



# ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle  
and Graded as Category-I University by MHRD-UGC]

(A State University Established by the Government of Tamil Nadu)

KARAIKUDI – 630 003



## Directorate of Distance Education

### B.Sc. [Computer Science]

I - Semester

130 14

## LAB: PROGRAMMING IN C

**Author**

**Shweta Harsh Tiwari, *Freelance Author***

"The copyright shall be vested with Alagappa University"

All rights reserved. No part of this publication which is material protected by this copyright notice may be reproduced or transmitted or utilized or stored in any form or by any means now known or hereinafter invented, electronic, digital or mechanical, including photocopying, scanning, recording or by any information storage or retrieval system, without prior written permission from the Alagappa University, Karaikudi, Tamil Nadu.

Information contained in this book has been published by VIKAS® Publishing House Pvt. Ltd. and has been obtained by its Authors from sources believed to be reliable and are correct to the best of their knowledge. However, the Alagappa University, Publisher and its Authors shall in no event be liable for any errors, omissions or damages arising out of use of this information and specifically disclaim any implied warranties or merchantability or fitness for any particular use.



VIKAS® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.

E-28, Sector-8, Noida - 201301 (UP)

Phone: 0120-4078900 • Fax: 0120-4078999

Regd. Office: 7361, Ravindra Mansion, Ram Nagar, New Delhi 110 055

• Website: [www.vikaspublishing.com](http://www.vikaspublishing.com) • Email: [helpline@vikaspublishing.com](mailto:helpline@vikaspublishing.com)

**Work Order No. AU/DDE/DE1-238/Preparation and Printing of Course Materials/2018 Dated 30.08.2018 Copies - 500**

---

# LAB: PROGRAMMING IN C

---

## SYLLABI

---

### BLOCK 1: C PROGRAM FUNDAMENTALS

1. Simple C programs
  2. Using IF and switch constructs programs
  3. Looping related problems
- 

### BLOCK 2: FUNCTIONS, ARRAYS, STRINGS

4. Programs using functions
  5. IF statement, If..else statement, nesting if else statement, else if ladder, switch statement, goto statement, while statement, do statement, for statement
  6. One-dimensional arrays, two dimensional arrays, multi dimensional arrays
  7. Initialization of string variables, reading and writing strings, string handling functions
- 

### BLOCK 3: STRUCTURE AND UNIONS

8. Programs using structures
  9. Programs using unions
- 

### BLOCK 4: POINTERS

10. Initialization of pointer variables, address of variable, accessing a variable through its pointer
  11. Pointer as Functions
  12. Strings with Pointer: pointers and character strings, pointers and structures
- 

### BLOCK 5: FILES

13. Programs based on file handling
  14. Error Handl
-

---

## INTRODUCTION

---

### NOTES

C Programming is an ANSI/ISO standard and powerful programming language for developing real-time applications. C programming language was invented by Dennis Ritchie at the Bell Laboratories in 1972. It was invented for implementing UNIX operating system. C is most widely used programming language even today. All other programming languages were derived directly or indirectly from C programming concepts.

This lab manual, *Programming in C*, contains several programs based on C concepts, such as IF and Switch, Looping, Functions and Arrays, to provide the concept of programming for beginners. In addition, it will help students in coding and debugging their programs. The book provides all logical, mathematical and conceptual programs that can help to write programs very easily in C language. These exercises shall be taken as the base reference during lab activities for students of B.Sc. (Computer Science). There are also many Try Yourself Questions provided to students for implementation in the lab.

# BLOCK 1 C PROGRAM FUNDAMENTALS

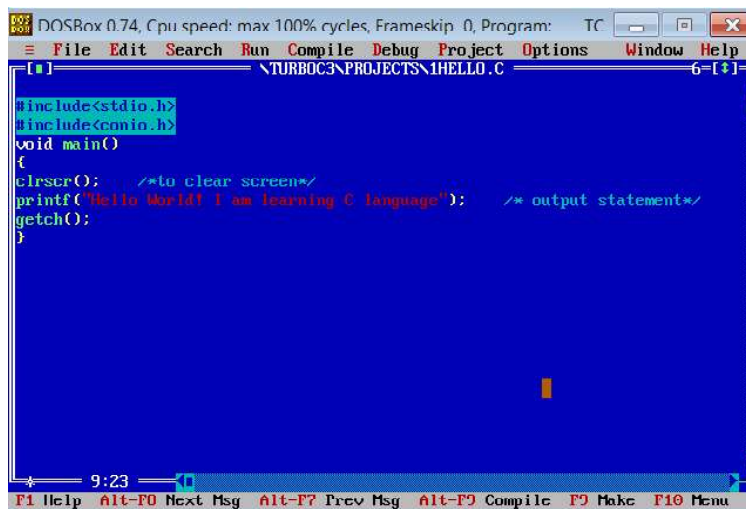
## NOTES

The first program while learning any programming language is 'Hello world' program. This program is used to check the programming environment. It will give you idea of program editor, how to compile program and how to run program? This program is very useful in learning the menus, shortcut and increase our confidence. It is the easiest program but still we have to run this program for understanding programming. For every program you have to first :

Write a program code → compile (Ctrl+F9) → Run(Alt +F9) → observe the output → save your program (F2), you can also save your program before compiling.

### 1. Write a program to print "Hello world!" on screen (Hello.c)

**Step 1:** Type the program in C or C++ editor

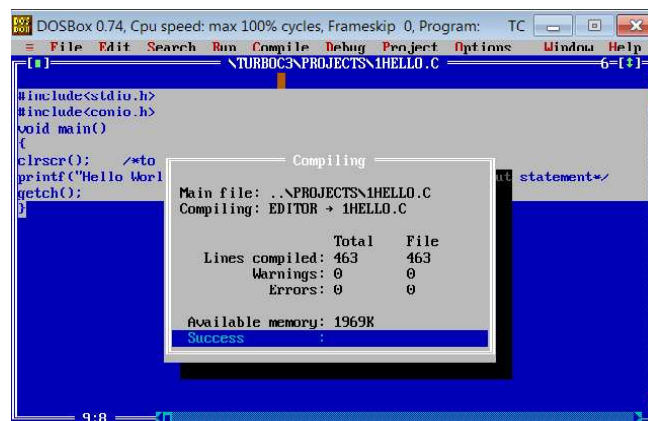


```

#include<stdio.h>
#include<conio.h>
void main()
{
clrscr(); /*to clear screen*/
printf("Hello World! I am learning C language"); /* output statement*/
getch();
}

```

**Step 2:** Click on compile to translate it into binary format (automatically .obj (object) file created. **Short cut for compiling is Ctrl +F9**



```

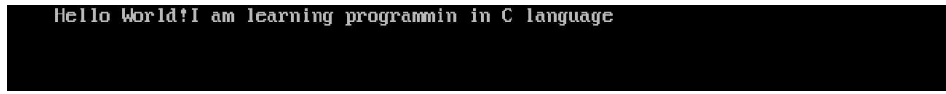
Main file: ..\PROJECTS\1HELLO.C
Compiling: EDITOR -> 1HELLO.C

```

	Total	File
Lines compiled:	463	463
Warnings:	0	0
Errors:	0	0

Available memory: 1969K  
Success

Step 3: Run the program and automatically \*.exe is generated. Short cut for run is Alt +F9. Output window will show you the output

**Output:****NOTES**

```
Hello World! I am learning programmin in C language
```

**2. Write a program to input your name and age and display on screen.**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
clrscr();                               /*to clear screen*/

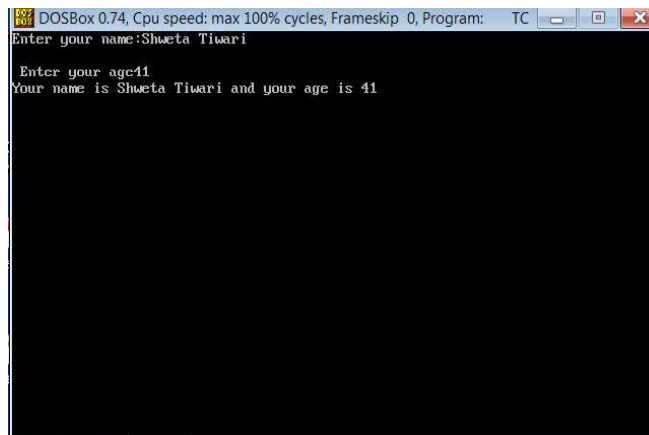
int age;                                 /*variable declaration */
char myname[20];

printf("Enter your name:");
gets(myname);                            /*input for string present in
string.h*/

printf("\n Enter your age");
scanf("%d",&age);                         /*input for integer*/

printf( "Your name is %s and your age is %d", myname, age
);    /*output for string and
integer*/

getch();
}
```

**Output Window:**

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter your name:Shweta Tiwari
Enter your age:41
Your name is Shweta Tiwari and your age is 41
```

### 3. Write a program to input principle, rate and time and display simple interest.

```
#include<stdio.h>
#include<conio.h>
void main()
{
float p,r,t,SI;          /*variable declaration */
clrscr();              /*to clear screen*/

printf("Enter Principle amount :");
scanf("%f",&p);        /*input */

printf("Enter rate of interest :");
scanf("%f",&r);

printf("Enter time in years :");
scanf("%f",&t);

SI=(p*r*t)/100;

printf("You have to pay simple interst of Rs.%f",SI);
/*output */

printf("\nTotal amount =%f",p+SI);
getch();
}
```

## NOTES

#### Output:

```
Enter Principle amount :1000
Enter rate of interest :10
Enter time in years :2
You have to pay simple interst of Rs.200.000000
Total amount =1200.000000_
```

### 4. Write a program to enter temperature in Fahrenheit and convert it in Celsius.

Formula :-  $C = (F - 32) * 5 / 9$

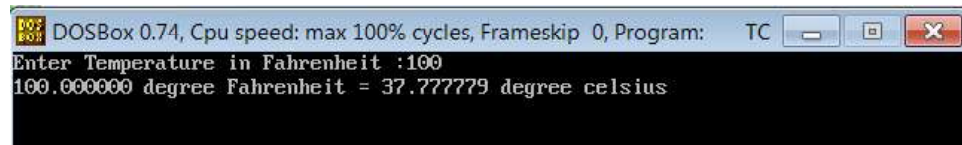
```
#include<stdio.h>
#include<conio.h>
void main()
{
float f,c;          /*variable declaration */
clrscr();          /*to clear screen*/
```

**NOTES**

```

printf("Enter Temperature in Fahrenheit :");
scanf("%f",&f);      /*input */
c=(f-32)*5/9;        //calculation
printf( "%f degree Fahrenheit = %f degree celsius",f,c);
getch();
}

```

**Output:**


```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter Temperature in Fahrenheit :100
100.000000 degree Fahrenheit = 37.777779 degree celsius

```

**5. Write a program to convert Celsius to Fahrenheit.**

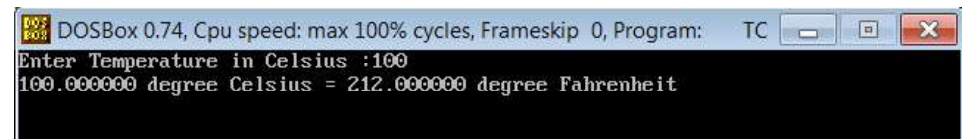
Formula :-  $F=c*9/5+32$

```

#include<stdio.h>
#include<conio.h>
void main()
{
float f,c;          /*variable declaration */
clrscr();          /*to clear screen*/
printf("Enter Temperature in Celsius :");
scanf("%f",&c);    /*input */
f=c*9/5+32;        //calculation
printf( "%f degree Celsius = %f degree Fahrenheit ",c,f);
/*output */

getch();
}

```

**Output:**


```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter Temperature in Celsius :100
100.000000 degree Celsius = 212.000000 degree Fahrenheit

```

**6. Write a program to swap the values of variables.**

Swapping is the technique to interchange the values, this technique is used in many situation in programming.

```

#include<stdio.h>
#include<conio.h>
void main()

```



```

{
int x, y, temp;          /*variable declaration */
clrscr();                /*to clear screen*/
printf("Enter the values of two variables X and Y");
scanf("%d%d",&x, &y);  /*input */
printf("value of X=%d and value of Y=%d",x,y);

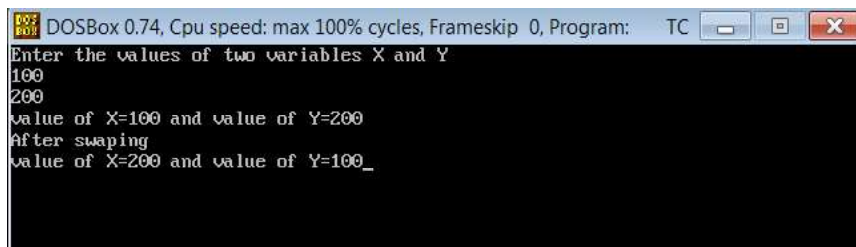
temp=x;                  //swapping
x=y;
y=temp;

printf("\nAfter swaping \nvalue of X=%d and value of
Y=%d",x,y);

getch();

}

```

**NOTES****Output:**

**7. Write a program to calculate your present age by entering your year of birth.**

Formula : current year- DOB\_year

```

#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int age,dob_year,curr_year;  /*variable declaration */
struct date d;              /*creates date variable

clrscr();                    /*to clear screen*/

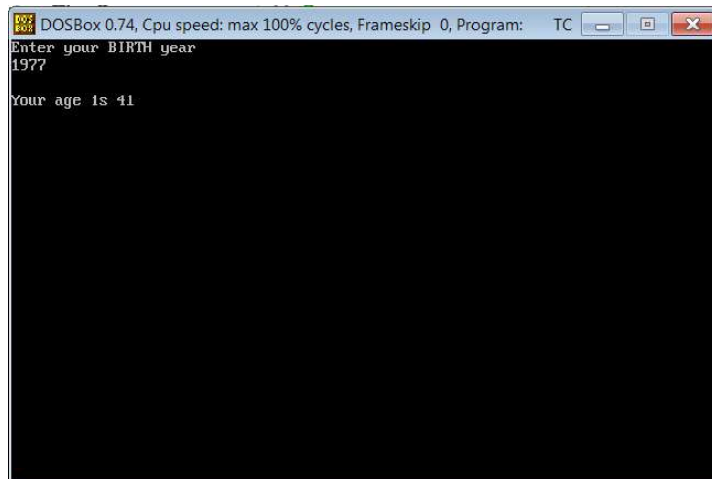
printf("Enter your BIRTH year");
scanf("%d",&dob_year);

```

## NOTES

```
getdate(&d);           //get current date in d variable  
curr_year=d.da_year;  //get current year in variable  
  
age=curr_year-dob_year; //calculate age  
  
printf("\nYour age is %d",age);  
getch();  
  
}
```

### Output:



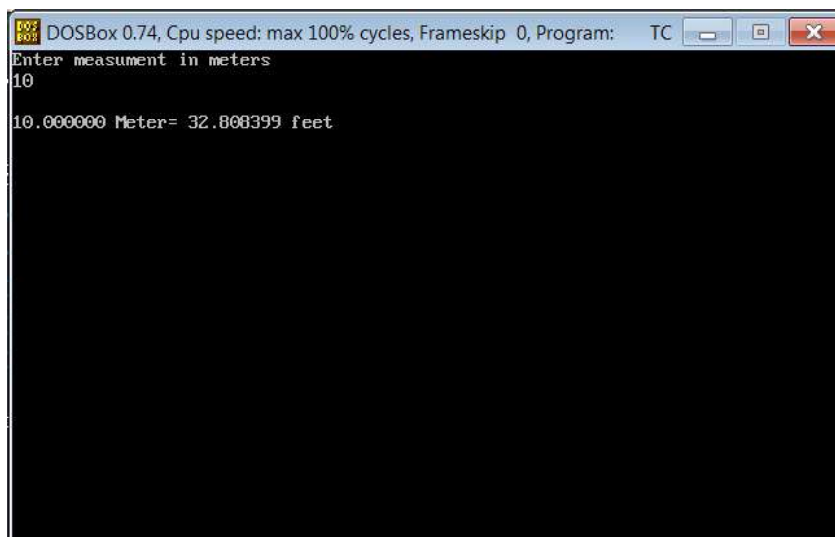
### 8. Write a program to enter measurement in meter and convert it into feet.

