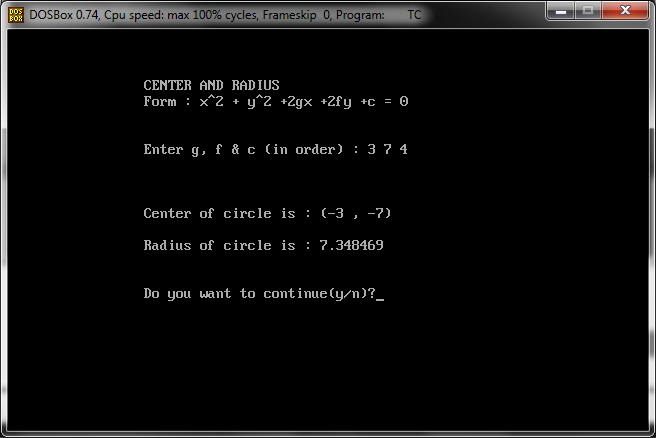
****

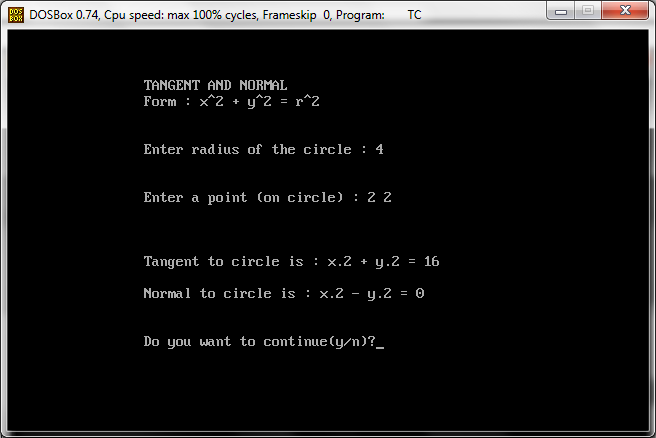
****

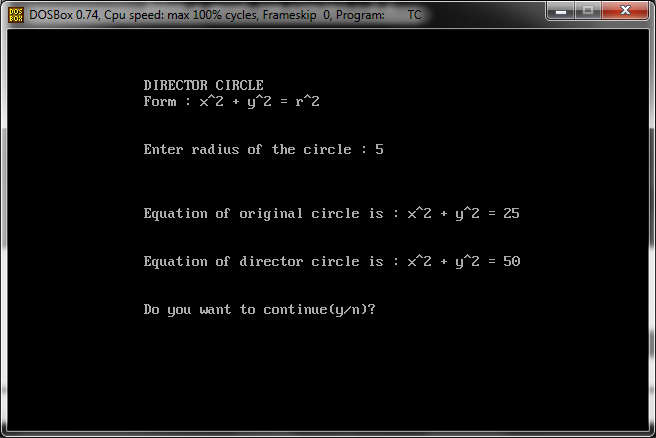
**SECTION: CIRCLE**

****

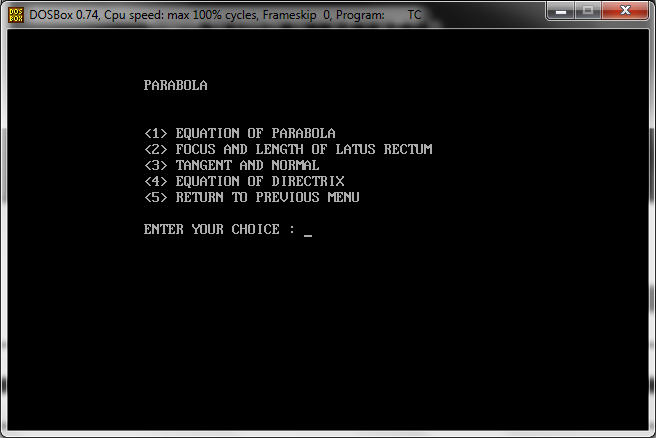
