Programing principal and C·language.

Unite (I)

Father of C-language Dennis Ritchie.
The first Name of C-language B, BPCL
The Created Time of C-language 1961-1972

int=1,2,3,4,5,6,7,8,9,0 %d
float= 2·5, 1·1, etc %F

Full Form of:-

Stdio.h:- Standerd + Input + Output + headerfile
Conio.h:- Console + Input + Output + header file

Program (I)

Q' # include <stdio.h> → Processor.
# include <conio.h> → Directive.
main() → main functions
main() It is well defined library function.
Printf(" "); → Function, It is outpu
library function defined in <stdio.h>
by wich we find output in our progra

Syntax

```
#  include<stdio.h> } Processor
#include<conio.h>   } Directive.

main()

{
    int a,b,Sum,mult, div;
 Printf("enter the two number");
Scanf~~printf~~ ("%d %d,"&a,&b);
 Sum = a+b;
 Printf ("Sum = %d," Sum);
 mult = a*b;
 Printf (" mult = %d,"mult);
 Div = a/b;
 printf ("div= %d," div);

      getch();
}.
```

{ left parenthesis.
} Right parenthesis.

How To work on C Program (Turbo C++)

New file Alt F
Open file Alt O

File Compile Run

Program to Print

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

main( )
{
Printf ("I am programer, I am working
                on c");

Getch ( ):
}
```

Topic-

Steps to execute C program

followings are the Steps to execute C program.

1. Step Double click on turbo C++'
2. Step Click on file.
3. Step Click on new file.
4. Step Type your program
5. Step Save your program - (Save as)
6. Step Compile your program.
7. Step No error.
8. Step Run your program.
9. Step Output.

Operators and expressions.

+ Operators

An operator perform an operation by wich we can find a value in the C-program.

- Minus Operator
% Moduls Operator
\ Slash Operator
+ Sum.

Types of Operators.

1. Arithmatic Operator
2. Logical operator
3. Assignment operator
4. ternary operator
5. Bitwise operator
6. Encrement operator or Decriment operator.
7. Relational operator.

1. Arithmatic operator.

An arithmatic operator is mathmatical operation wich gives us a result. There are two types of Arithmatic operators.

(i) Unary Operator.
(ii) Binary Operator.

(i) Unary Operator.

When an arithmatic operator work on only one or operated then operator is known as unary operator.

Exp. '-', '+'

Syntax (-a).

(ii) Binary Operator

When an operator is operated on two operators then operator is known as Binary operator.

Exp: % Modules, Persant
/ Slash
* Asterisk.

Que- Find value of Following.

```
main()
{
int a=15, b=5;
```

1. a%b=
2. a/b=

Ans. a%b= 0
a/b= 3

Relational operator

When an operator operate on two operator and provide relation ship between two operator that operator is known as relational operators.

Exp. a>b, a<b, a= =b, a!=b, a>=b

Assignment Operators-

A assignment operators provide and assignment which gives its a value.

"=" is an assignment operators.

Esep= $a=10$.

Increment or decrement Operators

An increment operator increases value of any variable one unit.

Esep= $a=5$

a++;

$a=a+1$

$a=5+1$

$a=6$

Decrement Operators.

A decrement Operator decreases value of any variable one unit

esep-

a= 10

a- -;

a= a-1

a= 10-1

a= 9.

11* Program to understand increment operators.

```
#include<stdio.h>
#include<conio.h>
      main()
{
  int a=10;
       a++;

Printf ("a=%d ");