Formula :  $ft=m/0.3048$

```
#include<stdio.h>  
#include<conio.h>  
#include<dos.h>  
void main()  
{  
float m,f;           /*variable declaration */  
  
clrscr();           /*to clear screen*/  
  
printf("Enter measument in meters");  
scanf("%f",&m);  
  
f=m/0.3048;  
  
printf("\n%f Meter= %f feet",m,f);
```

```
    getch();  
}
```

**Output:**



**NOTES**

**9. Write a program to convert feet to meter.**

Formula :  $m = f * 0.3048$

```
#include<stdio.h>  
#include<conio.h>  
#include<dos.h>  
void main()  
{  
    float m,f;          /*variable declaration */  
    clrscr();          /*to clear screen*/  
    printf("Enter measument in feets");  
    scanf("%f",&f);  
  
    m=f*0.3048;  
  
    printf("\n%f Feet= %f Meter",f,m);  
    getch();  
}
```

**10. Write a program to display ASCII character of any value between 0 to 255.**

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

**NOTES**

```

void main()
{
    unsigned int A;          /*variable declaration */
    clrscr();                /*to clear screen*/

    printf("Enter value for ASCII character between 0-255
    ");
    scanf("%d",&A);        //input value of ASCII character

    printf("\nASCII value %d = ASCII character %c",A,A);
    getch();
}

```

**Output:**

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter value for ASCII character between 0-255
65
ASCII value 65 = ASCII character A_

```

**11. Program to convert ASCII character to ASCII value.**

```

#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
    unsigned char A;        /*variable declaration */
    clrscr();                /*to clear screen*/

    printf("Enter ASCII character by pressing any key form
    keyboard ");
    scanf("%c",&A);        //input value of ASCII character

    printf("\nASCII character  %d = ASCII value %d",A,A);
    getch();
}

```

**Note:** const is a keyword used to declare constant variable.

**12. Write a program to calculate area and circumference of circle.  
Declare PI as constant.**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
const float PI=3.14;      /* constant variable
                           declaration */

float r,A,C;

clrscr();                /*to clear screen*/

printf("Enter Radius of a circle ");
scanf("%f",&r);          //input value of radius

A=PI *r*r;
C=2*PI*r;

printf("\nArea od a circle =%f \n\n Circumference of
circle=%f",A,C); //output

getch();

}
```

**NOTES**

**Output:**

```
Enter Radius of a circle 10
Area od a circle =314.000000

Circumference of circle=62.800003_
```

**13. What will be the output of program given below?**

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

void main()
{
int a=3,b=5,c;
c=!a || (b=7);
printf("\nb=%d\nc=%d", b,c);
getch();
}
```

**NOTES****14. Write the output of following program.**

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

void main()
{
    int x,y,z;
    x=48;
    y=49;
    z=50;

    printf("\n%c \t%c \t%c",x,y,z);
    getch();
}
```

**15. Write the output of the program given below.**

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

void main()
{
    int x,y,z;
    x=10;
    y=11;
    z=12;

    printf("\n%x \t%x \t%x",x,y,z);
    getch();
}
```

**16. Write the output of following program.**

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

void main()
{
    int x,y,z;
    x=10;
    y=11;
    z=12;

    printf("\no \to \to",x,y,z);
    getch();
}
```

**Try yourself:-**

- (i) Write a program to calculate volume of cylinder.

$$\text{Volume of cylinder} = \text{PI} * \text{r} * \text{r} * \text{h}$$

- (ii) Write a program to calculate curved surface area of cylinder.

$$\text{Curved surface area of cylinder} = 2 * \text{PI} * \text{r} * \text{h}$$
**NOTES****17. Write a program to input a character in lower case and convert it into upper case.**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
char x;                /* variable declaration */
clrscr();              /*to clear screen*/

printf("Enter any letter in small between a-z:- ");
scanf("%c",&x);        //input value of x

x=x-32;                //ASCII difference between small
                        and capital is 32( 97-65=32)

printf("\nSmall Letter x %c is converted into capital
                        letter %c",x+32,x); //output

getch();

}
```

**Output:**

```
Enter any letter in small between a-z:- s
Small Letter x s is converted into capital letter S_
```

Functions to read and write character form keyboard are:

Header file <stdio.h>

Functions to read single character: getch (), getchar ()

Example: x=getch ();

```
x=getchar ();
```

Function to put/write single character: putchar (variable);

Example: putchar(x);

**18. Write a program to enter a character by using `getchar ()`, convert it into opposite (small to capital and capital to small) and use `putchar ()` function to display result.**

**NOTES**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
    Char x;                /* variable declaration */
    Clrscr ();            /*to clear screen*/
    printf ("Enter any letter :- ");
    x=getchar ();        //input value of x
    if (x>=65 && x<=90)    //check letter is capital
        x=x+32;
    else if (x>=97 && x<=122)
        //check letter is small

        x=x-32;
    else
        printf ("Letter is not an Alplabet , Sorry can't
        convert");
    printf ("\nConverted letter is ");
    putchar (x);        //output
    getch ();
}
```

**Output:**

In case of capital letter

```
Enter any letter :- C
Converted letter is c_
```

In case of small letter

```
Enter any letter :- b
Converted letter is B
```

In case of any other character

```
Enter any letter :- 9
Letter is not an Alplabet , Sorry can't convert
Converted letter is 9_
```



**19. Write a program to check whether number is negative or positive using if...else.**

```

Logic : if number >=0 then positive else negative
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int x;          /* variable declaration */
clrscr ();     /*to clear screen*/
printf ("Enter any number :- ");
scanf ("%d",&x); //input value of x
if (x>=0)     //check for positive number
printf ("%d is a positive number",x);
else
printf ("%d is negative number",x);
getch ();
}

```

**NOTES****Output:**

In case of positive number

```

Enter any number :- 6
6 is a positive number_

```

In case of negative number

```

Enter any number :- -100
-100 is negative number_

```

**20. Write a program to enter a number and display whether the number is positive or negative using ternary operator.**

Ternary operator is a control statement similar to if...else statement:

Syntax of ternary operator is:

```
(Condition)?true statement :false statement ;
```

Where condition is formed using variable and operator (relational operator), true statement is the statement to be executed when the condition is true and false statement will be executed when the condition is false (just like else statement). It is used when logic is very simple in conditional statement.

**NOTES****Program:**

```

#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int x;          /* variable declaration */
clrscr ();     /*to clear screen*/

printf ("Enter any number :- ");
scanf ("%d",&x); //input value of x

(x>=0)?printf ("Number is positive"):printf("Number is
negative");

getch ();
}

```

**Output:**

In case of positive number

```

Enter any number :- 100
Number is positive

Enter any number :- -200
Number is negative_

```

**Try yourself:**

- (i) Write a program to display greatest number out of two numbers using ternary operator.
- (ii) Write a program to enter your age, and display the message whether "eligible to vote" or "not eligible to vote".

**21. Write a program to check whether number is even or odd.**

Logic : If the number is even , it is completely divisible by 2.

% (modulus ) is a operator , which return remainder after division and if the remainder is zero that means number is even. This logic applicable for checking any number divisibility.

**Program:**

```

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

```

```

#include<dos.h>
void main()
{
int x;      /* variable declaration */
clrscr (); /*to clear screen*/

printf ("Enter any number :- ");
scanf ("%d",&x);      //input value of x

if (x % 2==0)
printf ("Number is even");
else
printf ("Number is odd");

getch ();

}

```

**NOTES****Output:**

In case of even

```

Enter any number :- 12
Number is even

```

In case of odd

```

Enter any number :- 15
Number is odd_

```

**Try yourself:**

- (i) Write a program to enter a number and display whether number is divisible by 5 or not.
- (ii) Write a program to enter a number and check number is divisible by 2 and 4 both.
- (iii) Write a program to enter a number and check whether number is divisible by 3 and 7.

**22. Write a program to check character is vowel or consonant .(Example of nested if...else)**

```

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

```

## NOTES

```
void main()
{
char   x;   /* variable declaration */
clrscr(); /*to clear screen*/

printf ("Enter any character :- ");
x= getchar ();

if ((x>=65 && x<=90) || (x>=97 &&x<=122))
{
    if
(x=='A' || x=='E' || x=='I' || x=='O' || x=='U' || x=='a' || x=='e' || x=='i' || x=='o' || x=='u')
    {
        Printf ("Character is vowel");
    }
    else
    {
        printf ("Character is a consonant");
    }
}
else
{
printf (" Character is not an alphabet");
}
getch ();

}
```

### Output:

In case of vowel

```
Enter any character :- A
Character is vowel

Enter any character :- F
Character is a consonant

Enter any character :- 5
Character is not an alphabet
```

### 23. Write a program to enter two numbers and find the greatest number.

C Program Fundamentals

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int x,y;    /* variable declaration */
clrscr (); /*to clear screen*/

printf ("Enter any two numbers :- ");
scanf ("%d%d",&x,&y);

if (x>y)
{
printf ("%d is greater ",x);
}
else if(y>x)
{
printf ("%d is greater",y);
}
else
{
printf ("Both are equal");
}
getch ();
}
```

### NOTES

#### Output:

In case of greater

```
Enter any two numbers :- 10
20
20 is greater_
```

In case of equal

```
Enter any two numbers :- 30
30
Both are equal_
```

**24. Write a program to enter three numbers and find the smallest number.****NOTES**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int x,y,z; /* variable declaration */
clrscr (); /*to clear screen*/

printf ("Enter any three numbers :- ");
scanf ("%d%d%d",&x,&y,&z);

if (x>y && x>z)
{
printf ("%d is greater ",x);
}
else if (y>x && y>z)
{
printf ("%d is greater",y);
}
else if (z>x && z>y)
{
printf ("%d is greater",z);
}
else
{
printf ("All the equal");
}

getch ();
}
```

**Output:**

In case of greater

```
Enter any three numbers :- 10
20
30
30 is greater_
```

In case of all equal

C Program Fundamentals

```
Enter any three numbers :- 100
100
100
All the equal
```

## NOTES

### 25. Write a program to check leap year or not.

Logic :- Leap year is divisible by 4

Use of %(modulus operator) to check divisibility test

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int Y; /* variable declaration */
clrscr (); /*to clear screen*/

printf ("Enter year to check :- ");
scanf ("%d",&Y);
if (Y % 4 == 0) //divide by 4 and compare with zero
{
Printf ("%d year is a leap year and it has 29 days in Feb
and Total 366 days in a year",Y);
}
else
{
printf ("%d year is not a leap year and it has 28 days in
Feb Total 365 days in a year",Y);
}
getch ();
}
```

### Output:

In case of not a leap year

```
Enter year to check :- 2018
2018 year is not a leap year and it has 28 days in Feb Total 365 days in a year
```

In case of leap year

```
Enter year to check :- 2020
2020 year is a leap year and it has 29 days in Feb and Total 366 days in a year
```

## NOTES

**Try yourself:**

- (i) Write a program to enter your year of birth and check whether the year is leap year or not and also check how many leap years occurs after your birth.

**Hint:** Refer program 7 for the use of struct date structure. Also refer program 21 to check leap year . use calculation to find out number of leap years after your birth.

**26. Write a program to enter any character and find whether number is character, digit, or other character.**

Logic : If ASCII value is between (65 and 90) or(97 and 122) then character

Else If value is between (48 and 58) then digit

Else other character

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
char x;                /* variable declaration */
clrscr ();             /*to clear screen*/
printf ("Enter any character to check :- ");
scanf ("%c",&x);
if ((x >=65 && x<=90)|| (x>=97 &&x<=122))
                        //check for character
{
printf ("%c is an alphabet",x);
}
else if(x>=48 &&x<=57) //check for digit
{
printf ("%c is a digit ",x);
}
else
{
printf ("%c is not an alphabet nor an digit , it is
other character");
}
getch ();
}
```



**Output:**

In case of an alphabet

```
Enter any character to check :- A
A is an alphabet
```

In case of an digit

```
Enter any character to check :- 9
9 is a digit _
```

In case of an other character

```
Enter any character to check :- %
% is not an alphabet nor an digit , it is other character _
```

**NOTES****27. Write a program to enter number and check whether divisible by 4, 12 or both.**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int x;          /* variable declaration */
clrscr ();     /*to clear screen*/
printf ("Enter any number to check divisibility :- ");
scanf ("%d",&x);

if ((x %4 == 0) && (x %12 ==0))
{
printf ("%d is divisible by both 4 and 12",x);
}
else if(x %4 ==0)
{
printf ("%d is divisible by only 4 and not 12 ",x);
}
else if (x % 12 ==0)
{
printf ("%d is divisible only by 12 and not by 4",x);
}
else
```

**NOTES**

```
{  
    printf ("%d is not divisible by any of them",x);  
}  
getch ();  
}
```

**Output:**

In case of only 4

```
Enter any number to check divisibility :- 4  
4 is divisible by only 4 and not 12 _
```

In case of both

```
Enter any number to check divisibility :- 84  
84 is divisible by both 4 and 12 _
```

In case of divisible by 4 and not 12

