

Year

Language

1960	ALGOL - 60	(Cambridge University)
1962	CPL	
1967	B CPL	(Martin Richards)
1969	B	(Ken Thompson) - (Bell Labs)
1972	C	(Dennis Ritchie)
1989	ANSI - C / C - 89	
1990	ISC / C - 90	
1995	ISC + / C - 95	
1999	C - 99	
2011	C - 11	

Application Name - Turbo C++

Compile - Alt + F9

Run - Ctrl + F9

File Name . extension → File Name . CPP.

1- Documentation Section -

`// Text`

(Single line comment)

`/*
..... */`

(Multiline comment)

2- Link Section -

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

3- Definition Section -

`# define MAX 40``# define NAME "Alex"`

4- Global Variable Section -

`# define A 50`5- ~~Def~~ Main () -

{ Declaration Part

Executable Part }

Print f ("C Language") (in main)

6- Sub Program Section -

function - 1 -----

function - 2 -----

function - 3 -----

Example -

```
# include < stdio.h >
# include < conio.h >
main ()
{ clrscr ();
  printf ("Hello world");
  getch (); }
```

Data - Type name Variable name

int a; a
 (in memory) Value

Data type

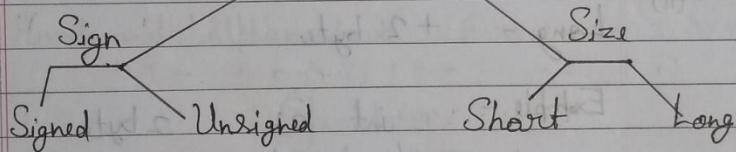
Primary Secondary

- | | |
|-------------------------|------------------|
| (i)- Integer (int) | (ii)- Array |
| (iii)- Character (char) | (iii)- Structure |
| (iv)- Float (float) | (iv)- Pointer |
| (v)- Double (double) | (v)- Enum |
| (vi)- Void (void) | (vi)- etc. |

Primary Data Type -

- 1- Integer - (int) (0-9) 2 byte (any number like as 1, 2, 11, 101, 200 etc)
- 2- Character - (char) (a-z, A-Z) 1 byte (any one letter like as a, b, c, A, Z etc)
- 3- Float - (float) (Decimal value) 4 byte (Small Decimal value like as 2.3, 2.034 etc)
- 4- Double - (double) (Decimal value) 8 byte (Long Decimal value like as 2.3212, 2.46793 etc)
- 5- Void - (void) (Empty data type) (Made for function)

Modifiers



Data type name Variable name.

Char a;

Float C;

double Abc;

int number = 40;

a = 'B';

C = 2.5;

Abc = 4.5542;

a - Sign - Use for integer and character

(i) Signed - Positive / Negative.

(ii) Unsigned - Positive

Example -

int x = 40;

signed int x;

unsigned int x;

Number

211

350

480

etc.

b - Size - Not allow in float and character

Number - (0 -

(i) Short - Same byte

Octal - (0 -

(ii) Long - + 2 byte

Hexadecimal -

Example - int a; 2 byte

short int a; 2 byte

long int a; 4 byte

double x; 8 byte

short double x; 8 byte

long double x; 10 byte

1- Arithmetic

(i) + Add

(ii) - Sub

(iii) * Mul

(iv) % Mod

(v) / Div

Variable -

int a;

a

a = 50;

50

a = 40;

40

a = 100;

100

printf(a);

100 (Show on screen)

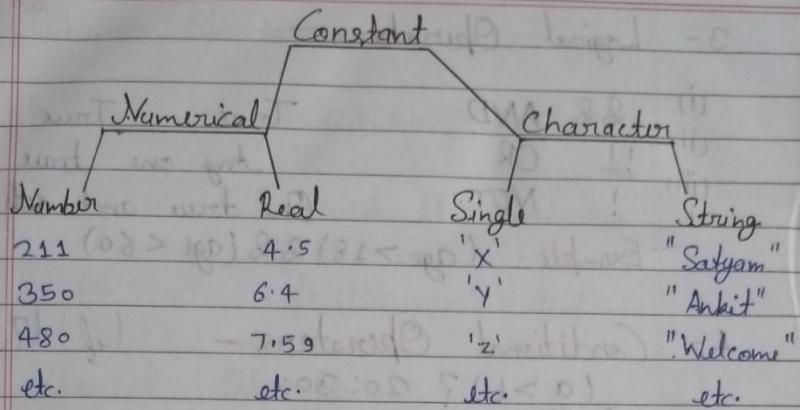
2- Relational

(i) < Less

(ii) > Great

(iii) >= Greater

(iv) == equal



Number - (0-9) like as 0 - 98, 108, 5979 etc.

Octal - (0-9) like as 0 - 06, 0769, 09769 etc.

Hexadecimal - Alphanumeric like as 0 - 0x23, 0x4578 etc.

Operators

1- Arithmetic Operators (+, -, *, %, /) -

- (i) + Addition
- (ii) - Subtract
- (iii) * Multiply
- (iv) % Modulus
- (v) / Divide

2- Relational Operators (>, <, >=, <=, !=) -

- (i) < Less than (iv) <= Less than or equal
- (ii) > Greater than (v) != Not equal to
- (iii) >= Greater than or equal Example - (5 < 8), (5 > 6) etc.
- (vi) = equal to

3- Logical Operators -

(i) $\&$ AND

True True

(ii) \sqcup OR

Any one true

(iii) ! NOT

One true and other false

Example - $(age > 18) \& (age < 60)$

4- Conditional Operators - (if) -(?)

 $(a > b) ? 20 : 30;$

Condition True or False

5- Assignment Operators -

 $a = 10, x = 50, b = 30.$

6- Increment / Decrement (++, --) -

(i) Pre $(++a, --a)$ - $a = 100$ $b = ++a *$ $b = 101$ $a = 101$

(ii)

Post $(a++, a--)$ + $a = 100$ $b = a++$ $b = 100$ $a = 101$ $b = 101$ $b = 100$ $a = 101$ $b = 101$ $a = 101$ $b = 100$ $a = 101$ $b = 101$

7- Bitwise

(i) &

(ii) :

(iii) ^

(iv) ^

(v) >

(vi) <

8- Spec

(i) , - , >

(ii) Size

5- Ass

Ty

Rea

Sca

j

Ch

St

C

7- Bit wise Operators -

- (i) & And
- (ii) : OR
- (iii) ^ Exclusive OR
- (iv) >> Right
- (v) << Left

8- Special Operator - (', sizeof)

(i) :- $x = (a = 20, b = 10, a + b)$

$$\xrightarrow{20} \xrightarrow{10} \xrightarrow{20+10=30}$$

$$x = 30$$

(ii) Size of -

Size of (x) 2 bytes

5- Assignment Operators -

Type variable = Value ;

Name

int a = 10;

Reading data from keyboard -

Scanf ("control string", &value1, &value2, ...)

int = %d	}	control string
Char = %c		
String = %s		
float = %f		

Example -

(i) #include <stdio.h>
#include <conio.h>
int main()
{ int number = 10;
printf("number");
return 0; }

(ii) #include <stdio.h>
#include <conio.h>
int main()
{ int number;
clrscr();
printf("Enter an integer: ");
scanf("%d", &number);
printf("you entered: %.d", number);
return 0; }