****

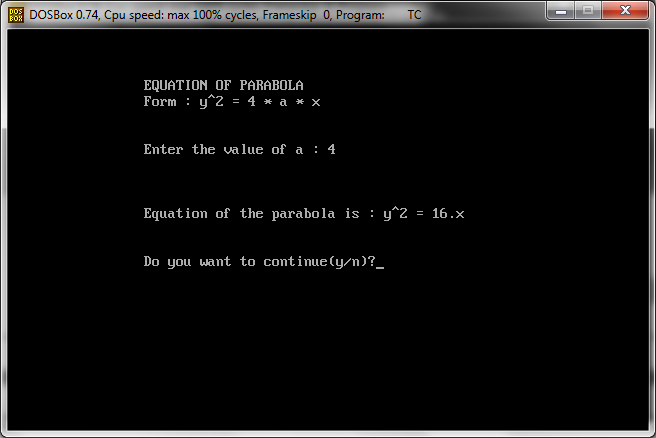
****

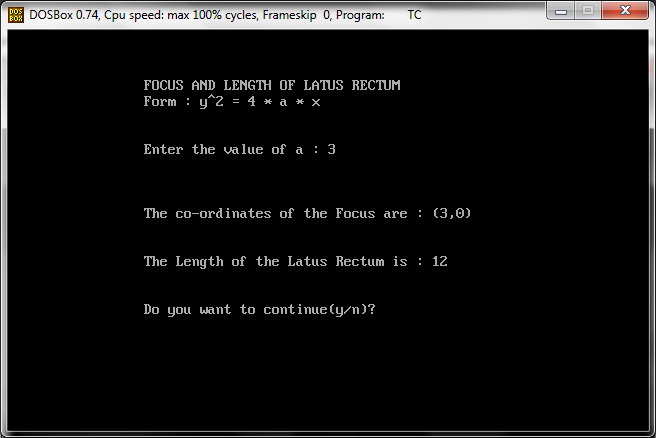
****

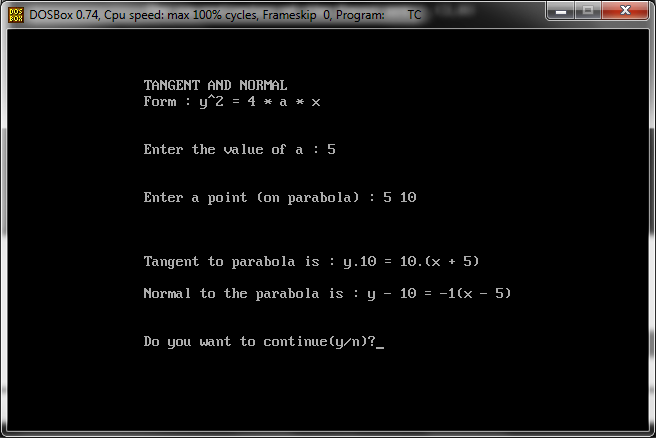
****

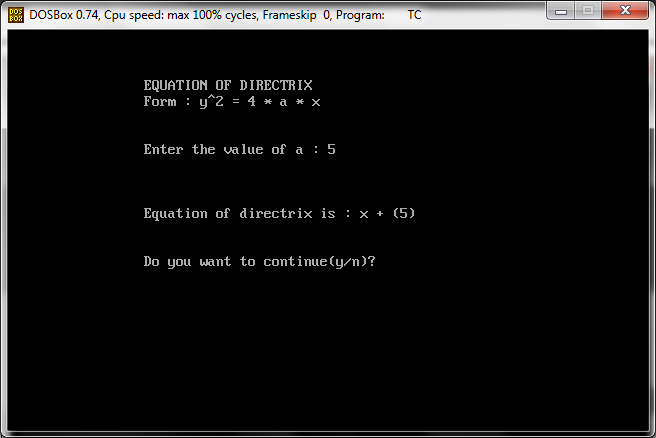
**SECTION: PARABOLA**

****

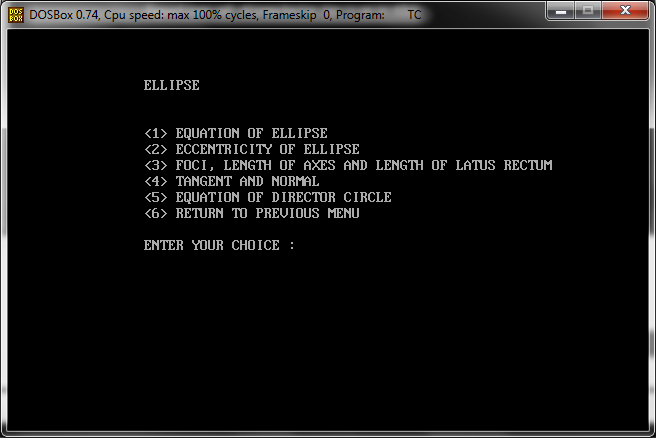