```
Enter any number to check divisibility :- 104  
104 is divisible by only 4 and not 12
```

In case of not divisible by any of them 4 and 12

```
Enter any number to check divisibility :- 42  
42 is not divisible by any of them _
```

**28. Write a program to enter a number between (1 to 7) and display accordingly Sunday to Saturday using switch case. Use goto statement to continue program till user's press Y(Yes)**

**Note:** 1 will be considered as Sunday and 7 will be as Saturday.

Switch ...case statement is used when we have many cases or conditions.

Syntax of switch ...case is as follows:

```
switch (variable)  
{  
    case value1: // executed if variable =value1;  
        break;  
    case value2: // executed if variable =value2;  
        break;  
    default: // executed if variable is not equal to any  
value  
}
```

**Program:**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int x;          /* variable declaration */
char ch;
clrscr();      /*to clear screen*/

startagain:    //label of goto statement

printf("\nEnter any number to display corresponding day
(Sun-Sat)between 1 to 7 :- ");
scanf("%d",&x);

switch (x)
{
case 1: printf("Sunday");
        break;
case 2: printf("Monday");
        break;
case 3: printf("Tuesday");
        break;
case 4: printf("Wednesday");
        break;
case 5: printf("Thursday");
        break;
case 6: printf("Friday");
        break;
case 7: printf("Saturday");
        break;
default: printf(" Not a valid number");
        break;
}
printf("\nDo you want to continue: Press Y or N");
ch=getch();

if(ch=='y' || ch =='Y')
```

**NOTES**

```

goto startagain;
else
printf("\nProgram is closing ");

getch();
}

```

**NOTES****Output:**

```

Enter any number to display corresponding day (Sun-Sat)between 1 to 7 :- 1
Sunday
Do you want to continue: Press Y or N
Enter any number to display corresponding day (Sun-Sat)between 1 to 7 :- 2
Monday
Do you want to continue: Press Y or N
Enter any number to display corresponding day (Sun Sat)btween 1 to 7 :- 3
Tuesday
Do you want to continue: Press Y or N
Enter any number to display corresponding day (Sun-Sat)between 1 to 7 :- 7
Saturday
Do you want to continue: Press Y or N
Enter any number to display corresponding day (Sun-Sat)between 1 to 7 :- 9
Not a valid number
Do you want to continue: Press Y or N
Enter any number to display corresponding day (Sun-Sat)between 1 to 7 :- 6
Friday
Do you want to continue: Press Y or N
Program is closing _

```

**29. Write the output of the following program :**

```

#include<stdio.h>
#include<conio.h>
void main()
{
int n;
printf("Enter no. between 1 and 3 ");
scanf("%d",&n);
clrscr();
switch(n)
{
case 1: printf("\n1");
break;
case 2: printf("\n2");
break;
case 3: printf("\n3");
break;
default: printf("\nWrong choice");
break;
}
getch();
}

```

**30. Write a program to enter two numbers x and y and also enter an operator (+,-,\*,/,%). Find out the result using switch case. Continue your program till the user press 'Y' to continue.**

Hint : use goto statement to continue program till user press 'y' and also use fflush(stdin); function to clear buffer, otherwise the user is not allowed to enter character

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int x,y,z;          /* variable declaration */
char ch,op;
clrscr();          /*to clear screen*/

startlabel:        //label of goto statement

printf("\nEnter the operator to be used with x and y( +,
-, *, /,%)" );
fflush(stdin);    //to clear memory of temp buffer
scanf("%c",&op);
printf("\nEnter any two numbers as x and y:- ");
scanf("%d%d",&x,&y);
switch (op)
{
case '+': printf("\nSum=%d",x+y);
           break;
case '-': printf("\nDifferen=%d",x-y);
           break;
case '*': printf("\nProduct=%d",x*y);
           break;
case '/': printf("\nQuotient=%d",x/y);
           break;
case '%': printf("\nRemainder=%d",x % y);
           break;
default: printf(" Not a valid operator");
         break;
}
}
```

## NOTES

**NOTES**

```

printf("\nDo you want to continue: Press Y or N");
ch=getch();

if(ch=='y' || ch=='Y')
goto startlabel;
else
printf("\nProgram is closing ");

getch();
}

```

**Output:**

```

Enter the operator to be used with x and y( +, -, *, /,%)+
Enter any two numbers as x and y:- 10
20

Sum=30
Do you want to continue: Press Y or N
Enter the operator to be used with x and y( +, -, *, /,%)-
Enter any two numbers as x and y:- 40
15

Differen=25
Do you want to continue: Press Y or N
Enter the operator to be used with x and y( +, -, *, /,%)*
Enter any two numbers as x and y:- 12
2

Product=24
Do you want to continue: Press Y or N
Enter the operator to be used with x and y( +, -, *, /,%)_

```

**31. Write a program to display numbers from 1 to 10 using goto statement.**

**Note:** goto can be used to repeat statement for number of times. Goto is controlled by if...else statement. Remember when we use goto statement in program the speed of a program is getting slow. It's better to use loop instead of goto statement. Goto is not considered as good programming practice.

```

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

void main()
{
int x=1;      /* variable declaration */
clrscr();    /*to clear screen*/

```

```

labelx:

printf("\n%d",x);
x=x+1;

if(x<=10)
goto labelx;

getch();
}

```

**Output:**

```

1
2
3
4
5
6
7
8
9
10_

```

**32. Predict the output of the following program.**

```

#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int x=1;    /* variable declaration */
clrscr();  /*to clear screen*/

A:
printf("\n%d",x);
x=x+1;

B:
printf("\n%d",x+1);
x=x+2;

if(x<=10)
goto A;
else if(x<=20)
goto B;
else
goto C;

```

**NOTES**

**NOTES**

```
C:
getch();
}
```

**33. Predict the output of the following program:**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int a=10;    /* variable declaration */
clrscr();   /*to clear screen*/

Label1:

printf("\n%d",a);
a=a+10;

Label2:
printf("\n%d",a);
a=a+20;
if(a<=100)
goto Label1;
else if(a>=100 && a<=300)
goto Label2;

getch();
}
```

**34. Write a program to display numbers from 20 to 1 in reverse order using goto statement.**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int a=20;    /* variable declaration */
clrscr();   /*to clear screen*/

Label1:
```



```
printf("\n%d",a);
a=a-1;

if(a>=1)
goto Labell;

getch();
}
```

**Output:**

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

**NOTES****35. Write a program to print your name on Screen using goto statement for infinite time.**

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{

clrscr();    /*to clear screen*/

Labell:
printf("Shweta Tiwari");

goto Labell;

getch();
}
```

**Output:**

**NOTES**

```

ta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta
TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta T
iwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiw
ariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiwar
iShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariS
hweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShw
a TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta
TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Ti
wariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiwa
riShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiwari
Shweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariSh
weta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShwe
+? .TiwariShweta? .TiwariShweta? .TiwariShweta? .TiwariShweta? .TiwariShweta? .TiwariShweta?
iwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiw
ariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiwar
iShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariS
hweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShw
eta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShwet
a TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta
wariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiwa
riShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiwari
Shweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariSh
weta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShwe
ta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta TiwariShweta
TiwariShweta TiwariShweta TiwariShweta TiwariShweta Tiwari
    
```

Press Alt+Ctrl+Del and open task manager and close the program.

**Looping**

Looping means to repeat certain statements for finite or infinite number of time .C language has three types of loops. You can use any of the loop to write a program. It's totally based on programmer choice to select type of loop. Once you will be able to run your program manually on paper then you will be mastering loop. When we run program on paper, then the process is known as dry run.

**Nested Loop:**

Loop within loop is called nested loop.

Three types of loops are

- while loop
- do while loop
- for loop
- **Syntax of while loop...**  

```

initialization;
while (condition is true)
{
Statements to be executed;
}
        
```
- **Syntax of do while loop:** do while will be executed at least once as the condition is checked at the end.  

```

Initialization;
        
```

```
do{
statements;
}while(condition is true);
```

- **Syntax for for...loop**

```
for(initialization; condition ; increment/ decrement )
{
Statement;
}
```

### 36. Write a program to print first 10 even numbers using while loop.

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
int E=2;          /*variable initialization */

clrscr();        /*to clear screen*/

while(E<=20)
{
printf("\n%d",E);
E=E+2;
}
getch();
}
```

#### Output:

```
2
4
6
8
10
12
14
16
18
20_
```

### 37. Write a program to print numbers from 10 to 1 in reverse order using while loop.

Program

```
#include<stdio.h>
#include<conio.h>
void main()
```

## NOTES

**NOTES**

```

{
int n=10;          /*variable initialization */

clrscr();        /*to clear screen*/

while(n>=1)
{
printf("\n%d",n);
n=n-1;
}

getch();
}

```

**Output:**

```

10
9
8
7
6
5
4
3
2
1

```

**Note:** In program 31, observe the condition and statement. condition is  $E \leq 20$  and  $E = E + 2$  (increment , from less to high) . In program number 32, condition is  $n \geq 1$  and statement is  $n = n - 1$  ; (more to less, decrement ) .

**Try yourself:**

- (i) Write a program to display first 20 odd numbers using the while loop.
- (ii) Write a program to print table of 5 on screen using while loop.

**38. Write a program to print the series on screen using while loop.**

```

        6        12        18        24        30        .....n times

#include<stdio.h>
#include<conio.h>
void main()
{
int n,t=6,x=1;          /*variable initialization */
clrscr();              /*to clear screen*/
printf("Enter the value of n");
scanf("%d",&n);
while(x<=n)

```

```

{
printf("\t%d",x*t);
x=x+1;
}
getch();
}

```

**Output**

```

Enter the value of n?
6      12      18      24      30      36      42_

```

**Try yourself:**

- (i) Write a program to display your name for 10 times on screen using while loop.
- (ii) Write a program to display the following series on screen.

```

4      10      16      22      28      34

```

**39. Write a program to print fabonacci series.**

```

0      1      1      2      3      5      8      13.....n

```

Fabonacci series is sued to learn programming and looping concept. The first two elements of fabonacci series are 0 and 1 and next element will be the sum of last two elements. In this we will take two variables a=0 and b=1 as first two elements and find out the next element by adding them .Also in every loop turn last two elements will be changed.

```

#include<stdio.h>
#include<conio.h>
void main()
{
int a=0,b=1,c,n;          /*variable initiliazation */

clrscr();                /*to clear screen*/
printf("Enter the value of n for number of term in
Fabonacci series");
scanf("%d",&n);

printf("\t%d\t%d",a,b); // as we displayed two terms
                        so n>=3

while(n>=3)
{

```

**NOTES**

```

c=a+b;

printf("\t%d",c);

a=b;
b=c;

n=n-1;
}

getch();

}

```

**NOTES****Output:**

```

Enter the value of n for number of term in Fabonacci series?
0      1      1      2      3      5      8      13      21_

```

**Try yourself:**

(i) Write a program to print the  $n^{\text{th}}$  value of Fibonacci series.

Example  $n=10$  then the result  $10^{\text{th}}$  Fibonacci element = 34

**40. Write a program to enter a number and check whether number is prime or not.**

Simple logic behind prime number is divisible by 1 and itself. We will count the number of times the number is completely divisible by other number from 1 to n.

```

#include<stdio.h>
#include<conio.h>
void main()
{
int n,count=0,x=1;          /*variable initiliazation */

clrscr();                  /*to clear screen*/

printf("Enter the number to check whether numner is prime
or not");
scanf("%d",&n);

while (x<=n)
{

```

```

if(n % x == 0)
count++;

x++;

}
if(count==2)
printf("Number is prime ");
else
printf("Number is not a prime number that means composite
number");

getch();
}

```

**NOTES****Output:**

In case of not a prime number

```

Enter the number to check whether number is prime or not
10
Number is not a prime number that means composite number_

```

In case of prime number

```

Enter the number to check whether number is prime or not
5
Number is prime

```

**41. Write a program to find out the Armstrong numbers from 0 to 999 range.**

This program computes all Armstrong numbers in the range of 0 and 999.

An Armstrong number is a number such that the sum of its digits raised to the third power is equal to the number itself.

For example, 371 is an Armstrong number, since

$$3^3 + 7^3 + 1^3 = 371.$$

```

#include<stdio.h>
#include<conio.h>
void main()
{
int a=0,count=0,x,y,z,temp,sum=0; /*variable
initiliazation */

clrscr(); /*to clear screen*/

```

## NOTES

```
while (a<=999)
{
temp=a;

x=temp%10;          //taking out last digit
temp=temp/10;
y=temp%10;
temp=temp/10;
z=temp%10;
temp=temp/10;

sum=(x*x*x)+(y*y*y)+(z*z*z);
if(sum==a)
{
count++;
printf("\nArmstrong number%d=%d",count,a);
}
a++;
}

getch();
}
```

### Output:

```
Armstrong number1=0
Armstrong number2=1
Armstrong number3=153
Armstrong number4=370
Armstrong number5=371
Armstrong number6=407
```

### 42. Write a program to enter a number between 100 and 999 and reverse its digits.

Example : 546 will be 645

Simple logic is to take all the digits in different variable and multiply them with 100,10,1 and add them to get reverse number

X=5

Y=4

Z=6

$(X * 100) + (Y * 10) + (Z * 1)$  to get reverse number



**Program:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=0,x,y,z,temp,rev=0; /*variable initialization */

clrscr();                /*to clear screen*/
A:
printf("Enter number between 100 to 999");
scanf("%d",&a);
if(a<100 ||a>999)
{
printf("Please enter three digit number");
goto A;
}

temp=a;

x=temp%10;              //taking out last digit
temp=temp/10;
y=temp%10;
temp=temp/10;
z=temp%10;
temp=temp/10;

rev=(x*100)+(y*10)+(z*1);
printf("\nNumber =%d \t Reverse number=%d",a,rev);

getch();
}
```

**Output:**

```
Enter number between 100 to 999
645

Number =645      Reverse number=546_
```

**NOTES**

**43. Write a program to enter three digit number and check whether number is palindrome or not. Palindrome is the number which will be same on reversing it.**

**NOTES**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=0,x,y,z,temp,rev=0; /*variable initialization
*/

clrscr(); /*to clear screen*/
A:
printf("Enter number between 100 to 999");
scanf("%d",&a);
if(a<100 ||a>999)
{
printf("Please enter three digit number");
goto A;
}

temp=a;

x=temp%10; //taking out last digit
temp=temp/10;
y=temp%10;
temp=temp/10;
z=temp%10;
temp=temp/10;

rev=(x*100)+(y*10)+(z*1);
if(a==rev)
printf("\nNumber =%d is palindrome",a);
else
printf("\nNumber =%d is not a palindrome",a);

getch();
}
```

**Output:**

In case of not a palindrome

```
Enter number between 100 to 999
100

Number =100 is not a palindrome_
```

In case of palindrome

```
Enter number between 100 to 999
818

Number =818 is palindrome
```

**NOTES****44. Write a program to display sum of first 10 natural numbers.**

**Sum = 1+2+3+4+5+6+7+8+9+10**

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n=1,sum=0;    /*variable initialization */

clrscr();        /*to clear screen*/

while(n<=20)
{
sum=sum+n;
n++;
}
printf("Sum of first 10 natural number is =%d",sum);

getch();
}
```

**Output:**

```
Sum of first 10 natural number is =210
```

**45. Write a program to find the factorial of a number.**

**Factorial of 5 =5\*4\*3\*2\*1 i.e. 120**

```
#include<stdio.h>
#include<conio.h>
void main()
```

## NOTES

```
{
int n,fact=1,temp;      /*variable initialization */

clrscr();              /*to clear screen*/
printf("Enter the number to take out factorial");
scanf("%d",&n);
temp=1;
while(temp<=n)
{
fact=fact*temp;
temp++;

}
printf("Factorial of a number %d is %d",n,fact);

getch();
}
```

### Output:

```
Enter the number to take out factorial
5
Factorial of a number 5 is 120
```

### Try yourself:

- (i) Write a program to find number of digits in a number.
- (ii) Write a program to find frequency of each digit in a number.
- (iii) Write a program to find the product of digits in a number.
- (iv) Write a program to find sum of first and last digit.
- (v) Write a program to print sum of first ten even numbers.
- (vi) Write a program to print product of first and last digit.
- (vii) Write a program to display first ten natural numbers from n to 1(descending order).