(iii) #include <stdio.h>
#include <conio.h>
int main()
{ int firstNumber, secondNumber, sumOfTwoNumbers;
printf("%d %d", &firstNumber + &secondNumber);
printf(" Enter two integers: ");
scanf("%d %d", &firstNumber + &secondNumber);
sumOfTwoNumbers = firstNumber + secondNumber;
printf("%d + %d = %d", firstNumber, secondNumber, sumOfTwoNumbers);
return 0; }

(iv) #include <stdio.h>
#include <conio.h>
main()
{

(v) #include <stdio.h>
#include <conio.h>
main()
{

(vi) #include <stdio.h>
#include <conio.h>
main()
{

(iv) # include <stdio.h>
include <conio.h>
main()
{ char ch;
printf("Enter a character: ");
scanf("%c", &ch);
printf("ASCII value of %c is %d", ch, ch);
getch();
}

(v) # include <stdio.h>
include <conio.h>
main()
{
char ch, d;
printf("Enter Two Characters: ");
scanf("%c%c", &ch, &d);
printf("ASCII value of %c is %d", ch, ch);
printf("ASCII value of %c is %d", d, d);
getch();
}

(vi) # include <stdio.h> #
include <conio.h> #
main()
{ clrscr();
for (int i = 0; i <= 255; i++)
{ printf("ASCII value of %c is %d \n", i, i);
}
getch();
}

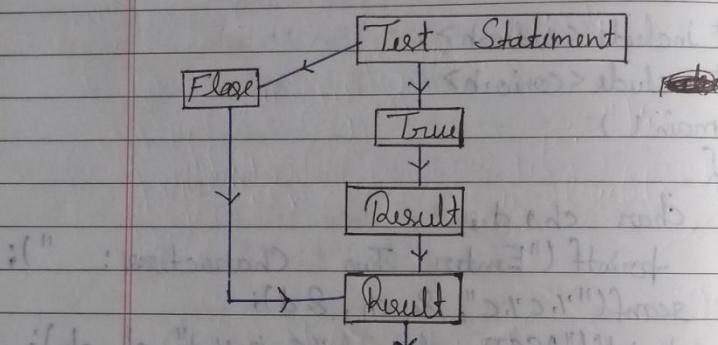
Decision Making & Branching -

(i) If statement

(ii) Switch statement

(iii) Conditional statement

(iv) Goto statement



(i) If statement

Simple if -

Example-(a) #include <stdio.h>

#include <Conio.h>

main()

```

int a;
printf("Enter a number");
scanf("%d", &a);
if (a % 2 == 0)
  
```

```
{  
    int a; // declaration  
    printf("%d is a even number\n", a);  
}  
}  
}
```

(b)

```
#include <stdio.h>  
#include <conio.h>  
main()  
{  
    int a; // declaration  
    printf("Enter a Number");  
    scanf("%d", &a);  
    if (a % 2 == 0)  
    {  
        if (a > b) : // condition  
            printf("%d is a even number\n", a);  
    }  
    else  
    {  
        if (a > b) : // condition  
            printf("%d is a odd number\n", a);  
    }  
    getch();  
}
```

(c)

```
#include <stdio.h>
#include <conio.h>
main()
{
    int a, b;
    printf("\nEnter Two Numbers");
    scanf("%d %d", &a, &b);
    if (a == b)
        printf("Result : %d = %d", a, b);
    else if ((a > b))
        printf("Result : %d > %d", a, b);
    else
        printf("Result : %d < %d", a, b);
    getch();
}
```

(d)

(d)

```
#include <stdio.h>
#include <conio.h>
main()
{
    int a, b, c;
    clrscr();
    printf("Enter Three Numbers");
    scanf("%d %d %d", &a, &b, &c);
    if (a > b)
    {
        if (a > c)
            printf("a is largest number");
        else
            printf("c is largest number");
    }
    else
    {
        if (b > c)
            printf("b is largest number");
        else
            printf("c is largest number");
    }
    getch();
}
```

(ii) Laddan Else if

```

if (condition)
{
    Block 1;
}

else if (condition)
{
    Block 2;
}

else if (condition)
{
    Block 3;
}

else
{
    Default Block;
}

```

Example -

```

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

main()
{
    clrscr();
    int a;
    clrscr();
    printf("\n \t [1] Addition");
    printf("\n \t [2] Subtract");
    printf("\n \t [3] Multiply");
    printf("\n \t [4] Divide");
    printf("\n Choose your Option: ");
}

```

```
scanf ("%d", &a);  
{  
    if (a == 1)  
        printf ("\n Addition");  
    else if (a == 2)  
        printf ("\n Subtract");  
    else if (a == 3)  
        printf ("\n Multiply");  
    else if (a == 4)  
        printf ("\n Divide");  
    else  
        printf ("\n Invalid Option");  
}  
getch();
```

3- Conditional Statement -

Condition ? Expression 1 : Expression 2 ;
 True ↑ False) ↑

```
#include <stdio.h>
#include <conio.h>
main()
{
  clrscr();
  printf("\n Enter First Number: ");
  scanf("%d", &a);
  printf("\n Enter Second Number: ");
  scanf("%d", &b);
  c = (a > b) ? a : b;
  printf("\n c = %d", c);
  getch();
}
```

ForLoop

While

Do while

1- For Loop -

#

#

main

{

int

for

{

: j

}

getch()

2- While -

#

#

main

{

int i

while

{

j = 0

i++

}

getch();

}

1- For Loop -

```
# include <stdio.h>
# include <conio.h>
main()
{
    int i;
    for (i = 0; i < 10; i++)
    {
        printf("%d\n", i);
    }
    getch();
}
```

2- While -

```
# include <stdio.h>
# include <conio.h>
main()
{
    int i;
    while (i < 10)
    {
        printf("%d\n", i);
        i++;
    }
    getch();
}
```

3- Do while -

```
# include <stdio.h>
# include <conio.h>
main()
{
    int i = 1;
    do
    {
        printf("%d \n", i);
        i++;
    }
    while (i <= 4 && i >= 2);
    getch();
    return 0;
}
```

Example -

Control case statement -

Switch case -

Switch (expression).

```
{  
    case label 1: statement  
    break;  
    case label 2: statement  
    break;  
    case label 3: statement  
    break;  
    default: statement  
}
```

Example- #include <stdio.h>
#include <conio.h>

main ()

{

int b=3;

switch(b)

{

case 1:

printf("value is 1 \n");

break;

case 2:

printf("value is 2 \n");

break;

case 3:

printf("value is 3 \n");

break;

default:

printf("value is other than 1,2,3 \n");

}

getch();

return 0;

}

Break - nite > billion +

<nl> #include <stdio.h>

#include <conio.h>

main()

{

int i;

for(i=0; i<10; i++)

{

if (i == 5)

{

printf("Coming out of for loop when:");

break;

}

printf("%d \n", i);

{}

getch();

}

Continue -

#include <stdio.h>

#include <conio.h>

main()

{

int j;

clrscr();

for(i=0; i<10; i++)

{

if (i == 5 || j == 6)

```
{ <stdio.h> included
printf("skipping %d from for loop\n", i);
};

continues;
}

printf("%d\n", i);
getch();
}

when i=5"
:(" Size -> rework refactored
# include < stdio.h >
# include < conio.h >
int main()
{
    : worked { rework refactored
        printf("Storage Size for int: %d\n",
               sizeof(int)); // size of int
        getch(); // rework refactored
        return 0;
    }
}

:(" Adig13 web. work" I Heired
{
    :(" Adig13 web. work" I Heired
}
```

```
# include < stdio.h >
# include < conio.h >
main()
{
    char name;
    int age;
    char city;
    int income;
    printf("\nEnter Your Name: ");
    scanf("%c", name);
    printf("\nEnter Your Age: ");
    scanf("%d", age);
    printf("\nEnter Your City: ");
    scanf("%c", city);
    printf("\nEnter Your Income: ");
    scanf("%d", income);
    printf("%c Name: ", name);
    printf("%d Age: ", age);
    printf("%c City: ", city);
    printf("%d Income: ", income);
    if ((age < 20) && (income < 5000))
    {
        printf("You are Eligible");
    }
    else
    {
        printf("You are Not Eligible");
    }
    getch();
}
```