****

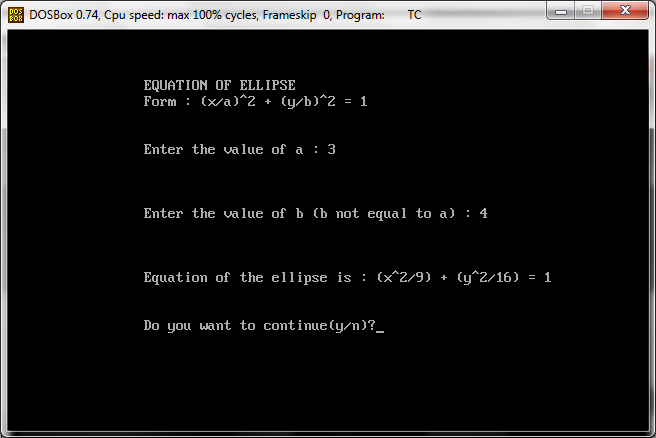
****

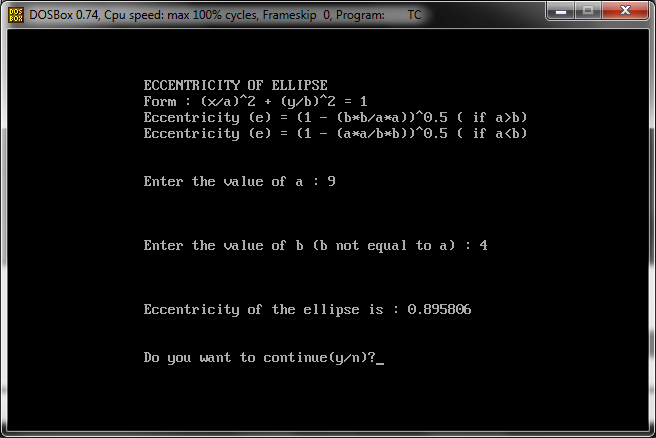
****

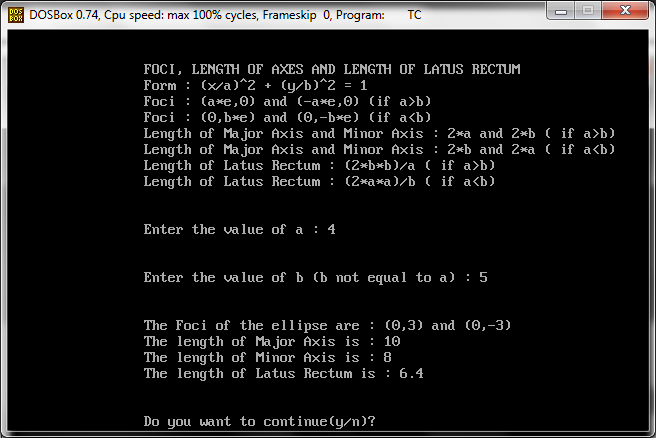
****

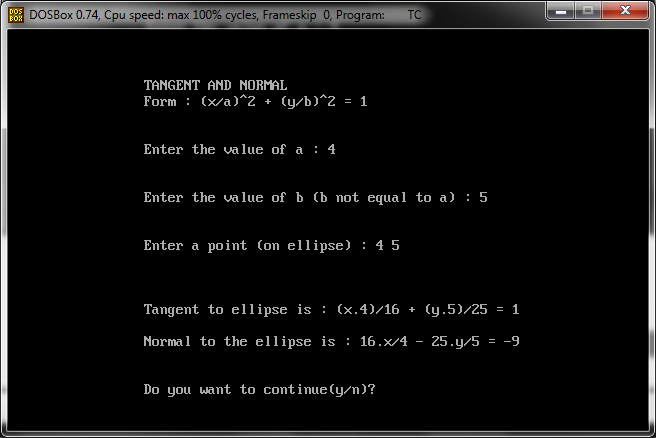