**Note:** do while loop will be executed at least once, whether the condition is true or false. In this first the statements are executed and then the condition is checked.

**46. Write a program to print table of any number entered by user upto n number of times.**

*Example:* Table of : 7

Upto how many times : 15

Then the table of 7 will be displayed upto  $7 \times 15 = 105$

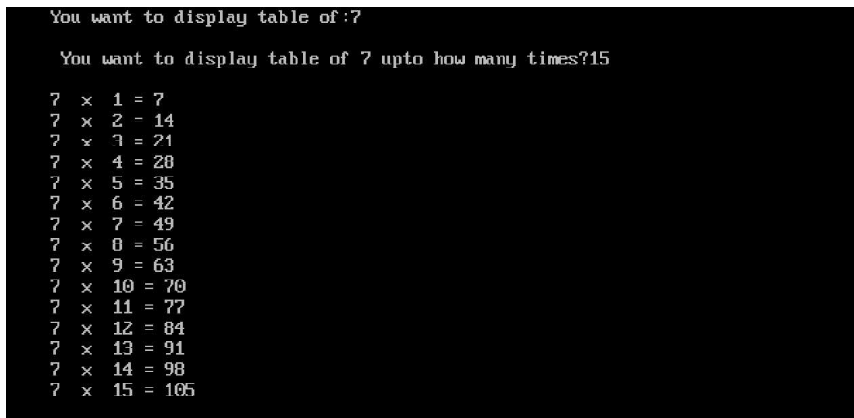
In this loop will be executed according to the number entered by user and that's why the condition is checked at last and the value is entered by user

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,t,x=1; /*variable initialization */

clrscr(); /*to clear screen*/
printf("You want to display table of:");
scanf("%d",&n);
printf("\n You want to display table of %d upto how many
times?",n);
scanf("%d",&t);
do
{
printf("\n%d x %d = %d",n,x,n*x);
x++;
}while (x<=t);

getch();
}
```

**Output:**



```
You want to display table of:7
You want to display table of 7 upto how many times?15
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
7 x 11 = 77
7 x 12 = 84
7 x 13 = 91
7 x 14 = 98
7 x 15 = 105
```

**NOTES**

**NOTES****47. Write a program to print natural numbers upto n times.**

```

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

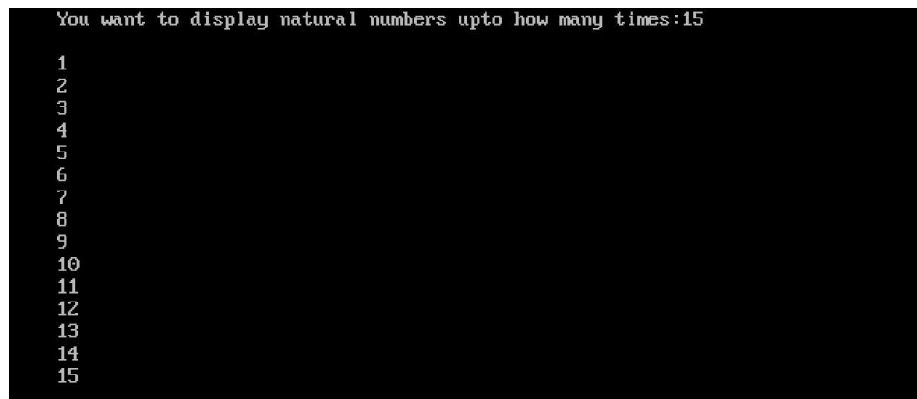
void main()
{
int n,x=1;    /*variable initialization */

clrscr();    /*to clear screen*/
printf("You want to display natural numbers upto how
many times:");
scanf("%d",&n);

do
{
printf("\n%d",x);
x++;
}while (x<=n);

getch();
}

```

**Output:**


```

You want to display natural numbers upto how many times:15
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```

**48. Write a program to print a random number till user want to print. In this program user is going to generate random numbers till user press 'Y' to continue.**

rand ( ) function is used in C to generate random numbers. If we generate a sequence of random number with rand ( ) function, it will create the same sequence again and again every time program runs. Say if we are generating 5 random numbers in C with the help of rand ( ) in a loop, then every time we compile and run the program our output must be the same sequence of numbers.

rand () function is present in <stdlib.h>

**Program:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char ch;          /*variable declaration */
    int x=1,n;

    clrscr ();       /*to clear screen*/

    do
    {
        printf ("\nYour %d random number is %d",x,rand());
        x++;
        printf ("\n\nDo you want to continue generating random
number,Press 'Y' or 'N':") ;
        fflush(stdin);
        scanf("%c",&ch);
    }while (ch=='Y' || ch=='y');

    getch();
}
```

**Output:**

```
Your 1 random number is 346
Do you want to continue generating random number,Press 'Y' or 'N':y
Your 2 random number is 130
Do you want to continue generating random number,Press 'Y' or 'N':y
Your 3 random number is 10982
Do you want to continue generating random number,Press 'Y' or 'N':Y
Your 4 random number is 1090
Do you want to continue generating random number,Press 'Y' or 'N':n
-
```

**49. Write a program to find out the number of digits in a number entered by user.**

```
#include<stdio.h>
#include<conio.h>
void main()
```

**NOTES**

## NOTES

```
{
int x,count=0,temp;      /*variable declaration */

clrscr();                /*to clear screen*/

printf("Enter the number to find out number of
digits ");
scanf("%d",&x);
temp=x;
do
{
temp=temp/10;
count++;
}
while (temp>0);
printf("This number contain %d digits",count);

getch();
}
```

### Output:

```
Enter the number to find out number of digits 3456
This number contain 4 digits
```

### 50. Predict the output of the following program.

```
Program
#include<stdio.h>
#include<conio.h>
void main()
{
int i=10,j=20;          /*variable declaration */
clrscr();              /*to clear screen*/

do
{
printf("\ni=%d\tj=%d",i++,j++);
i++;
j--;
}
while (i<=20);
```



```
getch();
}
```

**Try yourself:**

(i) Write a program to generate the following output:

10 20 40 80 160 320

**NOTES**

**51. Predict the output of the following program.**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i=1,j=2;          /*variable declaration */
clrscr();           /*to clear screen*/
do
{
printf("\ni=%d\tj=%d",++i,-j);
j--;
}
while (i<=10);
getch();
}
```

**Try yourself:**

Write a program to generate following output.

1 4 9 16 25 36 49 72 81 100

**52. Write the outout of following program :**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i=1;
while(i<=10)
printf("%d",++i);
getch();
}
```

**NOTES****53. Write the output for the following program :**

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

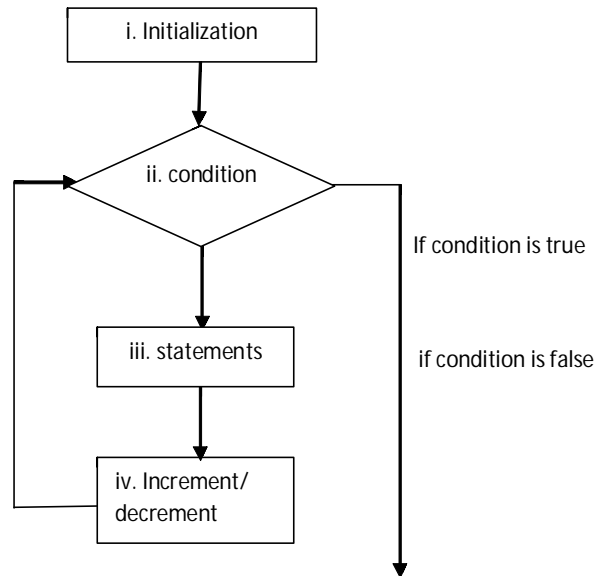
void main()
{
int i=1,j=2;      /*variable declaration */
clrscr();        /*to clear screen*/
do
{
printf("\ni=%d\tj=%d",i++,j++);
j++;
}
while (i<=10,j<=10);
getch();
}
```

**54. Write the output of the following program.**

```
Program
#include<stdio.h>
#include<conio.h>
void main()
{
int i=1;      /*variable declaration */
clrscr();    /*to clear screen*/
do
{
printf("\nOutput =%d",i * ++i);
i++;
}
while (i<=10);
getch();
}
```

**For loop**

For loop is the loop which contain initialization , condition checking and increment and decerment in one single statement .Steps of for loop is shown in the form of flowchart:

**NOTES**

**Note:** If the increment is pre increment or pre decrement then , step iv. Will be executed (increment/decrement) before step iii.(statements)

**Syntax:**

```

for (initialization; condition ; increment /decrement )
{
Statements;
}

```

**Example:**

```

for( int x=1; x<=10;x++)
{
printf("\t%d",x);
}

```

**Output:**

1 2 3 4 5 6 7 8 9 10

**55. Write a program to print first 10 odd numbers on screen.**

```

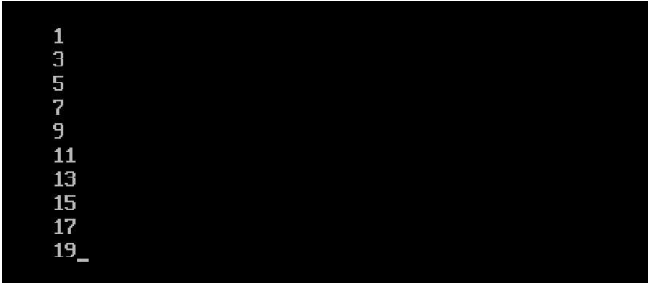
#include<stdio.h>
#include<conio.h>
void main()
{
int i;          /*variable declaration */
clrscr();      /*to clear screen*/
for (i=1;i<=19;i=i+2)
{
printf("\n%d",i);
}
}

```

```
}  
getch();  
}
```

## NOTES

### Output:



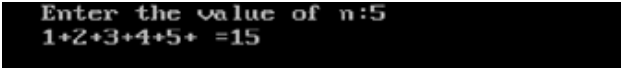
```
1  
3  
5  
7  
9  
11  
13  
15  
17  
19_
```

### 56. Write a program to enter a number (n) and find out sum of series upto n times.

Sum= 1+2 +3+4+5 +6.....n

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int n,sum=0,i;          /*variable declaration */  
clrscr();              /*to clear screen*/  
printf("Enter the value of n:");  
scanf("%d",&n);  
  
for (i=1;i<=n;i++)  
{  
printf("%d+",i);  
sum=sum+i;  
}  
printf(" =%d",sum);  
getch();  
}
```

### Output:



```
Enter the value of n:5  
1+2+3+4+5+ =15
```

**57. Write a program to print Fibonacci series upto n using for loop.**

```

#include<stdio.h>
#include<conio.h>
void main()
{
int a=0,b=1,c,n;          /*variable initialization */

clrscr();                /*to clear screen*/
printf("Enter the value of n for number of term in
Fabonacci series");
scanf("%d",&n);
printf("\t%d\t%d",a,b);  // as we displayed two terms
                        so n>=3

for (;n>=3;n-)
{
c=a+b;
printf("\t%d",c);
a=b;
b=c;
}

getch();
}

```

**NOTES****Output:**

```

Enter the value of n for number of term in Fabonacci series9
0      1      1      2      3      5      8      13      21

```

**58. Write a program to print factorial of a number using for loop.**

```

#include<stdio.h>
#include<conio.h>
void main()
{
int n,fact=1,temp;      /*variable initialization */
clrscr();              /*to clear screen*/
printf("Enter the number to take out factorial");
scanf("%d",&n);
temp=1;
for(temp=1;temp<=n;temp++)
{

```

**NOTES**

```

fact=fact*temp;
}
printf("Factorial of a number %d is %d",n,fact);
getch();
}

```

**Output:**

```

Enter the number to take out factorial6
Factorial of a number 6 is 720

```

**59. Write a program to enter a number and reverse its digits.**

```

#include<stdio.h>

#include<conio.h>
void main()
{
int n,temp,revnum,remainder; /*variable declaration
*/
clrscr(); /*to clear screen*/
printf("Enter the number to reverse it ");
scanf("%d",&n);
for(temp=n,revnum=0;temp!=0;temp=temp/10)
{
remainder=temp%10;
revnum=revnum*10+remainder;
}
printf("Original number=%d and reverse
number=%d",n,revnum);
getch();
}

```

**Output:**

```

Enter the number to reverse it 1234
Original number=1234 and reverse number=4321_

```

**60. Write the output of the following program :**