:(1-D) One Dimensional Array

One Dimensional
array

Two Dimensional
array

Multi Dimensional
Array

1- One Dimensional Array —

Structure - Data Type array name [array size]

```
int sub[5]
```

```
sub[0] = 20
```

```
sub[1] = 30
```

```
sub[2] = 35
```

```
sub[3] = 24
```

```
sub[4] = 15
```

In memory -

```
: [ ] + sub[5]
```

Sub[0]	20	{
Sub[1]	30	o = i root
Sub[2]	35	}
Sub[3]	24	3 times
Sub[4]	15	{

Example-1 - #include < stdio.h >

#include < conio.h >

main()

{

int sub[5];

for (i=0; i<8; i++)

{

scanf("%d %d", &sub[i]);

```

        printf("\n Number: %d\n", sub[i]);
    }
    getch();
}

```

2- Two D

Syntax

In mem

Example 2- #include <stdio.h>

[size memory] #include <conio.h>

main()

{

int sub[5];, total=0;

printf("Enter 5 Numbers: \n");

for(i=0; i<5, i++)

{

scanf("%d", & sub[i]);

total = total + sub[i];

}

for(i=0; i<5; i++)

{

printf("\n Number: %d", i+1, sub[i]);

printf("\n Total Marks ie %d", total);

getch();

return 0;

}

Examp

{

in

fa

{

get

{

get

{

get

{

if(i>duo, "N b. n")

2- Two Dimensional Array

Syntax int sub[2][3]

~~Row[0]~~

2- Row

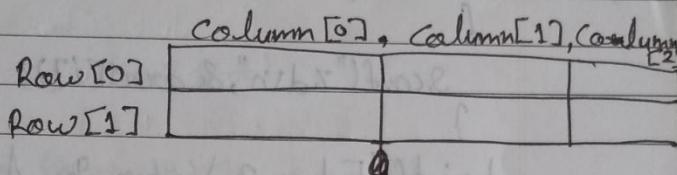
3- Column

Row[0] = column[0], column[1], column[2]

Row[1] = column[0], column[1], column[2]

In memory -

sub[2][3]



Example-1- #include <stdio.h>

#include <conio.h>

main()

{

int sub[2][3], i, j;

for(i=0; i<2; i++)

{

for(j=0; j<3; j++)

{

scanf("%d\n", &sub[i][j]);

printf("in Number %d \n", sub[i][j]);

};

}

getch();

}

Example - # include < stdio.h >

include < conio.h >
main ()

{
int arr1[3], arr2[3], arr3[3], i, j, k, a;
printf("Enter 3 Numbers In Array1 \n");
for (j = 0; j < 3; j++)

{
scanf("%d\n", &arr1[j]);
}

printf("Enter 3 Number In Array2 \n");
for (j = 0; j < 3; j++)

{
scanf("%d\n", &arr2[j]);
}

printf("Total Of Array1 And Array2 Is \n");
for (k = 0; k < 3; k++)

{
arr3[k] = arr1[k] + arr2[k];
}

for (a = 0; a < 3; a++)

{
printf("Array3 %d : %d \n", a, arr3[a]);
}

getch();
}

3.-

3 - Multi Dimensional Array -

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

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

```
main()
```

```
{
```

```
int sub[2][3][4], i, j, k;
```

```
for(i=0; i<2; i++)
```

```
{
```

```
for(j=0; j<3; j++)
```

```
{
```

```
for(k=0; k<4; k++)
```

```
{
```

```
scanf("%d\n", &sub[i][j][k]);
```

```
} printf("\nNumber %d\n", sub[i][j][k]);
```

```
}
```

```
getch();
```

```
return 0;
```

```
}
```

```
);
```

Function

Built-In Function

printf();
scanf();
getch();
CloseConsole();

User Define Function

addition();
printname(char a)
{
 // code
}

Return type

Function name (Parameters)

Elements of User Define Function

1- Function Declaration - Type

Function Name

Parameter list / Argument

Termination semicolon.

2- Function Call

3- Function Definition - Type

Function Header

Function Name

Parameter list / Argument

Function Body

Local variable declaration

Statements

Return value

Example -

```
# include <stdio.h>
# include <conio.h>
int addition ( int a, int b); (Function Declaration)
```

```
main ()
```

```
{
```

```
    int d=0, a=6, b=5;
```

```
    d = addition (a, b); (Function Call)
```

```
    printf ("\n %d", d); Actual Argument
```

```
    getch();
```

```
    return 0;
```

```
}
```

```
int addition ( int a, int b) (Function Definition)
```

```
{
```

```
    int c=0;
```

```
    c = a + b;
```

```
    return c; (Return value)
```

```
}
```

Arguments

Formal

Actual

Return Type

Return value

Return

Type of Function —

1- Function without return type and without argument -

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

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

```
void display(void);
```

```
main()
```

```
{
```

```
display();
```

```
getch();
```

```
}
```

```
void display(void)
```

```
{
```

```
int a, b, c;
```

```
printf("\nEnter First Number: ");
```

```
scanf("%d", &a);
```

```
printf("\nEnter Second Number: ");
```

```
scanf("%d", &b);
```

```
c = a + b;
```

```
printf("\nAddition is %d", c);
```

```
}
```

2- Function

```
with
```

```
#include
```

```
#include
```

```
void
```

```
main()
```

```
{
```

```
close
```

```
int
```

```
print
```

```
scanf
```

```
print
```

```
scanf
```

```
display
```

```
getch
```

```
}
```

```
void
```

```
{
```

```
int
```

```
c =
```

```
prin
```

```
}
```

2- Function with out return type and with argument -

```
#include <stdio.h>
#include <conio.h>
void display(int, int);
main()
{
    clrscr();
    int a, b;
    printf("\nEnter First Number: ");
    scanf("%d", &a);
    printf("\nEnter Second Number: ");
    scanf("%d", &b);
    display(a, b);
    getch();
}

void display(int x, int y)
{
    int c = 0;
    c = x + y;
    printf("\nAddition is %d", c);
```

3- Function with return type and without argument-

```
#include <stdio.h>
#include <conio.h>
int display (void);
main()
{
    int c,
    c= display ();
    printf("\nAddition is %d", c);
    getch();
}
int display (void)
{
    int a, b;
    printf("\nEnter First Number: ");
    scanf("%d", &a);
    printf("\nEnter Second Number: ");
    scanf("%d", &b);
    return (a+b);
}
```

4 -

4 - Function with return type and with argument-

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

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

```
int add Multiply (int a, int b);
```

```
main ()
```

```
{
```

```
int d = 0, a = 6, b = 5;
```

```
d = Multiply (a, b);
```

```
printf ("\n%d", d);
```

```
getch ();
```

```
return 0;
```

```
}
```

```
int multiply (int x, int y)
```

```
{
```

```
int c = 0;
```

```
c = x * y;
```

```
return c;
```

```
}
```