**SECTION: ELLIPSE**

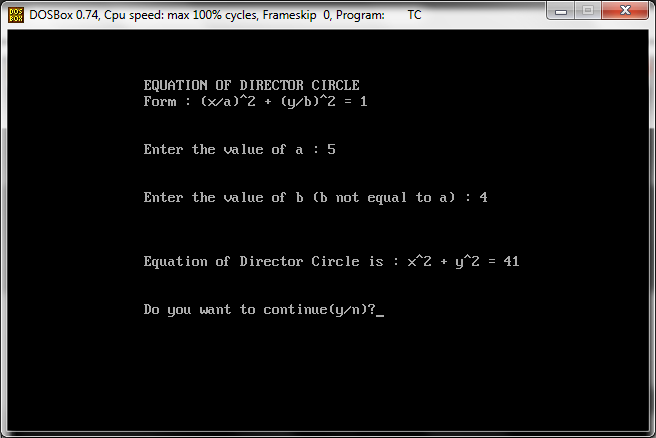
****

****

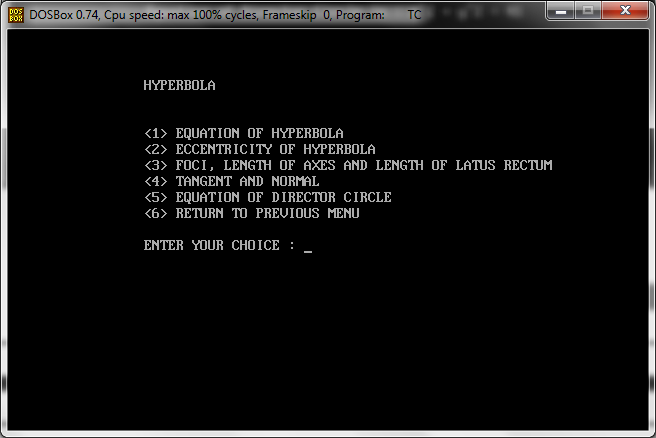
****

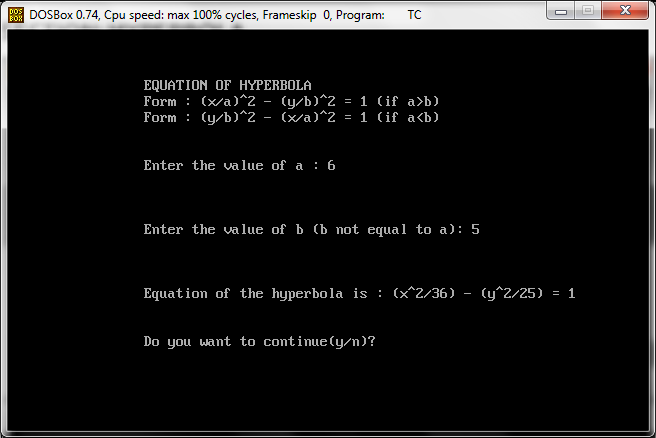
****

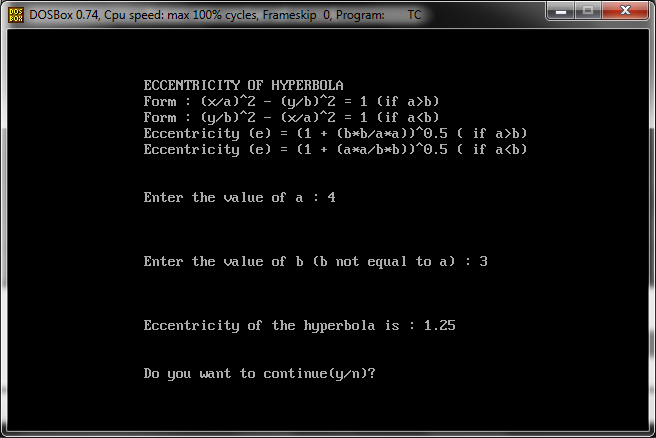
****

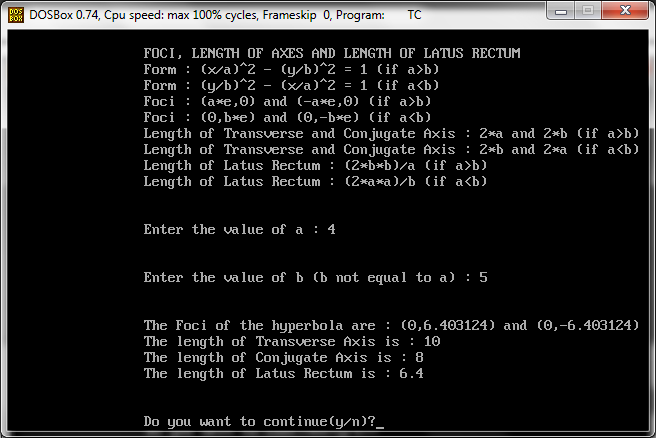