```

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

```

```

int i=1,j,k;
i=1;
j=1;
clrscr();
for(k=1;k<=3;k++)
{
printf("\n%d\t%d\t%d",i,j,k);
}
getch();
}

```

## NOTES

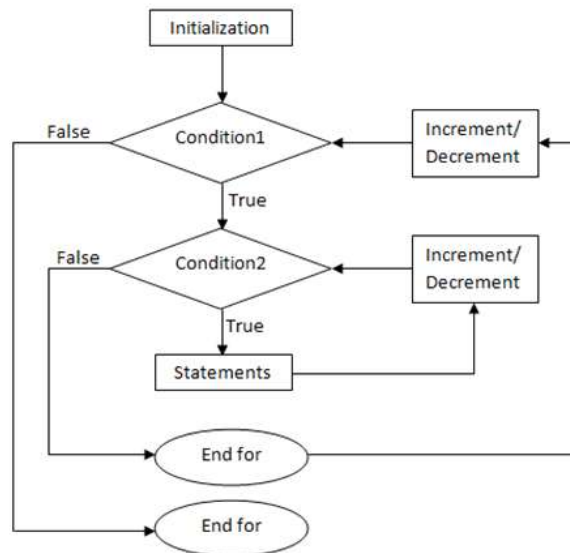
### Nested loop

Loop within a loop is called nested loop.

Generally we use nested loop using for loop, as it is short to write and easy to understand

Nested loop has more than one loop. The first loop is known as outer loop. The loop within the loop is known as inner loop.

Nested loop working is explained through flowchart



**61. Write a program to print the pattern on screen using nested loop.**

```

1
12
123
1234
12345

```

**NOTES**

In the program outer loop is for number of rows(5)  
Inner loop is used to print elements in one col

If we observe the pattern , when row=1, then col=1(1)  
When row=2 , then col =2 (1,2)  
When row=3, then col=3(1,2,3)  
When row=4, then col=4(1,2,3,4)  
When row=5, then col=5(1,2,3,4,5)

In each program of pattern first you establish relation between row and col. Number of rows will be executed in outer loop. And number of columns in inner loop.

Now observe the program.

Once you will be able to establish relation then you can make any program. These pattern are used to learn programming concept. These concept will be implemented in programming like matrix multiplication, arrays etc. These pattern are very important to learn programming concepts.

**Program:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int row,col;           /*variable declaration
                        */

clrscr();              /*to clear screen*/

for(row=1;row<=5;row++) //outer loop
{
    for (col=1;col<=row;col++) //inner loop
    {
        printf("\t%d",col)
    }
    printf("\n")        //closing of inner loop
                        //to change row
                        otherwise all the
                        elements will be printed
                        on same line

}                       //closing of outer loop

getch();
}
```



**Output:**

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
-
```

**NOTES**

**62. Write a program to print the following pattern on screen**

```
12345
1234
123
12
1
```

Now in this program the scenario is reverse.

When row=1 , then col=5

When row=2 , then col=4

When row=3, then col=3

When row =4, then col=2

When row =5, then col=1

If the scenario is reverse we can do reverse counting i.e. from row 5 to 1.

Now row=5 ,then col=5

Row=4, then col=4

Row=3, then col=3

Row=2, then col=2

Row=1 then col=1

Now observe the program

**Program:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int row,col;          /*variable declaration */

clrscr();             /*to clear screen*/

for(row=5;row>=1;row-) //outer loop
```

**NOTES**

```

{
    for (col=1;col<=row;col++) //inner loop
    {
        printf("\t%d",col);
    } //closing of inner loop

    printf("\n"); //to change row
                //otherwise all the
                //elements will be
                //printed on same line

} //closing of outer loop
getch();
}

```

**Output:**

```

1      2      3      4      5
1      2      3      4
1      2      3
1      2
1
-

```

**63. Write a program to print the following pattern on screen.**

```

*
**
***
****

```

Now in this program the logic is same as in program number 61.

Number of rows = number of columns

Only difference is that, instead of value of column we have to print \*

Now observe the program and compare with program number 61.

**Program:**

```

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

```

```

int row,col;          /*variable declaration */

clrscr();            /*to clear screen*/

for(row=1;row<=4;row++) //outer loop
{
    for (col=1;col<=row;col++) //inner loop
    {
        printf("\t *");
    }
    printf("\n");
}
getch();
}

```

**NOTES****Output:**


```

*
*  *
*  *  *
*  *  *  *

```

**64. Write a program to print the following pattern on screen.**

```

10  8  6  4  2
10  8  6  4
10  8  6
10  8
10

```

This program is similar to program number 62.

The only difference is instead of column number, we have to print even number in reverse order.

Now observe the program with two different logic and same output

**Logic 1 program:**

```

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

```

**NOTES**

```

int row,col,n;          /*variable declaration */
clrscr();              /*to clear screen*/
for(row=5;row>=1;row-) //outer loop
{
    for (col=1,n=10;col<=row;col++,n-=2)
        //inner loop
    {
        printf("\t%d",n);
    }
    printf("\n");      //closing of inner loop
                        //to change row otherwise all
                        the elements will be printed
                        on same line
}                      //closing of outer loop
getch();
}

```

**Output:**

```

10  8  6  4  2
10  8  6  4
10  8  6
10  8
10

```

**Logic 2 Program:**

```

#include<stdio.h>
#include<conio.h>
void main()
{
int row,col,n;          /*variable declaration */
clrscr();              /*to clear screen*/
for(row=2;row<=10;row=row+2) //outer loop
{
for (col=10;col>=row;col=col-2)//inner loop
{
    printf("\t%d",col);
}
    printf("\n");      //closing of inner loop
                        //to change row otherwise
                        all the elements will be
                        printed on same line
}
}

```

```

} //closing of outer loop
getch();
}

```

**Output:**

```

10 8 6 4 2
10 8 6 4
10 8 6
10 8
10

```

**NOTES**

Now it depend on totally your logic to make program.

Brain booster:- You can try out for any third logic.

**65. Write a program to print the following pattern on screen.**

```

*
**
***
****
*****
*****
****
***
**
*

```

In this pattern if you combine the two programs with two different nested loop then you get the following pattern.

**Program:**

```

#include<stdio.h>
#include<conio.h>
void main()
{
int row,col,n; //variable declaration */

clrscr(); //to clear screen*/
//loop 1
for(row=1;row<=5;row++) //outer loop
{
for (col=1;col<=row;col++ //inner loop
{
printf("\t*");

```

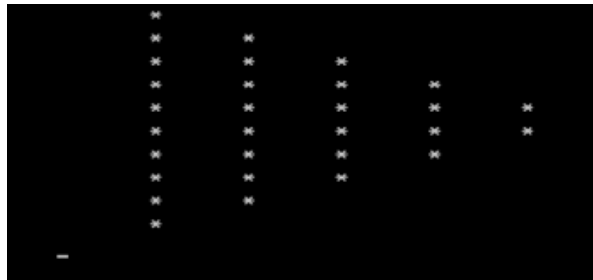
**NOTES**

```
    } //closing of inner loop
    printf("\n"); //to change row otherwise
                    //all the elements will be
                    //printed on same line
    } //closing of outer loop
    //loop2

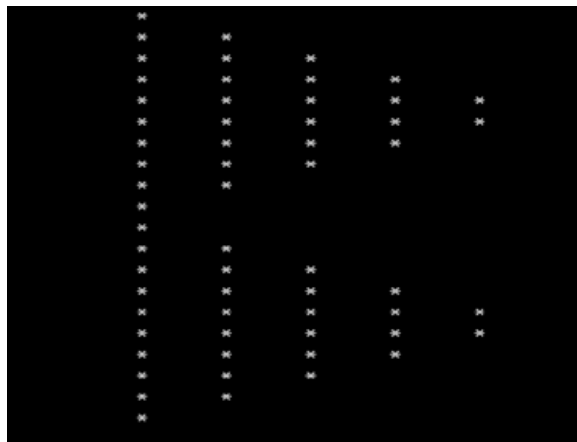
for(row=5;row>=1;row-);
{
    for (col=1;col<=row;col++)
    {
        printf("\t*");
    }
    printf("\n");
}

getch();
}
```

**Output:**



**66. Write a program to print the above output pattern on screen for two times.**



**Program:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int row,col,n;          /*variable declaration */

clrscr();              /*to clear screen*/

for(n=1;n<=2;n++)
{

for(row=1;row<=5;row++) //outer loop
{

for (col=1;col<=row;col++)//inner loop
{
printf("\t*");
} //closing of inner loop
printf("\n"); //to change row otherwise
all the elements will be
printed on same line
} //closing of outer loop

for(row=5;row>=1;row-)
{
for (col=1;col<=row;col++)
{
printf("\t*");
}
printf("\n");
}
}
getch();
}
```

**NOTES**

**NOTES****Try yourself:**

(i) Write a program to print the following pattern on screen

```

5  4  3  2  1
4  3  2  1
3  2  1
2  1
1

```

(ii) Write a program to print the following pattern on screen

```

1
2  2
3  3  3
4  4  4  4
5  5  5  5  5

```

**67. Write a program to print sum of the following series**

$$\text{Sum} = 1/1! + 2/2! + 3/3! + 4/4! + 5/5!$$

In this program we have to take float for result.

```

#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,fact;
float sum=0;      /*variable declaration */

clrscr();        /*to clear screen*/

for(i=1;i<=5;i++)
{
for(fact=1,j=i;j>=1;j--)
{
fact=fact*j;
}
sum=sum+(float)i/fact;
}
printf("Sum of series =%f",sum);
getch();
}

```

**Output:**

```
Sum of series =2.708333
```



## BLOCK 2 FUNCTIONS, ARRAYS AND STRINGS

### NOTES

#### Arrays

- Arrays can be defined as group of similar data type accessed by one variable.
- Items can be accessed by using subscript.
- A one-dimensional array is like a list
- A two dimensional array is like a table or matrices.
- Multidimensional arrays have no limits on the number of dimensions in an array.
- Arrays can be declared as :

Data type variablename[n];

Where variable name is array name and n is the number of elements in an array.

*Example:*

```
int a[10];
```

- Array is actually a pointer (address variable).
- Array can be initialized at the time of declaration by using { } :

*Example:* `int a[5]={1,2,3,4,5};`

```
float b[5]={1.5, 2.5, 3.5, 4.5, 5.5};
```

- Array can also be initialized by user at run time.
- Arrays can also be created dynamically by using `calloc ()` function at run time.
- Array store its first element at [0] subscript.
- Arrays looks like

	A[0]	A[1]	A[2]	A[3]	A[4]
A	10	20	30	40	50
Address	1001	1003	1005	1007	1009

1. Write a program to declare an array of five elements and display it on screen.

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

**NOTES**

```
int a[5],i;           //array declaration

clrscr();           /*to clear screen*/

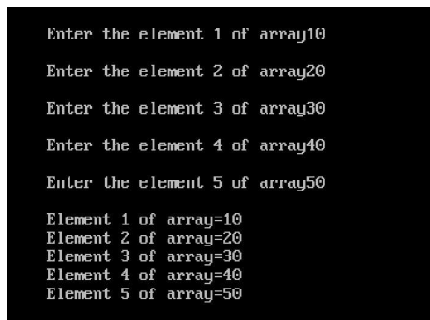
for(i=0;i<=4;i++)   //array initilazation
{
printf("\nEnter the element %d of array",i+1);
scanf("%d",&a[i]);
}

for(i=0;i<=4;i++)   //array display
{
printf("\nElement %d of array=%d",i+1,a[i]);
}

getch();

}
```

**Output:**



**2. Write a program to enter an array of 10 elements and display its sum.**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[10],i,sum=0;    //array declaration
clrscr();    /*to clear screen*/

for(i=0;i<=9;i++)   //array initilazation
{
printf("\nEnter the element %d of array",i+1);
scanf("%d",&a[i]);
```

```

sum=sum+a[i];
}
printf("\nSum of all elements= %d ",sum);
getch();
}

```

**Output:**

```

Enter the element 1 of array10
Enter the element 2 of array20
Enter the element 3 of array30
Enter the element 4 of array40
Enter the element 5 of array50
Enter the element 6 of array60
Enter the element 7 of array70
Enter the element 8 of array80
Enter the element 9 of array90
Enter the element 10 of array100
Sum of all elements= 550 _

```

**NOTES****3. Write a program to enter five elements of an array and display them in reverse order.**

```

#include<stdio.h>
#include<conio.h>
void main()
{
int a[5],i;           //array declaration

clrscr();           /*to clear screen*/

for(i=0;i<=4;i++)   //array initialization
{
printf("\nEnter the element %d of array",i+1);
scanf("%d",&a[i]);
}

for(i=4;i>=0;i-)
{
printf("\nElement %d of array =%d",i+1,a[i]);
}
getch();
}

```

## NOTES

```
Enter the element 1 of array1
Enter the element 2 of array2
Enter the element 3 of array3
Enter the element 4 of array4
Enter the element 5 of array5

Element 5 of array =5
Element 4 of array =4
Element 3 of array =3
Element 2 of array =2
Element 1 of array =1_
```

### 4. Write a program to sort an array of 20 elements.

**Note:** Sorting will use swapping logic

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[10],i,j,temp;      //array declaration

clrscr();               /*to clear screen*/

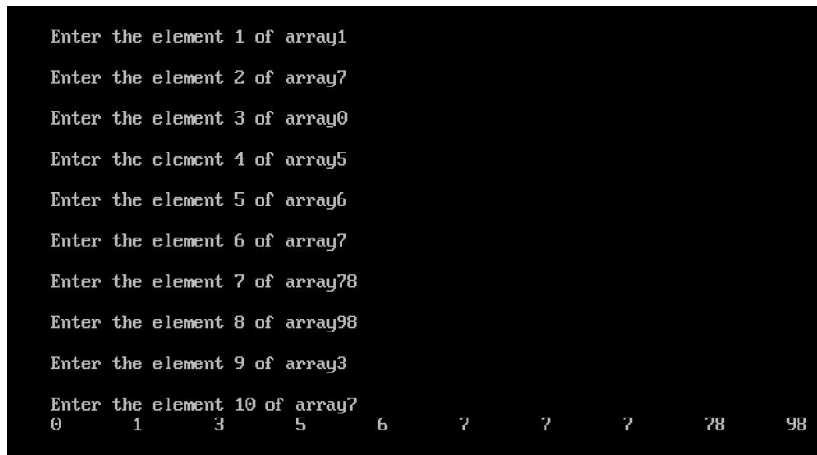
for(i=0;i<=9;i++)      //array initialization
{
printf("\nEnter the element %d of array",i+1);
scanf("%d",&a[i]);
}

for(i=0;i<=9;i++)      //sorting
{
for(j=0;j<=9;j++)
{
if(a[i]<a[j])
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
}
```

```
for(i=0;i<=9;i++)
{
printf("%d\t",a[i]);
}
getch();
}
```

**NOTES**

**Output:**



**Try yourself:**

- (i) Write a program to enter 5 elements in an array and swap first and last item and then display array.
- (ii) Write a program to display the size of array using sizeof().

*Hint: n=sizeof(a)/sizeof(int);*

sizeof() function is present in stdio.h. It will give you the total memory size occupied by variable and then divide by sizeof each data type.

If array contain 5 elements then sizeof(function returns 10, then 10/2=5 are the number of elements .

- (iii) Write a program to create array of 10 elements and replace all the prime numbers by 0.

Note: This program requires nested loop , one for array and other for checking prime number.

- (iv) Write a program to initialize an array of 20 elements and replace all the even number by 0 and all the odd number by 1.

- (v) Write a program to search a number in an array of size 10 and also display the position of array element if found otherwise display the message "not found".

**NOTES**

- Two dimensional array is used to represent matrices.
- It is divided into rows and columns
- It is declared as datatype variable[n][m];

*Example:* int A[2][3];

This creates array of 2 rows and 3 columns.

A[0][0]	A[0][1]	A[0][2]
A[1][0]	A[1][1]	A[1][2]

- It can be initialized at the time of declaration like this :