Function Call

Call By Value

Call By Reference

Nested Function -

Example -

```
#include <stdio.h>
#include <conio.h>
void add(int, int);
void sub(int);
void multi(int, int);
main()
{
    int x, y;
    clrscr();
    printf("\nPlease Enter First Number: ");
    scanf("%d", &x);
    printf("\nPlease Enter Second Number: ");
    scanf("%d", &y);
    add(x, y);
    getch();
}
```

```
void add(int a, int b):
{
```

```
    int c = 0;
    c = a + b;
    printf("\nAddition is %.d", c);
    multi(a, b);
}
```

```
void multi(int p, int q)
{
```

```
    int r = 0;
    r = p * q;
```

```
printf("\nMultiplication Is %.d", a);
sq1(b);
sq1(c);
}
void sq1(int i)
{
    int j=0
    j = i*i;
    printf("\n Square Of %.d Is %.d", i, j);
}
```

Example 2- (Remove Multiplication only)-

```
#include <stdio.h>
#include <conio.h>
void add(int, int);
void sq1(int);
void multi(int, int);
main()
{
    int x, y;
    clrscr();
    printf("\nPlease Enter First Number: ");
    scanf("%d", &x);
    printf("\nPlease Enter Second Number: ");
    scanf("%d", &y);
    add(x, y);
    getch();
}

void add(int a, int b)
```

```
{ "b": "P", "c": "B", "d": "B", "e": "B", "f": "B", "g": "B", "h": "B", "i": "B", "j": "B", "k": "B", "l": "B", "m": "B", "n": "B", "o": "B", "p": "B", "q": "B", "r": "B", "s": "B", "t": "B", "u": "B", "v": "B", "w": "B", "x": "B", "y": "B", "z": "B"}  
int c=0;  
c = a+b;  
printf("Addition %d %.d", c);  
// multi (a,b);  
squr(a);  
squr(b);  
}
```

```
void multi (int p, int q)  
{
```

```
int r=0;  
r = p * q;  
printf("Multiplication %d %.d", r);  
squr(p);  
squr(q);  
}
```

```
void squr(int).  
{
```

```
int j=0;  
j = i * i;  
printf("Square Of %.d %.d", i, j);  
}
```

Recursion Function -

```
#include < stdio.h >
#include < conio.h >
int sum(int num);
int main()
{
    int num, ans;
    printf("Enter A Number: ");
    scanf("%d", &num);
    printf("\n Sum = %d", ans);
    getch();
    return 0;
}

int sum(int num);
{
    if (num != 0)
    {
        return num + sum(num - 1);
    }
    else
    {
        return num;
    }
}
```

Address -

```
Example-1- #include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    int a;
    printf("\nEnter A Number: ");
    scanf("%d", &a);
    printf("\nNumber Is %d", a);
    printf("\nAddress Of Number Is %u", &a);
    getch();
    return 0;
}
```

Example-2- #include <stdio.h>

```
#include <conio.h>
main()
{
    clrscr();
    int a, b;
    printf("\nEnter Two Numbers: ");
    scanf("%d %d", &a, &b);
    printf("Numbers are %d %d", a, b);
    printf("\nAddress Of Numbers are %u %u", &a, &b);
    getch();
    return 0;
}
```

Example -3- #include <stdio.h>

#include <conio.h>

main()

{

int a, b;

printf("\nEnter A Number: ");

scanf("%d", &a);

b = a;

printf("Numbers are %d %d", a, b);

printf("\nAddress of Numbers are %u %u", &a, &b);

getch();

return 0;

}

Example -4- # include <stdio.h>

include <conio.h>

main()

{

int a, b, c;

printf("\nEnter A Number: ");

scanf("%d", &a);

b = a;

c = a;

printf("Numbers are %d %d %d", a, b, c);

printf("\nAddress of Numbers are %u %u %u", &a, &b, &c);

getch();

return 0;

}

Pointer -

Syntax - Data type * variable Name ;

Example-2 # include < stdio.h >

include < conio.h >

int main()

{

clrscr();

int a = 10;

int * b;

b = &a;

printf("Value of A is %d", a);

printf("Value of *b is %d", b);

getch();

return 0;

Example-2 # include < stdio.h >

include < conio.h >

int main()

{

clrscr();

int a = 10;

int * b;

b = &a;

printf("Value of A is %d", a);

printf("Value of *B is %d", b);

getch();

return 0;

Example - 3- # include < stdio.h >

```
# include < conio.h > → int main()
{
    int a = 10;
    int * b;
    b = &a;
    printf("Value of A is %d", a);
    printf("Value of *B is %p", b);
    getch();
    return 0;
}
```

Example - 4- # include < stdio.h >

```
# include < conio.h >
int main()
{
    clrscr();
    int a = 10;
    int * b;
    b = &a;
    printf("Value of A is %d", a);
    printf("Value of *B is %x", b); → hex x
    getch();
    return 0;
}
```

%d - Address In Number

%p - Address In Alfabat and Number

%X - Address In A and N Capital letter.

%x - Address In A and N small letter.

Example - 5- #include <stdio.h>
#include <conio.h>
int main()
{
 clrscr();
 int a = 100;
 int *b = &a;
 int sum;
 printf("Value of A %d : %d", a, *b);
 printf("\nValue of A %d : %d", &a, a);
 printf("\nValue of B %d : %d", b, *b);
 printf("\nValue of B %d : %d", *b, b);
 getch();
 return 0;
}

Example - 6- #include <stdio.h>

#include <conio.h>
int main()
{
 int a = 100;
 int *b = &a;
 int *c = b;
 printf("Value of A %d : %d", a, *b);
 printf("\nValue of A %d : %d", &a, a);
 printf("\nValue of B %d : %d", (*b + 100), *b);
 printf("\nValue of B %d : %d", (*b + 100), *c);
 printf("\nValue of B %d : %d", *b, *b);
 printf("\n %d + %d = %d", a, *b, (a + *b));
 printf("\n %d", *(&a));
}

```
printf("n %u", *c);
```

```
getch();
```

```
return 0;
```

```
}
```

Null Pointer -

Example - # include < stdio.h >

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

```
int main ()
```

```
{
```

```
int a = 100;
```

```
int * b = NULL;
```

```
printf("Value of B is : %p", b);
```

```
getch();
```

```
return 0;
```

```
}
```

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

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

```
main ()
```

```
{
```

```
int a[3] = {10, 20, 30};
```

```
int * b[3]; i;
```

```
for (i = 0; i < 3; i++)
```

```
{
```

```
printf("Number %d : %d \n", i, a[i]);
```

```
b[i] = &a[i];
```

```
}
```

```
for (i = 0; i < 3; i++)
```

```
{  
    printf("\n Pointer Number %d: %d\n", i, *a[i]);  
    printf("\n Address Of A %d: %p, &i, a[i]);  
}  
getch();  
}
```

Pointer Of Pointer.

```
#include <stdio.h>  
#include <conio.h>  
int main()  
{  
    clrscr();  
    int a = 10;  
    int *b = &a;  
    int **c = &b;  
    printf("Value Of A %d: %d\n", a);  
    printf("Address Of A %d: %u, &a);  
    printf("\n Value Of *B %d: %d\n", *b);  
    printf("\n Address Of *B %d: %u, &b);  
    printf("\n Value Of **C %d: %d\n", **c);  
    printf("\n Address Of ***C %d: %u, c);  
    getch();  
    return 0;  
}
```

```
#include <stdio.h>
#include <conio.h>
int main()
{
    clrscr();
    char char1[5];
    int i;
    for(i=0; i<5; i++)
    {
        scanf("\n %c", &char1[5]);
        printf("\n Value of char at %d is %c", i, char1[i]);
        printf("\n Address of char at %d is %u", i, &char1[i]);
    }
    getch();
}
```