****

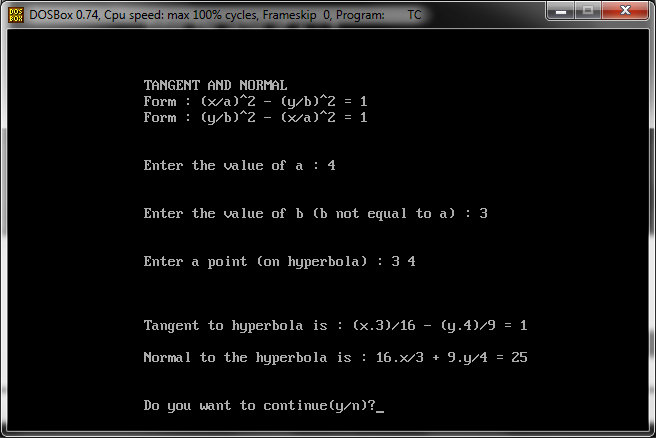
**SECTION: HYPERBOLA**

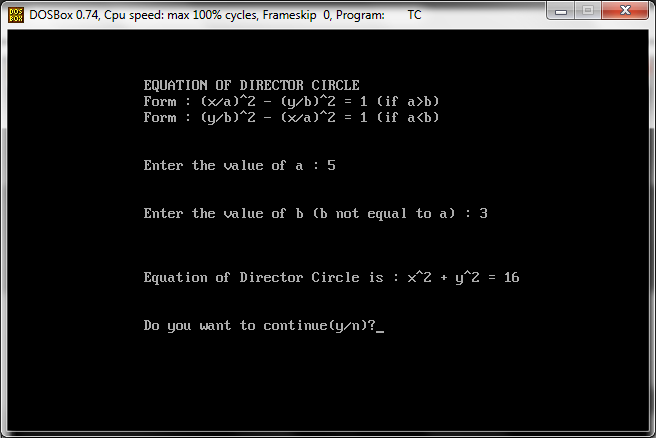
****

****

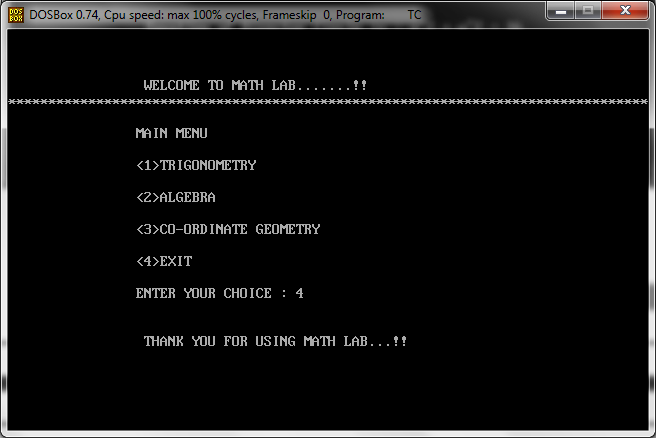
****

****

****

****

**END SCREEN**

****

**THANK YOU**