```
int A [2][3]={  
    {10, 20, 30}  
    {100,200,300}  
};
```

Or

```
int A[2][3]={10,20,30,100,200,300};
```

A[0][0] 10	A[0][1] 20	A[0][2] 30
A[1][0] 100	A[1][1] 200	A[1][2] 300

- It can be created dynamically at run time by using calloc () function.
- To enter the values of two dimensional array we use nested loop.
- Valid and invalid declaration of c

```
int abc[2][2] = {1, 2, 3 ,4 } /* Valid declaration*/  
int abc[][2] = {1, 2, 3 ,4 } /* Valid declaration*/  
int abc[][] = {1, 2, 3 ,4 } /*invalid declaration*/  
int abc[2][] = {1, 2, 3 ,4 } /*invalid declaration */
```

**5. Write a program to create and initialize 3 × 3 array and display it.**

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int a[3][3],i,j;           //array declaration  
  
clrscr();                 /*to clear screen*/
```

```

for(i=0;i<=2;i++)          //array initialization
{
    for(j=0;j<=2;j++)
    {
        printf("\nEnter the element[%d][%d] of
array",i,j);
        scanf("%d",&a[i][j]);
    }
}
for(i=0;i<=2;i++)
{
    for(j=0;j<=2;j++)
    {
        printf("\t%d",a[i][j]);
    }
printf("\n");
}

getch();
}

```

**NOTES**

**Output:**

```

Enter the element[0][0] of array1
Enter the element[0][1] of array2
Enter the element[0][2] of array3
Enter the element[1][0] of array10
Enter the element[1][1] of array20
Enter the element[1][2] of array30
Enter the element[2][0] of array100
Enter the element[2][1] of array200
Enter the element[2][2] of array300
  1   2   3
 10  20  30
 100 200 300

```

**6. Write a program to perform matrix addition on 2 × 2 matrix**

Example

First Matrix	+	Second Matrix	=	third
matrix				
3    4		1    4		4    8
1    2		8    3		9    5

**NOTES**

**Program**

```

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

void main()
{
int a[2][2],b[2][2],c[2][2],i,j;
                                //array declaration

clrscr();                        /*to clear screen*/

for(i=0;i<=1;i++)                //array initialization
{
    for(j=0;j<=1;j++)
    {
        printf("\nEnter the element a[%d][%d] of array",i,j);
        scanf("%d",&a[i][j]);
    }
}
for(i=0;i<=1;i++)                //array initialization
{
    for(j=0;j<=1;j++)
    {
        printf("\nEnter the element b[%d][%d] of array",i,j);
        scanf("%d",&b[i][j]);
    }
}
for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
        c[i][j]=a[i][j]+b[i][j];
    }
}
printf("\nA=\n");
for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)

```



```

    {
    printf("\t%d",a[i][j]);
    }
    printf("\n");
}
printf("\nB=\n");
for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
    printf("\t%d",b[i][j]);
    }
printf("\n");
}
printf("\nC=\n");
for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
    printf("\t%d",c[i][j]);
    }
printf("\n");
}

getch();
}

```

**Output:**

```

Enter the element a[1][0] of array1
Enter the element a[1][1] of array1
Enter the element b[0][0] of array2
Enter the element b[0][1] of array2
Enter the element b[1][0] of array2
Enter the element b[1][1] of array2

A=
    1    1
    1    1

B=
    2    2
    2    2

C=
    3    3
    3    3

```

**NOTES**

**7. Write a program for matrix multiplication for  $2 \times 2$  matrix****NOTES**

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

void main()
{
int a[2][2],b[2][2],c[2][2],i,j,k,sum=0;

//array declaration

clrscr(); //to clear screen*/

for(i=0;i<=1;i++) //array initilazation
{
for(j=0;j<=1;j++)
{
printf("\nEnter the element a[%d][%d] of
array",i,j);

scanf("%d",&a[i][j]);
}
}
for(i=0;i<=1;i++) //array initilazation
{
for(j=0;j<=1;j++)
{
printf("\nEnter the element b[%d][%d] of
array",i,j);
scanf("%d",&b[i][j]);
}
}
for(i=0;i<=1;i++) //multiplication
{
for(j=0;j<=1;j++)
{
for (k=0,sum=0;k<=1;k++)
{
sum=sum+a[i][k]*b[k][j];
}
}
}
}
```

```

        c[i][j]=sum;
    }
}

printf("\nA=\n");
for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
        printf("\t%d",a[i][j]);
    }
    printf("\n");
}

printf("\nB=\n");
for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
        printf("\t%d",b[i][j]);
    }
}
printf("\n");

printf("\nC=\n");
for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
        printf("\t%d",c[i][j]);
    }
}
printf("\n");
}
getch();
}

```

## NOTES

## NOTES

```
Enter the element a[1][0] of array3
Enter the element a[1][1] of array1
Enter the element b[0][0] of array2
Enter the element b[0][1] of array0
Enter the element b[1][0] of array1
Enter the element b[1][1] of array2

A=
  1  2
  3  4

B=
  2  0
  1  2

C=
  4  4
  10 0
```

### Try yourself:

- (i) Write a program to enter  $3 \times 3$  matrix and replace all the even number by 0 and odd number by 1. The matrix you get is sparse matrix.
- (ii) Write a program to arrange the elements of  $2 \times 2$  matrix in descending order.
- (iii) Write a program to find maximum and minimum of array from  $4 \times 4$  matrix.
- (iv) Write a program in c to find the transpose of matrix (transpose is row will be converted as columns and columns will be considered as rows)

### String Handling in c

- String is an array of characters.
- At last string contain NULL character('\0') for ending the string
- In c language <string.h> header file is used for string handling.
- It contains some of the common functions like

```
strcpy() - to copy string
strcat() - to concatenate string
strlen()- to display string length
strcmp()- to compare strings
```

- Different ways of declaring string are:

```
char str1[20]; // Character array
char str2[20] = { 'h', 'e', 'l', 'l', 'o', '\0' };
// Array initialization

char str3[20] = "hello"; // Shortcut array
// initialization
```

```
char str4[20] = ""; // Empty or null C string
                    of length 0, equal to ""
```

*Functions, Arrays and Strings*

We can also declare a C string as a pointer to a char:

```
char* str5 = "hello";
```

- To read string use: gets(str); function
- To display string use: puts(str); function

### 8. Write a program in C language to read a string and display it.

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
char str[30];
printf("Enter String ");
gets(str);
printf("\nString is ");
puts(str);
getch();
}
```

#### Output:

```
Enter String computer science
String is computer science
```

### 9. Write a program to copy string from str1 to str2.

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

void main()
{
char str1[30],str2[30];
clrscr();
printf("Enter String ");
gets(str1);
strcpy(str2,str1);
printf("\nCopied String is ");
```

## NOTES

```
puts(str2);  
getch();  
}
```

## NOTES

### Output:

```
Enter String Information Technology  
Copied String is Information Technology  
_
```

**10. Write a program to enter two strings, compare them and display the message whether the strings are equal or not.**

```
#include<stdio.h>  
#include<conio.h>  
#include<string.h>  
void main()  
{  
char str1[30],str2[30];  
clrscr();  
printf("Enter String1 ");  
gets(str1);  
printf("Enter String2");  
gets(str2);  
if (strcmp(str1,str2)==1)  
{  
printf("Both the strings are same");  
}  
else  
{  
printf("String are not same");  
}  
getch();  
}
```

### Output:

```
Enter String1 Computer  
Enter String2 Science  
String are not same
```

**11. Write a program to enter a string and find out whether the string is palindrome or not.** *Functions, Arrays and Strings*

In this we use `strrev(str);` function to reverse the string.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char str1[30],str2[30];
clrscr();
printf("Enter String1  ");
gets(str1);

strcpy(str2,strrev(str1)); //reverse the string and
                           copy it to str2

if (strcmp(str1,str2)==0) //compare two strings
{
printf("Both the strings are Palindrome");
}
else
{
printf("String are not palindrome");
}
getch();
}
```

**NOTES**

**Output:**

```
Enter String1 madam
madam
Both the strings are Palindrome
```

**Try yourself:**

- (i) Write a program to find the length of string using `strlen()` function.
- (ii) Write a program to concatenate the two strings entered by user.
- (iii) Write a program in c to change the case of letters into opposite .i.e capital will change into small and small will change into capital.

**NOTES**

User defined function are used in structural programming. C is structural programming and it supports user defined functions. When we create functions, it require three things:

- declaration (before definition ),
- definition ,
- function call

Function declaration is used before definition .To declare any function we have to write:

Return type fun\_name (arguments);

- o Return type can be any data type which is returned by function.
- o Function name should be unique, without spaces and it should not be keyword.
- o Arguments are the list of values provided to function for calculation

**Function definition:**

```
Return type fun_name (arguments)
{
    _____;
    _____;
}
```

**Function call:**

Function call is used to call function and after execution of function it come back to the position from where we called the function. Function cannot be executed without call.

```
Fun_name (argument value);
```

User defined functions are of four types:

- (a) Return no value , passes no arguments
- (b) Return no value , passes arguments
- (c) Return value , passes no argument
- (d) Return value , passes arguments

Now consider the same program with four different types:

**12. Write a program to create a function sum with all four types and call it in program.**

```
#include<stdio.h>
#include<conio.h>
void sum1(); //function declaration with no return
```



```

value, no arguments
void sum2(int, int); //function declaration with no
                    return value, passes arguments

int sum3();         //declaration with return value,
                    passes no arguments
int sum4(int, int); //declaration with return value,
                    passes arguments

void main()
{
char ch='y';
int a,b,c,n;
clrscr();

while (ch=='y' ||ch=='Y')
{

printf("\nEnter which function you want to call");
printf("\n1. No return value, no argument");
printf("\n2. No return value, passes arguments");
printf("\n3. Return value, Passes no argument");
printf("\n4. Return value, Passes arguments");
printf("\n Enter your choice form 1-4");
scanf("%d",&n);
if(n==1)
{
sum1();           //function call
printf("return from function ");
}
else if(n==2)
{
printf("Enter two number for addtion");
scanf("%d%d",&a,&b);
sum2(a,b);       //function call
}
else if(n==3)
{
c=sum3();        //function call

```

## NOTES

**NOTES**

```
printf("Sum=%d",c);
}
else if(n==4)
{
printf("Enter two number for addtion");
scanf("%d%d",&a,&b);
c=sum4(a,b); //function call
printf("sum=%d",c);
}
else
{
printf("Invalid choice, Try again");
}
printf("\nDo you want to continue, press 'y' or
'n'");
fflush(stdin);
scanf("%c",&ch);
}
getch();
}

void sum1() //function definition
{
int a,b,c;
printf("Enter two number for addtion");
scanf("%d%d",&a,&b);
c=a+b;
printf("Sum=%d",c);
}

void sum2(int x, int y) //fucntion definition
{
int z;
z=x+y;
printf("Sum=%d",z);
}
```

```

int sum3() //function definition
{
int a,b,c;
printf("Enter two number for addtion");
scanf("%d%d",&a,&b);
c=a+b;
return(c);
}

int sum4(int a, int b) //function definition
{
int c;
c=a+b;
return(c);
}

```

## NOTES

### Output:

```

Enter which function you want to call
1. No return value, no argument
2. No return value, passes arguments
3. Return value, Passes no argument
4. Return value, Passes arguments
Enter your choice form 1-4
1
Enter two number for addition
10
20
Sum=30return from function
Do you want to continue, press 'y' or 'n'y

Enter which function you want to call
1. No return value, no argument
2. No return value, passes arguments
3. Return value, Passes no argument
4. Return value, Passes arguments
Enter your choice form 1-4

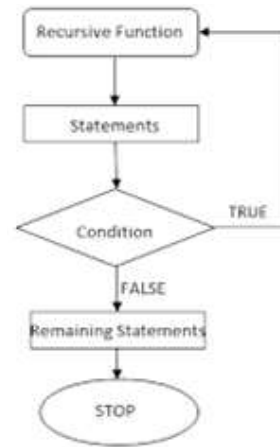
```

### Try yourself:

- (i) Write a function which return the value and passes argument for calculating factorial of a number.
- (ii) Write a function in c which return value but passes no argument for calculating Simple Interest.
- (iii) Write a function in c which return no value, but passes argument for displaying Fibonacci series upto n times.
- (iv) Write a function in c which return no value , but passes argument for finding out number of digit in a number.

**Recursive function:-** Recursive function is a function which call itself. Function which calls itself is called recursive function. The working of recursive function is shown in flowchart.