Passing Pointer To Function

```
#include <stdio.h>
#include <conio.h>
void swap(int*, int*);
main()
{
    int a, b;
    printf("\nEnter Number For a: ");
    scanf("%d", &a);
    printf("\nEnter Number For b: ");
    scanf("%d", &b);
    printf("\nValue Of A Is: %d \n Value Of B Is: %d", a, b);
    swap(&a, &b);
    printf("\nValue Of A Is: %d \n Value Of B Is: %d", a, b);
    getch();
    return 0;
}
```

```
void swap(int* x, int*y)
{
```

```
    int z;
    *x = *y;
    *y = z;
```

S
Systre ch
in memory

Exa

Exa

wh

String -

Syntax char a[5] = "ANKIT";
In Memory -

A	N	K	I	T	\0
0	1	2	3	4	5

NULL Character

Example-1 #include <stdio.h>

#include <conio.h>

clrscr();

char name[10];

printf("Enter Your Name: ");

~~scanf("%s", name);~~

printf("\nYour Name Is: %s", name);

getch();

Of B9e

Of B9e

Example-2 #include <stdio.h>

#include <conio.h>

clrscr();

char name[10];

printf("Enter Your Name: ");

scanf("%s", name);

printf("\nYour Name Is: %s", name);

getch();

while (name[i] != '\0') {

printf("Letter %c Is Turned %c", name[i], name[i]);

```
j++;
```

Structure -

```
struct student
```

```
{
```

```
int roll;
```

```
char name;
```

```
} stu = {101, "ANKIT"};
```

```
stu.roll → 101
```

```
stu.name → ANKIT
```

Example - #include < stdio.h >

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

```
struct student
```

```
{
```

```
int roll;
```

```
char name[10];
```

```
} stu1 = {101, "Mukesh"};
```

```
main()
```

```
{
```

```
struct student stu2;
```

```
printf("Roll No. Of First Student: ");
```

```
scanf("%d", &stu1.roll);
```

```
printf("Enter Name of First Student: ");
```

```
scanf("%s", stu1.name);
```

```
printf("Enter Roll No. of Second Student: ");
```

```
scanf("%d", &stu2.roll);
```

```
printf("Enter Name of Second Student: ");
```

```
scanf("%s", stu2.name);
```

```
printf("\nRoll No. Of Second Student %d", stu2.  
roll);  
printf("\nName Of Second Student : %.5s", stu2.name);  
getch();  
}
```

Array Within Structure —

```
#include <stdio.h>  
#include <conio.h>  
struct struct student  
  
int roll;  
char name[10];  
int mark[5];  
};  
main()  
{  
    struct student stu1;  
    int i, sum=0;  
    printf("\nEnter Roll No. of First Student: ");  
    scanf("%d", &stu1.roll);  
    printf("\nEnter Name of First student: ");  
    scanf("%s", stu1.name);  
    printf("\nEnter Mark Marks in 5 Subject: ");  
    for(i=0; i<5; i++)  
    {  
        scanf("%d", &stu1.mark[i]);  
        sum = sum + stu1.mark[i];  
    }
```

```
printf("\nRoll No. Of Student : %d", stu1.roll);
printf("\nName of Student : %s", stu1.name);
for(i=0; i<5; i++)
{
    printf("\nMark of subject %d : %d", i+1, stu1.mark[i]);
}
printf("\nTotal mark obtain in Sub %d is %d", i+1, stu1.sum);
```

```
return 0;
```

```
getch();
```

Array Of Structure —

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

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

```
struct student
```

```
{
    int roll;
    char name[10];
};
```

```
main()
```

```
{
```

```
int i;
```

```
struct student stu[5];
```

```
printf("Enter details of 5 Students: ");
for(i=0; i<5; i++)
```

```
stu[i].roll);
    name);
}
,i+1,stu,
marks[i]);
as %d from
mark[i]);
```

{

```
printf("\nEnter Roll No. Of Student %d", i+1);
scanf("%d", &stu[i].roll);
printf("\nEnter Name Of Student %d: ", i+1);
scanf("%s", stu[i].name);
}

printf("Details of Student Are: ");
for(i=0; i<5; i++)
{
    printf("\nName of student %d is %s", i+1,
           stu[i].namename);
    printf("Roll No. of student %d is %d", i+1,
           stu[i].roll);
}

getch();
```

Nested Structure -

```
#include <stdio.h>
#include <conio.h>
struct student
{
    int roll;
    char name[10];
    struct stu_mark
    {
        char sub[10];
        int m;
    } mark;
};

main()
{
    struct student stu;
    printf("Enter Roll No.: ");
    scanf("%d", &stu.roll);
    printf("\nEnter Student Name: ");
    scanf("%s", stu.name);
    printf("\nEnter Subject Code: ");
    scanf("%s", stu.mark.sub);
    printf("\nEnter Mark: ");
    scanf("%d", &stu.mark.m);
    printf("\nStudent Details: ");
    printf("Roll No.: %d", stu.roll);
    printf("Student Name: %s", stu.name);
    printf("Mark: %d", stu.mark.m);
    getch();
}
```

1- S
2- S
3- S
4- S
5- S
6- S
7- S

Syntax

String -

In Built Function -

include <string.h>

1- strlen();

2- strcat();

3- strncat();

4- strcpy();

5- strncpy();

6- strcmp();

7- strnstr();

1- strlen();

Syntax- Variable name = strlen("string");

Example -

#include < stdio.h >

#include < conio.h >

#include < string.h >

main()

{

char name[10];

int i;

printf("Please Enter A Name : ");

scanf("%s", name);

i = strlen(name);

printf("Your Name Have %d Letters", i);

getch();

}

2- Strcat ()

Syntax- strcat (Destination String, Source String);

Example-

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

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

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

```
main()
```

```
{
```

```
char str1[10];
```

```
char str2[10];
```

```
printf("Enter First String: ");
```

```
scanf("%s", str1);
```

```
printf("nEnter Second String: ");
```

```
scanf("%s", str2);
```

```
strcat(str1, str2);
```

```
printf("First String is %s", str1);
```

```
getch();
```

```
}
```

3- Strncat ()

Syntax- strncat (Destination string, Source String, Number of character);

D.S- Computing

SS- College

n = 4

computing coll.

Example-

#include

#include

#include

main()

{

char

char

pr

sc

pr

st

pri

ge

}

4- Str

Syntax- str

Examp

#inc

#inc

#inc

{

ch

ch

pr

sc

st

pri

ge

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
    char str1[10];
    char str2[10];
    printf("Enter First String:");
    scanf("%s", str1);
    printf("\nEnter Second String:");
    scanf("%s", str2);
    strcat(str1, str2, 4);
    printf("\nFirst String Is %s", str1);
    getch();
}

```

4- strcpy();

Syntax- strcpy(Destination string, Source string);

Example-

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
    char name[10];
    char temp[10];
    printf("Enter A Name:");
    scanf("%s", temp);
    strcpy(name, temp);
    printf("\nName Is: %s", name);
    getch();
}

```

5- **Strncpy()**

Syntax - **strncpy(Destination string, Source string, n);**

Example -

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

main()
{
    char name[10];
    char temp[10];
    printf("Enter A Name:");
    scanf("%s", temp);
    strncpy(name, temp, 5);
    name[5] = '\0';
    printf("\nName Is : %s", name);
    getch();
}
```

6- **strcmp()**

Syntax - **strcmp(string1, string2);**

Note - ASCII Value compare.

Example -

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

main()
{
    int i;
    char str1[10], str2[10];
    if(str1 > str2)
        printf("str1 is greater");
    else if(str1 < str2)
        printf("str2 is greater");
    else
        printf("Both are equal");
}
```

```
printf("Enter First String : ");
scanf("%s", str1);
printf("\nEnter Second String : ");
scanf("%s", str2);
i = strcmp(str1, str2);
if (i < 0)
{
    printf("String1 Is less than String2");
}
else if (i > 0)
{
    printf("String1 Is Greater Than String2");
}
else
{
    printf("String1 Equal To String2");
}
getch();
```

7- ~~Storrev()~~ ~~char strrev(char*)~~
syntax- ~~strrev(string);~~
Example- ~~henceforth~~

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
    char name[10];
    printf("Enter Name: ");
    scanf("%s", name);
    strrev(name);
    printf("Reversed Name: %s", name);
    getch();
}
```

Factorial -

```
#include < stdio.h >
#include < conio.h >
main()
{
    int number, i, fact = 1;
    printf("Enter A Number: ");
    scanf("%d", & number);
    for (i = number; i >= 1; i--)
    {
        fact = fact * i;
    }
    printf("Factorial Of %d Is: %d", number, fact);
    getch();
}
```

Palindrome —

#include < stdio.h >

#include < conio.h >

#include < string.h >

main()

{

char one[20], two[20];

int cmp;

printf("Enter A Name: ");

scanf("%s", one);

strcpy(two, one);

strrev(one);

cmp = strcmp(one, two);

if (cmp == 0)

{

printf("Entered String Is A Palindrome");

else

{

printf("Entered String Is Not A Palindrome");

} getch();

Reversing String Without Using `Strrev()` Function -

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

main()
{
    char name[10];
    int i, j;
    printf("Enter A Name : ");
    scanf("%s", name);
    i = strlen(name);
    printf("Reverse String Is : ");
    for (j = i - 1; j >= 0; j--)
    {
        printf("%c", name[j]);
    }
}
```

Union -

Syntax - Union tag-name

```
{ <data type> member1;  
  <data type> member2;
```

```
    } Union variable;
```

Example -

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

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

```
Union student
```

```
{
```

```
    int roll;
```

```
    char name [10];
```

```
};
```

```
main()
```

```
{
```

```
union student stu;
```

```
printf("Enter Student Roll No.: ");
```

```
scanf("%d", stu.roll);
```

```
printf("\nEnter Student Name: ");
```

```
scanf("%s", stu.name);
```

```
printf("\nStudent Details Are: ");
```

```
printf("\nStudent Roll No. : %d ", stu.roll);
```

```
printf("\nStudent Name : %s ", stu.name);
```

```
getch();
```

Structure	Union
<p>Stu ↑ 12 byte</p> <p>Separate Memory For all Members</p>	<p>↑ 10 byte</p> <p>Memory Sharing among All Members</p>

Typedef -

Syntax - typedef <Existing data-type> <aliasname>;

Example -

```
#include <stdio.h>
#include <conio.h>
main()
{
    typedef int number;
    number n1=100, n2=200, sum=0;
    printf("First Number: %d", n1);
    printf("\nSecond Number: %d", n2);
    sum=n1+n2;
    printf("\nAddition is %d", sum);
    getch();
}
```

main();
main();

1- `malloc()` —

`#include <stdio.h>`

`#include <conio.h>`

`#include <stdlib.h>`

`main()`

{

`int n, *ptr, sum=0, i, *p;`

`printf("Enter The Size Of Array: ");`

`scanf("%d", &n);`

~~`printf("Enter The Elements In Array: ");`~~

`ptr=(int *)malloc(n * sizeof(int));`

`p=ptr;`

`printf("n Enter The Elements In Array: ");`

`for(i=1; i<=n; i++)`

`scanf("%d", ptr);`

`sum = sum + *ptr;`

~~`ptr;`~~

}

`printf("n Array Elements are: ");`

`for(i=1; i<=n; i++)`

{

`printf("n %d", *p);`

~~`p++;`~~

}

`printf("n Addition Is %d", sum);`

`getch();`

{

Example - "b/w op with h/w" Twisted.

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

main()
{
    int n, *ptr, sum = 0, i, *p;
    printf("Number Of Elements To Be Entered: ");
    scanf("%d", &n);
    ptr = (int *)malloc(n * sizeof(int));
    p = ptr;
    if (ptr == NULL)
    {
        printf("\nEnter: Out Of Memory");
        exit(0);
    }
    printf("\nEnter The Elements: ");
    for (i = 1; i <= n; i++)
    {
        scanf("%d", ptr);
        sum = sum + *ptr;
        ptr++;
    }
    printf("\nElements Are: ");
    for (i = 1; i <= n; i++)
    {
        printf("\n%d", *p);
        p++;
    }
}
```

```

        printf("\n Addition Is %d", sum);
    } free(ptr);
    getch();
}

```

Dynamic Memory Allocation —

1- Malloc

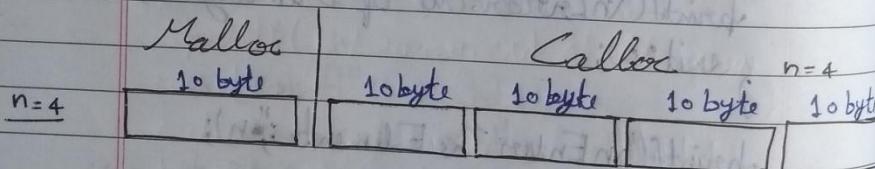
2- Free

3- Alloc

4- Realloc

For Free Memory -

Syntax - free(pointer-name);



3- Alloc —

#include < stdio.h >

#include < conio.h >

#include < stdlib.h >

main()

{

int n, *ptr, sum = 0, *p;

printf("Number Of Element To Be Entered: ");

scanf("%d", &n);

ptr = (int *) alloc(n * sizeof(int));

p = ptr;

```
if(ptr == NULL)
{
    printf("\nError: out Of Memory");
    exit(0);
}

printf("\nEnter The Elements:\n");
for(i=1; i<=n; i++)
{
    scanf("%d", &ptr);
    sum = sum + *ptr;
    ptr++;
}

printf("\nElements Are: ");
for(i=1; i<=n; i++)
{
    printf("\n%d", *p);
    p++;
}

printf("\nAddition Is %d", sum);
getch();
}
```

4- Dealloc -

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

main()
{
    clrscr();
    int n, *ptr, sum=0, i, *p, *q;
    printf("Enter Size Of Array:");
    scanf("%d", &n);
    ptr=(int *)malloc(n * sizeof(int));
    p=ptr;
    if(ptr==NULL)
    {
        printf("\nError: Out of Memory");
        exit(0);
    }
    printf("\nAllocated Memory %u", ptr);
    printf("\nEnter %d Elements:", n);
    for(i=1; i<=n; i++)
    {
        scanf("%d", ptr);
        sum = sum + *ptr;
        ptr++;
    }
    printf("\nElements Are: ");
    for(i=1, i<=n; i++)
    {
        printf("\n%d", *p);
    }
}
```

```
    } p++;  
    printf("\nAddition Is %d", sum);  
    printf("\nEnter New Size of Array: ");  
    scanf("%d", &n);  
    ptra = (int*)malloc(ptr, n * sizeof(int));  
    q = ptra;  
    if (ptr == NULL)  
    {  
        printf("\nError: Out of Memory");  
        exit(0);  
    }  
    printf("\nAllocated Memory %u", ptr);  
    printf("\nEnter %d Enter %d Elements: ", n);  
    for (i = 1; i <= n; i++)  
    {  
        scanf("%d", ptra);  
        sum = sum + *ptra;  
        ptra++;  
    }  
    printf("\nElements Are: ");  
    for (i = 1; i <= n; i++)  
    {  
        printf("\n%d", *q);  
        q++;  
    }  
    printf("\nAddition Is %d", sum);  
    getch();  
}
```