**NOTES**



**13. Write a recursive function to count\_upto 10.**

```

#include<stdio.h>
#include<conio.h>
void count_uptoten(int count);

void main()
{
    count_uptoten ( 0 );
}

void count_uptoten ( int count )
{
    /* keep counting if we have a value less than ten
    if ( count < 10 )
    {
        count_uptoten( count + 1 );
    }
}
    
```

**14. Write a recursive function to calculate factorial of a number.**

```

#include <stdio.h>
#include<conio.h>
int fact(int n);

void main()
    
```

```

{
    int n;
    printf("Enter the number to take out factorial:
");
    scanf("%d", &n);
    printf("Factorial of number%d is %d", n,
fact(n));
}

int fact(int n)
{
    if (n >= 1)
        return n*fact(n-1);
    else
        return 1;
}

```

## NOTES

### Output:

```

Enter the number to take out factorial6
Factorial of a number 6 is 720_

```

### 15. Write the output of following program:

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n=10;
    printf("%d",n);
    main();
    getch();
}

```

### Try yourself:

- (i) Write recursive function to print your name for 10 times.
- (ii) Write recursive function to display the sum of array.
- (iii) Write recursive function to display sum of n numbers.

---

## BLOCK 3 STRUCTURES AND UNION

---

**NOTES****Structure and Union (User Defined data type)**

C language also provide a facility to create user defined data type using the keyword struct and union. Structure and union are the two data types which can be created in a program. Some structure are predefined as we have used it previously struct date which is present in dos.h header file. If the structure is predefined we can use it in our program using the header file.

- **Defining structure**

```
struct structname
{
Datatype membername;
Datatype membername;
_____ ;
_____ ;
};
```

- **Declaring structure variable**

```
struct structname variablename;
```

- **Accessing structure member**

```
structvariablename.membername;
```

Next program of examples will show how to define structure, declare a structure variable, accessing structure members.

**1. Write a program to create a structure called as employee with the following members:**

Empno, Fname, basicsalary, da, ta, hra, deduction, grosssal, netsal.

Also write function for getinfo () and putinfo()

Calculate da=7%,ta=3500,hra=16%

**Program**

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

struct emp //defining structure
{
int empno;
char fname[30];
float basicsal;
```

```

float da;
float ta;
float hra;
float deduction;
float gross;
float netsal;
};

struct emp getinfo(struct emp); //declaring function
void putinfo(struct emp);      //declaring function

void main()
{
struct emp E;                  //declaring struct
                               //variable

clrscr();
E=getinfo(E);                  //fuction call
putinfo(E);                    //fuction call
getch();
}

struct emp getinfo(struct emp E) //function definition
{
printf("\nEnter emp code");
scanf("%d",&E.empno);
printf("\nEnter emp name");
fflush(stdin);
gets(E.fname);
printf("\nEnter basic sal");
scanf("%f",&E.basicsal);
printf("\nDA ,TA, HRA and gross salis calulated\n");
E.da=E.basicsal*7/100;
E.ta=3500;
E.hra=E.basicsal*16/100;
E.gross=E.basicsal+E.da+E.ta+E.hra;
printf("Enter the loan amoun to be deducted");
scanf("%f",&E.deduction);
E.netsal=E.gross-E.deduction;
return(E);
}

```

**NOTES**

**NOTES**

```

}

void putinfo(struct emp E)
{
printf("\nEmp code=%d",E.empno);
printf("\nEmp name=%s",E.fname);
printf("\nBasic sal=%f",E.basicSal);
printf("\nDA = %f\nTA = %f\nHRA = %f\nGross
sal=%f",E.da,E.ta,E.hra,E.gross);
printf("\nLoan amoun deduction=%f",E.deduction);
printf("\n Net sal=%f",E.netsal);
}

```

**Output:**

```

Enter emp code 101
Enter emp nameShweta Tiwari
Enter basic sal 35000

DA ,TA, HRA and gross salis calculated
Enter the loan amoun to be deducted 10000

Emp code=101
Emp name=Shweta Tiwari
Basic sal=35000.000000
DA=2450.000000
TA=3500.000000
HRA=5600.000000
Gross sal=46550.000000
Loan amoun deduction=10000.000000
Net sal=36550.000000_

```

Next program will show you the example of creating array of structure.

**2. Write a program to create array of structure for emp. Program should enter the info of 5 employees and display it.**

The same above program is modified to illustrate example.

**Program:**

```

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

struct emp //defining structure
{
int empno;
char fname[30];
float basicSal;
float da;
float ta;
float hra;

```



```

float deduction;
float gross;
float netsal;
};
struct emp *getinfo(struct emp E[]);
//declaring function
void putinfo(struct emp E[]); //declaring function
void main()
{
struct emp E[5],*Eptr; //declaring struct
variable

clrscr();
Eptr=getinfo(E); //fucntion call
putinfo(Eptr); //fucntion call
getch();
}

struct emp * getinfo(struct emp E[5])
//function definition
{
int i;
for(i=0;i<=4;i++)
{
printf("\nEnter emp code");
scanf("%d",&E[i].empno);
fflush(stdin);
printf("\nEnter emp name");
gets(E[i].fname);
printf("\nEnter basic sal");
scanf("%f",&E[i].basicsal);
printf("\nDA ,TA, HRA and gross salis calulated\n");
E[i].da=E[i].basicsal*7/100;
E[i].ta=3500;
E[i].hra=E[i].basicsal*16/100;
E[i].gross=E[i].basicsal+E[i].da+E[i].ta+E[i].hra;
printf("Enter the loan amoun to be deducted");
scanf("%f",&E[i].deduction);
E[i].netsal=E[i].gross-E[i].deduction;
}
}

```

**NOTES**

**NOTES**

```

    }
    return(E);
}

void putinfo(struct emp E[5])
{
    int i;
    for(i=0;i<=4;i++)
    {
        printf("\nEmp code=%d",E[i].empno);
        printf("\tEmp name=%s",E[i].fname);
        printf("\nBasic sal=%f",E[i].basicsal);
        printf("\tDA=%f\tTA=%f\tHRA=%f\tGross
        sal=%f",E[i].da,E[i].ta,E[i].hra,E[i].gross);
        printf("\nLoan amoun deduction=%f",E[i].deduction);
        printf("\n Net sal=%f",E[i].netsal);
    }
}

```

The output will be same as of the above program with the five employee input and output.

Consider the nest program where the arguments for structure can be passed seperaetly.

**3. Write a program to initialize structure directly at the time of declaration and print it by using the function putdata (), which accepts the structure elements separately.**

```

#include<stdio.h>
#include<conio.h>
struct student
{
    char name[20];
    int roll_no;
    int marks;
};

void putdata(char name[], int roll_no, int marks);

void main()
{
    struct student stu = {"Ram", 1, 80};
    clrscr();
}

```

```

        putdata(stu.name, stu.roll_no, stu.marks);
        getch();
    }

void putdata(char name[], int roll_no, int marks)
{
    printf("Name: %s\n", name);
    printf("Roll no: %d\n", roll_no);
    printf("Marks: %d\n", marks);
    printf("\n");
}

```

**Output:**

```

Name: Ram
Roll no: 1
Marks: 80

```

**Structure within structure (Nested structure)**

In this example structure is used within a structure and the condition is called as nested structure.

**4. Program to create structure within structure and access their members. Create a structure date for DOB and use it within the structure student.**

```

#include<stdio.h>
#include<conio.h>
struct dob
{
    int dd,mm,yyyy;
};

struct student
{
    char name[30];
    struct dob d;           //structure within structure
};

void main()
{
    struct student s;
    clrscr();
    printf("Enter name");

```

**NOTES**

**NOTES**

```

fflush(stdin);
gets(s.name);
printf("\nEnter dob");
scanf("%d%d%d",&s.d.dd,&s.d.mm,&s.d.yyyy);
printf("Name is %s",s.name);
printf("\nDOB is %d/%d/%d",s.d.dd,s.d.mm,s.d.yyyy);
getch();
}

```

**Output:**

```

Enter name Ananya
Enter dob 15
7
2004
Name is Ananya
DOB is 15/7/2004

```

**Try yourself:**

- (i) Write a program to store and print the roll no., name, age and marks of a student using structures.
- (ii) Write a program to add two distances in inch-feet using structure. The values of the distances is to be taken from the user.
- (iii) Write a program to add, subtract and multiply two complex numbers using structures to function.
- (iv) Write a program to compare two dates entered by user. Make a structure named Date to store the elements day, month and year to store the dates. If the dates are equal, display "Dates are equal" otherwise display "Dates are not equal".

**Union in c language**

A **union** is a special data type available in C that allows to store different data types in the same memory location.

Unions can be useful in many situations where we want to use the same memory for two or more members.

Union is like a structure. In union, all members share the same memory location.

The main difference between union and structure is union occupies less memory space as compared to structure.

```

Example of union
#include <stdio.h>
#include <string.h>
#include<conio.h>

```

```

union userdata {
    int i;
    float f;
    char str[20];
};
void main( ) {
    union userdata d;
    clrscr();
    printf( "Memory size occupied by data : %d Bytes\n",
sizeof(d));
    getch();
}

```

**Output:**

```

Memory size occupied by data : 20 Bytes
-

```

**5. Predict the output for the program.**

```

#include <stdio.h>
#include<conio.h>
union myunion          //union definition
{
    char fname[30];
    float sal;
    int empno;
} U;

struct mystruct
{
    char fname[30];
    float sal;
    int emocode;
} S;

void main()
{
    printf("The size of union = %d", sizeof(U));
    printf("\nThe size of structure = %d", sizeof(S));
    getch();
}

```

**NOTES**

## BLOCK 4 POINTERS

### NOTES

#### Pointers

- Pointer are address variable .
- They are used to store address of variable .
- By using pointers we can change the values stored in variable.
- Pointer are declared by using the following syntax

```
Datatype *pointername;
```

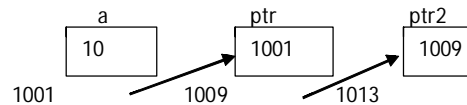
- Pointer uses two operators :

\* read as value stored at

Example: \*ptr is read as value stores at ptr(address)  
& address of

Example: &a is read as address of a

Example of pointer declaration and initialization and pointer handling



```
int *ptr, a, **ptr2 ; //ptr is pointer, a is a variable
                      ,ptr2 is used to store address
                      of pointer
```

```
a=10;
```

```
ptr =& a; //read as ptr is equal to address
         of a
```

```
ptr2=&ptr; //store address of ptr in ptr2
```

```
printf("%d",ptr); //displays address of a i.e. 1001
```

```
printf("%d",*ptr); //displays value stored at ptr
                  i.e. 10
```

```
printf("%d",&a); //displays address of a i.e. 1001
```

```
printf("%d",a) //displays value of a i.e. 10
```

```
*ptr=20; //will initialize a with 20
```

```
printf("%d", a) //will display 20
```

```
*ptr ++; //will increment a or *ptr with
         1
```

```
printf("%d",a) //will display 21
```

```
printf("%d",**ptr2); //will display 21
```

**Note:** The very important things are \*(value stored at) and & (address of ) operator . When you read the operator properly, there is no chance of errors or mistakes in pointer programs.

### Uses of pointer variable

- Arrays are also indirectly the pointer variable which stores the starting address. Pointers are used to handle arrays.
- Strings can also be handled using pointer variable.
- Pointers can also be used for passing arguments to functions.
- When we want to return more than one values from functions than pointers are used.
- Pointers are used to change values of variable from any other function
- Pointers are used to create dynamic data structure or to create variables at run time.
- Programs with pointers run more efficiently and are faster.

Write down the output of following programs of pointers

#### 1. Predict the output of the following program.

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

void main()
{
    int x[] = { 10, 20, 30, 40, 50} ;
    int *ptr;
    ptr = X;
    printf(" %d ", *( ptr + 1 ) );
    getch( );
}
```

#### 2. Predict the output of the following program.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i = 5, *p;
    p = &i;
    printf("%d\n", i * *p * i + *p);
    getch();
}
```

### NOTES

**NOTES****3. Predict the output of the following program.**

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

void main()
{
    int x = 10, *y, *z; // Assume address of x is 1001
                        // and integer is 2 byte size

    y = &x;
    z = y;
    *y++;
    *z++;
    x++;
    printf("x = %d, y = %d, z = %d \n", x, y, z);
    getch();
}
```

**4. Predict the output of the following program.**

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

void main()
{
    int x = 100;
    int *p1, **p2;
    p1 = &x;
    p2 = &p1;
    printf("x = %d, p1 = %d, p2 = %d\n", x, *p1, **p2);
    getch();
}
```

**5. Predict the output of the following program.**

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

void main()
{
    char x[] = { 'A', 'B', 'C', 'D', 'E' };
    char * ptr = &x[0];
    *ptr++;
    printf("%c %c ", *++ptr, -*ptr);
    getch();
}
```



**6. Write a program to access string through pointer.**

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

void main() {

char str[15] = "Hello World!";

char *ptr = str;      //string to pointer

while(*ptr != '\0')  //print string using while loop
{
printf("%c", *ptr);
ptr++;
}

getch();
}
```

**NOTES****7. Write a program to represent array of strings through pointer. Store the strings shown in figure into array of string using pointer.**

	0	1	2	3	4	5	6	7	8	9	10
0	B	h	o	p	a	l	'\0'				
1	N	a	g	p	u	r	'\0'				
2	C	h	e	n	n	a	i	'\0'			

**Program:**

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

void main()
{
char *ptr[3] =          //array of pointer
{
"Bhopal",
"Nagpur",
"Chennai",
};
int i, j;
for (i = 0; i <=2; i++)
{
j = 0;
```

**NOTES**

```

while(*(ptr[i] + j) != '\0')
{
    printf("%c", *(ptr[i] + j));
    j++;
}
printf("\n");
}
getch();
}

```

**8. Write a program to arrange the string in lexicographical order (dictionary order).**

```

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

void main()
{
    int i, j;
    char str[10][20], temp[20];

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

    for(i=0; i<5; ++i)
scanf("%s^\n",str[i]);

    for(i=0; i<5; ++i)
for(j=i+1; j<5 ; ++j)
{
    if(strcmp(str[i], str[j])>0)
    {
        strcpy(temp, str[i]);
        strcpy(str[i], str[j]);
        strcpy(str[j], temp);
    }
}

    printf("\nIn lexicographical order: \n");
    for(i=0; i<10; ++i)
    {
puts(str[i]);
    }
}

```

```

    getch();
}

```

**Output:**

```

C:\TURBODC3\BIN>TC
Enter 5 words:
elephant
deer
monkey
bear
lion

In lexicographical order:
bear
deer
elephant
lion
monkey

```

**NOTES****Dynamic memory allocation:**

When we create variables at run time is called dynamic memory allocation. We use <stdlib.h> header file use the functions which are used in memory allocation. There are two functions used in dynamic memory allocation.

- malloc() - used to allocated memory with the number of bytes required by data type
- calloc()- used to allocate array elements at run time. It initializes array to zero and return the pointer.  
One function to free memory space occupied by malloc(), calloc() or any other variables.
- free()- de allocate the memory space

**Syntax of malloc ()**

```
pointer = (cast-type*) malloc(byte-size);
```

Example

```
ptr = (int*) malloc(10 * sizeof(int));
```

The above statement will allocate either 20 or 40 bytes according to size of int 2 or 4 bytes.

**Syntax of calloc()**

```
pointer = (cast-type*)calloc(n, element-size);
```

Example

```
ptr = (int*) calloc(5, sizeof(int));
```

The above statement allocates continuous space in memory for an array of 5 elements each of size of int, i.e, 2 or 4 bytes.

Syntax of free( )

```
free(pointer);
```

Example

```
free(ptr);
```

It is always a better practice to free the space occupied in our program before ending it.

**9. Write a program to create array of n elements using pointer and malloc () function and display its sum.**

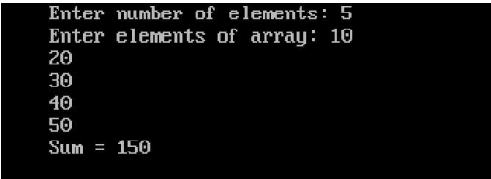
**NOTES**

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
    int n, i, *ptr, sum = 0;
    clrscr();

    printf("Enter number of elements: ");
    scanf("%d", &n);

    ptr = (int*) malloc(n* sizeof(int));
                               //memory allocated using malloc
    if(ptr == NULL)
    {
        printf("Error! not sufficient memory ");
        exit(0);
    }
    printf("Enter elements of array: ");
    for(i = 0; i < n; i++)
    {
        scanf("%d", ptr + i);
        sum = sum+ *(ptr + i);
    }
    printf("Sum = %d", sum);
    free(ptr);
    getch();
}
```

**Output:**

A screenshot of a terminal window showing the output of the program. The text is as follows:

```
Enter number of elements: 5
Enter elements of array: 10
20
30
40
50
Sum = 150
```

**10. Write a program to create array using calloc () function for n number of elements and display them.**

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

void main () {
    int i, n;
    int *p;
    clrscr();
    printf("Enter Number of elements to be entered:");
    scanf("%d",&n);

    p = (int*)calloc(n, sizeof(int));
    printf("%d numbersof elements ",n);

    for( i=0 ; i < n ; i++ )
    {
        scanf("%d",&p[i]);
    }

    printf("The numbers entered are: ");
    for( i=0 ; i < n ; i++ )
    {
        printf("%d ",p[i]);
    }
    free( p );
    getch();
}
```

**Output:**

```
Enter Number of elements to be entered:10
10 numbersof elements 1
2
3
4
5
6
7
8
9
10
The numbers entered are: 1 2 3 4 5 6 7 8 9 10
```

**NOTES**

---

## BLOCK 5 FILES

---

### NOTES

File is a physical place in secondary memory. Variables are created in RAM and files are created in hard disk. The data stored in files are permanent. When the program stop executing all the data is lost and if we want to save it permanently than we have to use files.

C uses some special statements to perform operations on files. In c language we can create two types of files

- Text file
- Binary file

Text file are having extension as .txt file. Text file is less secure and takes huge amount of storage.

### Binary files

Binary files are having .bin files in your computer. Binary files store the data in the form of binary format. They are more secure and store large amount of data.

### File operation

The basic file operations are

- Create file
- Open file
- Read and write in file
- Close file

For any type of file operation first we need to create file pointer. This pointer is used to create, open, read and write and close file. All the functions related with files are present in <stdio.h>

```
FILE *fptr;
```

### Function to open file

```
fptr = fopen("fileopen.ext", "mode")
```

where mode can any one of the following

"r"	Opens file in read only mode
"rb"	Opens file in read only in binary format
"w"	Opens file in write mode
"wb"	Opens file in binary write mode
"a"	Opens files in append mode, add at last
"ab"	Opens files in append mode , binary format
"r+"	Opens file in read and write mode
"rb+"	Opens files in read and write mode in binary
"w+"	Open in both read and write mode
"wb+"	Open in both read and write mode in binary mode

**Function to close file**

```
fclose(fptr);
```

**Function to put string in file**

```
fputs("string ...", fptr);
```

**Function to get string from file**

```
char *fgets(char *str, int n, FILE *stream)
```

Where fgets returns file pointer, str is the string to get values from file, n is the maximum number of character and file \* stream is file pointer.

```
Example      fgets(str, 50, fptr);
```

Will get the maximum 50 characters from file in str string.

**Function for file pointer positioning**

```
fseek (fptr, n, position);
```

where fptr is file pointer, n is the number of character from position , and position can be any one out of three

```
SEEK_SET/ SEEK_CUR / SEEK_END
```

SEEK\_Set is used for beginning

SEEK\_CUR is used for current position

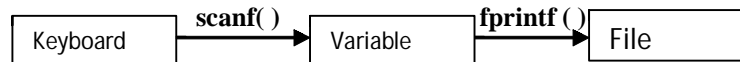
SEEK\_END is sued for ending position

Example

```
fseek(fptr, 10, SET_CUR)
```

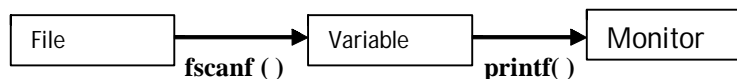
file pointer is set after 10 characters form current position

How to write in file using function from keyboard to file?



Read from keyboard(Input)scanf( )àto variable àvariable to file (output)fprintf( )

How to read from file and display on monitor?



Read from file(input from file )fscanf( )àto variable àoutput to monitor printf( )

**1. Write a program to write in text file using file pointer and functions.**

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

void main()
{
```

**NOTES**

**NOTES**

```

int n; //file pointer
FILE *fptr;
char ch='y';
fptr = fopen("myfile.txt","w"); //open file in
                                write mode
if(fptr == NULL) //check if unable to
                 open file
{
    printf("Error!Not able to open file");
    exit(1);
}

while (ch=='y' || ch=='Y')
{
    printf("Enter num to be written on file : ");
    scanf("%d",&n);

    fprintf(fptr,"%d",n); //write in file
    printf("\nDo you want to write more number");
    fflush(stdin);
    scanf("%c",&ch);
}

fclose(fptr);
printf("file succefully created");
getch();
}

```

**Output:**

```

C:\TURBIDC>
Enter num to be written on file : 1

Do you want to write more number
Enter num to be written on file : 2

Do you want to write more number
Enter num to be written on file : 3

Do you want to write more number
file succefully created_

```

If you will observe the file myfile.txt is created in your current drive.

**2. Write a program to open file, read from file and display it on monitor.**

```

#include <stdio.h>
#include <stdlib.h>
void main()

```



```

{
    int n;                //file pointer
    FILE *fptr;
    clrscr();
    fptr = fopen("myfile.txt","r"); //open file in
                                   write mode
    if(fptr == NULL)        //check if unable to
                           open file
    {
        printf("Error!Not able to open file");
        exit(1);
    }
    fscanf(fptr,"%d",&n);    //read from file
    printf("\n%d",n);        //write on monitor
    fclose(fptr);
    printf("\nfile succefully closed ");
    getch();
}

```

**NOTES****Output:**

```

123
file succefully closed

```

**3. Write a program to append data in myfile.txt and again open the file and display the data.**

```

#include <stdio.h>
#include <stdlib.h>
void main()
{
    int n;                //file pointer
    FILE *fptr;
    char ch='y';
    clrscr();
    fptr = fopen("myfile.txt","a"); //open file in
                                   append mode
    if(fptr == NULL)        //check if unable to
                           open file
    {
        printf("Error!Not able to open file");
    }
}

```

**NOTES**

```

        exit(1);
    }
    while (ch=='y' || ch=='Y')
    {
        printf("Enter number to be append in file ");
        scanf("%d",&n);
        fprintf(fptr,"%d",n);    //write in file
        printf("\nDo you want to add more numbers");
        fflush(stdin);
        scanf("%c",&ch);
    }
    fclose(fptr);
    printf("\nfile succefully append and closed ");
    fptr=fopen("myfile.txt","r");
    fscanf(fptr,"%d",&n);
    printf("%d",n);
    getch();
}

```

**Output:**

```

Enter number to be append in file 4
Do you want to add more numbersy
Enter number to be append in file 5
Do you want to add more numbersn
file succefully append and closed 12345

```

**4. Write a program to show the use of fseek.**

```

#include <stdio.h>

void main () {
    FILE *fptr;

    fptr = fopen("myfile.txt","w+");
    fputs("I learned c langauge", fptr);

    fseek( fptr, 1, SEEK_END );
    fputs(" with lot of fun", fptr);
    fclose(fptr);
    getch();
}

```

**5. Write a program to read file in reverse order and display it.**

```
//Write a program to print reverse content of file
#include <stdio.h>
#include <string.h>
void main()
{
    FILE *fp;

    int count = 0;
    int i = 0;
    char c;
    fp = fopen("myfile.txt","r");
    if( fp == NULL )
    {
        printf("\n File can not be opened : \n");
        exit(1);
    }

    //moves the file pointer to the end.
    fseek(fp,0,SEEK_END);
    //get the position of file pointer.
    count = ftell(fp);

    while( i < count )
    {
        i++;
        fseek(fp,-i,SEEK_END);
        c=fgetc(fp);
        printf("%c",c);
    }
    printf("\n");
    fclose(fp);

    getch();
}
```

**Output:**

```
C:\TURBUC\BIN>TC
nuf fo tol htiw equagnal c denrael I
-
```

**NOTES**

**NOTES****Try yourself:**

- (i) Write a program to open a file "mydata.txt" in write mode and write your name, age and address in file .
- (ii) Write a program to open the file "mydata.txt" in read mode and display the data present in it.
- (iii) Write a program to create file "myinfo.txt" to open file in read and write mode and write information about your hobbies using string and display the information from file.

**Command line argument**

Command line argument are the values passed form command line .These values are called command line arguments. They are important for your program, when we want to control your program from outside instead of hard coding those values inside the code.

The command line arguments are handled using main () function arguments where **argc** refers to the number of arguments passed, and **argv [ ]** is a pointer array which points to each argument passed to the program.

Remember to run command line program form dos shell, by typing the program name and argument.

**Steps to run command line argument program**

- (i) Save file
- (ii) Compile file
- (iii) Press F9 to make file
- (iv) Ctrl+F9 to run file for first time
- (v) Then open dos shell
- (vi) You are in c:\turbo++\bin>
- (vii) Type cd..
- (viii) You will get c:\turbo++>
- (ix) Then type cd source
- (x) You will get c:\turbo++\source> prompt
- (xi) Type the program name and arguments and press enter

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
C:\TURBOC3\BIN>TC
Type EXIT to return to Turbo C++. . .
Welcome to DOSBox v0.74
For a short introduction for new users type: INTRO
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
C:\TURBOC3\BIN>cd..
C:\TURBOC3>cd source
C:\TURBOC3\SOURCE>copyfile myfile1.txt yourfile.txt_

```

## NOTES

### 6. Program to display the number of arguments provided in command line argument.

```

#include <conio.h>
#include <stdio.h>
int main(int argc, char *argv[])
{
    int i;
    clrscr();
    if( argc >= 2 )
    {
        printf("The arguments supplied are:\n");
        for(i = 1; i < argc; i++)
        {
            printf("%s\t", argv[i]);
        }
    }
    else
    {
        printf("argument list is empty.\n");
    }
    return 0;
}

```

This program shows the arguments provided to command line.

**NOTES****7. Write a program in c to copy the one file to another using command line arguments.**

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    FILE *fptr1,*fptr2;
    char ch;

    clrscr();
    if(argc!=3)
    {
        printf("\n Enter Two File Names for copyfile \n");
        exit(1);
    }

    if(argc==3)
    {
        fptr1 = fopen (argv[1],"r");
        fptr2 = fopen (argv[2],"w");
        if(fptr1==NULL )
        {
            puts("\nFile can not be Open...\n");
            exit(1);
        }
        while((ch=getc(fptr1))!=EOF)
        {
            putc(ch,fptr2);
            putchar(ch);
        }
        printf("\n1 File copied \n");
    }
    else
    {
        printf("\n Now goes to File Menu - DOS Shell option
in TurboC++\n ");
    }
}
```

```

        printf("\n and then Enter Two File Names for copy
\n");
        printf("\n Press Enter to continue... \n");
    }

    fclose(fp1);
    fclose(fp2);
    return(1);
}

```

This program will act as copy command of dos and copy the file1 to file2

```

I learned c langauge with lot of fun
1 file copied

C:\TURBOC3\SOURCE>_

```

## NOTES

### 8. Predict the output of the code given below.

If we type `> file good night` as command line arguments

```

#include<stdio.h>

int main(int argc, char *argv[])
{
    printf("%d %s", argc, argv[1]);
    return 0;
}

```

### 9. Write a program to add two numbers in command line arguments.

```

#include<stdlib.h>
int main(int argc, char * argv[]) {
    int i, sum = 0;
    if (argc != 3) {
        printf("Please enter two numbers.");
        exit(1);
    }
    printf("The sum is : ");
    sum= atoi(argv[1])+atoi(argv[2]);
    printf("%d", sum);
    return 0;
}

```

## Instant Programming card for c language

## NOTES

datatype	range	bytes	format
signed char	+128 to -127	1	%c
unsigned char	0 to 255	1	%c
short signed int	-32768 to 32767	2	%d,%i,%o,%x
short unsigned int	0 to 65533	2	%u
long signed int	-2147483648 to 2147483647	4	%ld
long unsigned int	0 to 4294967295	4	%lu
float	-3.4e38 to +3.4e38	4	%f
double	-1.7e308 to +1.7e308	8	%lf
long double	1.77e4932 to 1.7e4932	10	%Lf

command	Shortcut
F1	Help
F2	Save file
F3	Load file
F5	Zoom the current window
F6	Switch between window
F7	Run program in debug mode
F9	Compile and make exe
Alt+F9	Compile and create .obj
Ctrl+F9	Run the program
Ctrl+F7	Watch on variable
Alt+F7	Show the previous error
Alt+F8	Shows the next error
Alt+x	Quit from TC

Operator	Description
()	Function call
[]	Array element
>	Pointer to structure
.	Structure element
-	Negation
++	Increment
--	Decrement
!	(Not)Logical negation
~	One's complement
*	Value stored at (pointer operator)
&	Address of (Pointer operator)
sizeof ( )	Returns size in bytes
*	Multiplication
/	Division
%	Modulus( returns remainder)
+	Addition
-	Subtraction
<<	Bitwise shift left
>>	Bitwise shift right
<	Less than
>	Greater than
<=	Less than equal to
>=	greater than equal to
==	Equal to
!=	Not equal to
&	Bitwise and
	Bitwise or
&&	Logical and
	Logical or
?:	Conditional operator
+= -=	Compound assignment
*= /= %=	Example a+=10;
,	Separation of arguments or variable
=	Assignment

Escape sequence	\r	Carriage return
\b	Backspace	\f
\n	Newline	Form feed
\t	Tabular	backslash
		doublequote

Debugging Tips(Error removing )	
Check for semicolon after every statement ;	
Check for variable name mismatch;	
Check for case sensitive	
Check for parenthesis	
Check for header files	
Check for linker error	
Do not ut semicolon in control statement like if...else, for , while etc	
Check for declaration and definition mismatch	

Statement	Use
goto label;	To reach to defined label
break;	Termination of loop
Continue;	Continue executing loop
return;	Return back from function
Return(exp);	Return back form function with value
while( ){ }	Looping statement
for( )	Looping statement
do while ( );	Looping statement
if( )	Contioanal statement
If( )...else	Conditional statement
switch...case	Conditional statement
( )?...:.....;	Ternary operator

Common header files	
#include<stdio.h>	
#include<conio.h>	
#include<stdlib.h>	
#include<string.>	
#include<dos.h>	
#include<math.h>	
#include<time.h>	
Tip : We can deifne our own header file by saving file with <.h> extention	