Graphics -

```
include <graphics.h>
```

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

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

```
#include <graphics.h>
```

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

```
int main()
```

```
{
```

```
int gd=DETECT, gm, err;
```

```
int graph(gd, &gm, "E:\\TURBO C\\BGI\\");
```

```
err=graphresult();
```

```
if(err!=grOk)
```

```
{
```

```
printf("Graphics Error % s", grapherrormsg);
```

```
getch();
```

```
exit(1);
```

```
}
```

```
outtext("Graphics Loaded");
```

```
getch();
```

```
clgraph();
```

```
end();
```

1- Line

2- Circle

3- Rectangle

4- Ellipse

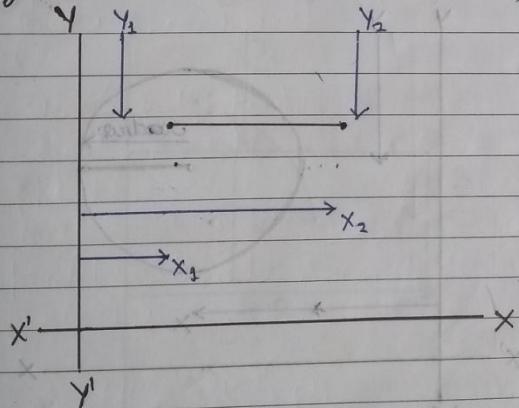
5- Filled Ellipse

<graphics.h>

2- Line -

```
#include <conio.h>
#include <graphics.h>
int main()
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "C:\TURBOC3\BGI");
    line(100, 100, 300, 100);
    getch();
    closegraph();
    return 0;
}
```

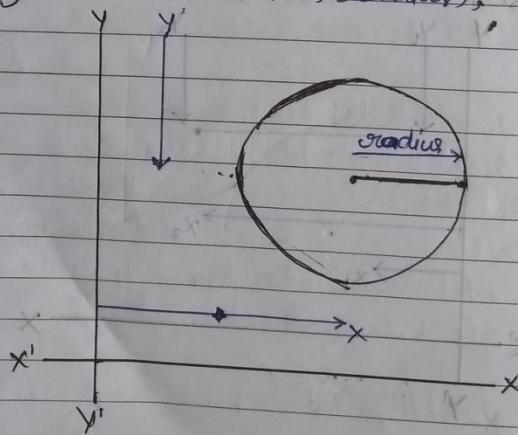
overcome(ero);

Syntax - line (x_1, y_1, x_2, y_2);

2- Circle -

```
#include <conio.h>
#include <graphics.h>
int main()
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI\\");
    circle(200, 150, 100);
    getch();
    closegraph();
    return 0;
}
```

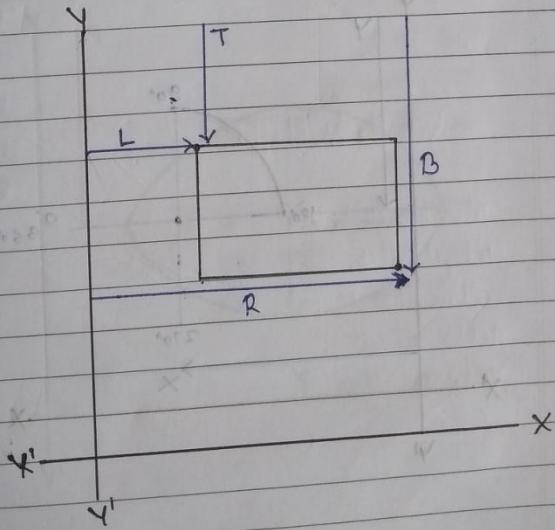
Syntax - circle(x, y, radius);



3- Rectangle -

```
#include <conio.h>
#include <graphics.h>
int main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm, "C:\\TURBOC3\\BG.I\\");
    rectangle(20,20,80,80);
    getch();
    closegraph();
    return 0;
}
```

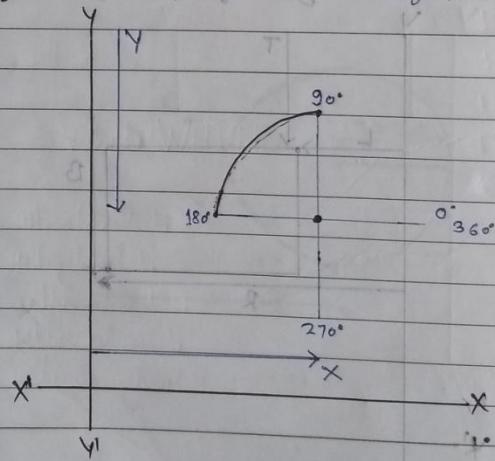
syntax- rectangle (Left, Top, Right, Bottom);



4- Ellipse →

```
#include <conio.h>
#include <graphics.h>
int main()
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "C:\TURBO C\BG1");
    ellipse(100, 100, 90, 180, 100, 50);
    getch();
    closegraph();
    return 0;
}
```

Syntax - ellipse(x, y, Start angle, End angle);



5- Fillellipse —

#include <conio.h>

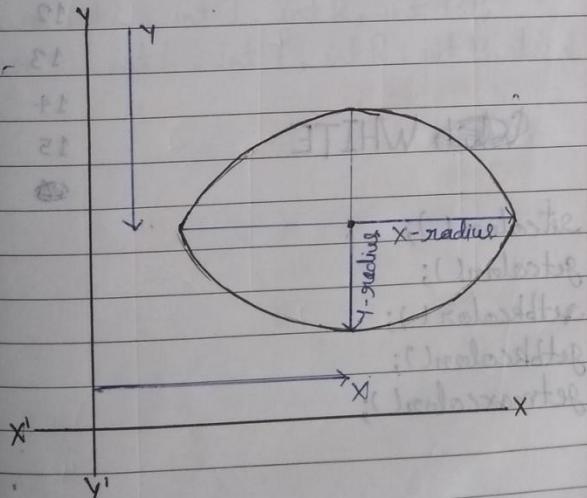
#include <graphics.h>

int main()

{

```
int gd=DETECT, gm;  
initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");  
fillellipse(100, 200, 100, 50);  
getch();  
closegraph();  
return 0;  
}
```

Syntax - fillellipse(x, y, x-radius, y-radius);



Color -

color

BLACK

Value

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

~~Color~~ WHITE

setcolor();
getcolor();
setbkcolor();
getbkcolor();
getmaxcolor();

SOLID_FILL

LINE_FILL

LTSLASH_FILL

SLASH_FILL

BKSLASH_FILL

LTBKSLASH_FILL

XHATCH_FILL

HATCH_FILL

INTEREVE_FILL

WIDE_DOT_FILL

CLOSE_DOT_FILL

setfillstyle(int pattern, int color);

floodfill(int x, int y, int border);

border(int L, int T, int R, int B);

border3d(int L, int T, int R, int B, int depth, int top,

Flag);

Example -

```
1- #include <graphics.h>
    #include <conio.h>
    int main()
    {
        clrscr();
        int gd=DETECT, gm, ix=50, iy=50, c;
        initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
        c=getmaxcolor();
        for(i=0; i<c; i++)
        {
            setcolor(i);
            outtextxy(ix, iy, "I-T-Shiksha");
            iy=iy+20;
        }
        getch();
        closegraph();
        return 0;
    }
```

2-

```
2- #include <graphics.h>
#include <conio.h>
int main()
{
    clrscr();
    int gd=DETECT,gm,i,x=50,y=50,c;
    initgraph(&gd,&gm,"C:\\TURBOC3\\BGI\\");
    setcolor(RED);
    setfillstyle(2,5);
    fillellipse(100,100,50,25);
    circle(200,200,50);
    floodfill(200,200,4);
    getch();
    closegraph();
    return 0;
}
```

```
3- #include <graphics.h>
#include <conio.h>
int main()
{
    int gd=DETECT, gm, i, x=50, y=50, c;
    initgraph(&gd, &gm, "C:\TURBOC3\BGI");
    setfillstyle(SOLID_FILL, RED);
    bar(150, 100, 200, 419);
    setfillstyle(LINE_FILL, GREEN);
    bar(255, 150, 275, 419);
    setfillstyle(SLASH_FILL, BLUE);
    bar3d(300, 420, 350, 150, 10, 1);
    getch();
    closegraph();
    return 0;
}
```

4- #

#

#

L

b

N

A

H

M

S

F

L

C

J

A

S

O

D

E

F

G

H

I

J

K

L

M

N

O

```
4- #include < conio.h >
#include < graphics.h >
#include < dos.h >
int main()
{
    int gd=DETECT, gm, i, j=0;
    initgraph(&gd, &gm, "C:\TURBO C\BGI");
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 2);
    outtextxy(25, 240, "Press Any Key To View Moving
car");
    getch();
    setviewport(0, 0, 639, 440, 1);
    for(i=0; i<420; j++)
    {
        j = i+10;
        rectangle(150+i, 275, 150+i, 400);
        rectangle(150+i, 350, 200+i, 400);
        circle(175+i, 410, 10);
        circle(175+i, 410, 10);
        setcolor(j);
        delay(100);
        if(i==420)
        {
            break;
        }
    }
    clearviewport();
    getch();
    closegraph();
    return 0;
}
```

File Management

Syntax - File Name . File Type / Extension

In Built function -

FILE *fp;

fopen(→ → Filenamet(w/r)

w = Write

a = Add / Append

r = read

fclose(fp);

getc();

putc();

Example -

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

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

```
int main()
```

```
{ clrscr(); // clear screen
```

```
FILE *fp; // file pointer declaration
```

```
char name[20]; // character array declaration
```

```
fp = fopen("Alma.txt", "w"); // open file
```

```
printf("Enter Name: "); // print message
```

```
scanf("%s", name); // read input
```

```
fprintf(fp, "Name: %s", name); // write to file
```

```
fclose(fp); // close file
```

```
getch(); // wait for key
```

```
return 0; // exit program
```

```
}
```

Character -

```
char ch;
```

```
putc;
```

```
getc;
```

Example -

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

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

```
main()
```

```
{
```

```
clrscr();
```

```
FILE *fp;
```

```
char ch;
```

```
fp=fopen("Dell.txt","w");
```

```
printf("Input Line And Press EOF.\n");
```

```
while((ch=getchar())!=EOF)
```

```
{
```

```
putc(ch,fp);
```

```
}
```

```
fclose(fp);
```

```
printf("\nOutput\n");
```

```
fp=fopen("Dell.txt","r");
```

```
while((ch=getch(fp))!=EOF)
```

```
{
```

```
printf("%c",ch);
```

```
}
```

```
fclose(fp);
```

```
getch();
```

```
}
```

String -

```
char str[-];
```

```
fputs;
```

```
fgets;
```

Example -

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

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

```
main()
```

```
{
```

```
clrscr();
```

```
FILE *fp;
```

```
char str[20];
```

```
fp = fopen("Dell1.txt", "w");
```

```
fputs("This is IT Shiksha.", fp);
```

```
fclose(fp);
```

```
fp = fopen("Dell1.txt", "r");
```

```
fgets(str, 20, fp);
```

```
printf("Text is : %s", str);
```

```
fclose(fp);
```

```
getch();
```

```
}
```

Integer -

```

#include <stdio.h> //for file handling
#include <conio.h> //for clrscr()
main()
{
    clrscr(); //clear screen
    FILE *fp;
    int num;
    fp = fopen("Number.txt", "w"); //open file in write mode
    for(i=1; i<=5; i++)
    {
        printf("%d", i); //print numbers from 1 to 5
        if(i==5)
            fputc('\n', fp); //new line character
    }
    fclose(fp); //close file
    fp = fopen("Number.txt", "r"); //open file in read mode
    while((num=fgetc(fp))!=EOF) //read file until EOF
    {
        printf("%d", num); //print read numbers
    }
    fclose(fp); //close file
    getch(); //wait for key press
    return 0;
}

```

fprintf And fscanf Use -

fprintf - Input data

fscanf - Output data

Example -

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

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

```
main()
```

```
{
```

```
char ch; /* for character input */
```

```
FILE *fp; /* for file pointer */
```

```
int roll;
```

```
char name[20]; /* for string input */
```

```
printf("Enter Student Roll No.: ");
```

```
scanf("%d", &roll);
```

```
printf("Enter Student Name: ");
```

```
scanf("%s", name);
```

```
fclose(fp);
```

```
printf("\nStudent Details Are: ");
```

```
fp = fopen("Student", "w");
```

```
fprintf(fp, "%d %s", roll, name);
```

```
fclose(fp);
```

```
printf("\nStudent Details Are: ");
```

```
fp = fopen("Student.txt", "r");
```

```
fscanf(fp, "%d %s", &roll, name);
```

```
printf("\nStudent Name: %s \n Student Roll No. %d", name, roll);
```

```
fclose(fp);
```

```
getch();
```

```

    feof(); and ferror();
    #include <stdio.h>      <directives> header
    #include <conio.h>      <directives> header
    main();
    {
        clrscr();
        FILE *fp;
        char str[20];
        fp=fopen("dell.txt","r");
        gets(str,50,fp);
        printf("Text is : %s",str);
        if(feof(fp)==0)
        {
            printf("\nEnd Of File");
        }
        if(ferror(fp)==0)
        {
            printf("\nError");
        }
        fclose(fp);
        getch();
    }
    student Roll No. 'id',
    name, rname,
    );

```

Checking File Open Or Not Open -

#include < stdio.h > < Wright.h > std::wif

#include < conio.h > < d.mir > std::wi

#include < stdlib.h > : () mors

main()

{

clrscr();

FILE *fp;

char str[20], file[20]; // () mors

printf("Enter File Name:");

scanf("%s", file);

fp = fopen(file, "r");

if (fp == NULL)

{

printf("File Does Not Exist");

getch();

exit(0);

}

fgets(str, 50, fp);

printf("\nFile Content: %s", str);

fclose(fp);

getch();

}

Read	r
Write	w
Add / Append	a
Read and Write	r+ (Current position) w+ (File text erase or create New File) a+ (End position)

Note - For back position use symbol "--" before position Numerical value.
And this work for only Negative value.