     Getch();
 }
```

Output= 11

Logical Operator.

A logical Operat combines two or more than two expressions and gives result true or faulse.

Their are three types of Logical operators.

1. And Operator (&&)
2. Or Operator (||)
3. Not Operator (!)

And Operator (&&)

And Operator combines two Conditions and gives result True or false.

Boolean Table for and (&&) Operator

Condition	Condition	Result.
True 1	True 1	True 1
True 1	False 0	False 0
False 0	true 1	False 0
False 0	False 0	False 0

OR Operator (||)

OR operators combines Two Condition and gives result True or false.

Boolean Table for Or (||) operator.

Condition	Condition	Result.
True	True	True
True	False	True
False	True	True
False	False	False.

Not Operator (!)
Not operator provide negiation for any Condition

Boolean table for Not (!) Operator.

Condition	Result
True	False
False	True.

Ternary Operator :-
It requires three operands this can be written as- opperands ?

Syntax b>b? a:b

max- Net a=5 , b=10
Solve- false.

Bitwise Operator :-	Meaning.
&	Bitwise and
|	Bitwise or
<<	Left swift
>>	Right Swift.

History of C. Language.

The C. language was developed in 1970 by denish ritchie in USA for the operating System. called unix.

It is derived from besic language ken Thompson in AT &T Ball Laboratory.

This is also adopted from ~~A~~ ~~BD~~ BCPL

BCPL - Besic Combind programming language.

It is middle level language.

```
 # include<stdio.h>
# include<conio.h>
 void main ()
{
    int a;
    int b;
    int c;

printf ("inter the two number");
 Scanf ("%d%d", &a,&b);
      C= a+b;
  printf ("C = %d",c);
             getch();
   }.
```

Tokens:-

The tokens are besic elements recognised by compiler a token is source program text that compiler can not be break in too small elements.

for example -

Keyward. Identifiers, Operators.

Data Type.

Constant :-

Constant is a value thaut can not be change the execution of C. Program.

Syntase.

Const = C = 5;.

Types of Constant:-

Their are three type of Constant.

1. Numeric Constant.
2. Charector Constant.
3. String Constant.

1. Numeric Constant:-

Numeric constant is numeric value wich may or may not be decimal point-

Syntase.

Const C= 2.95;

2. Charecter Constant:-

A charecter constant is an alphabet that can be write in single quart

Exp. 1 'A'
2 'B'
3 'C'
4 'D'

Type of Numeric COnstant.

Their are two Types of Numeric Constant.

1. Integer Constant
2. Real or floating point Constant.

1. Integer Constant:-

Integer Constant are whole number which have no decimal point.

Exp. 0,1,2,3,4 ----- 9

Types of integer Constant.

1. Decimal constant:-
0,1,2,3,4 ----- 9 (Base 10)
2. Octal Constant
0,1,2,3,4 ----- 7 (Base 8)
3. Hexadecimal Constant
0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F (Base 16)

Branching and looping

Types of loop.
following are the Type of loop.

1. while loop
2. do while loop
3. for loop

Loop -
It is a technique for Solving problems or exicuting our program. Many times It is Keywords.

7 Sintax

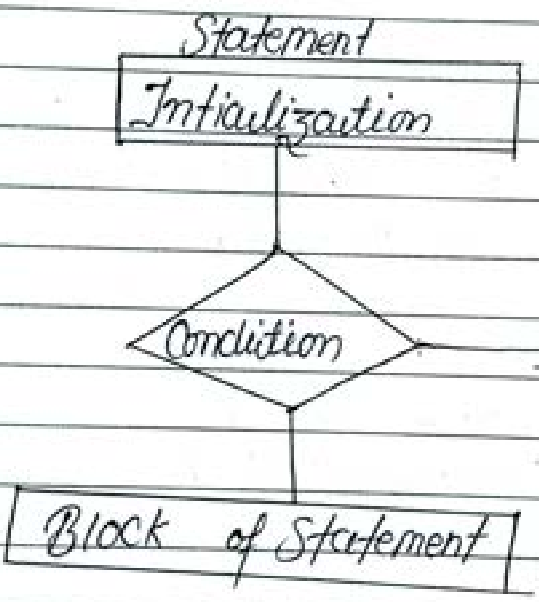
```
        #include<Stdio.h>
        # include<conio.h>
      main()
  {
       int n= 1;
      while (n< 10)
  {
    printf ("n= %d",n);
         n = n+1;
    }.
    Getch();
    }
```

While loop

While loop is used to execute our program or part of program many times.

while loop is use to

While condition



First is evaluate condition if the condition is true the program or part of program is executed if condition is false then it goes of ~~a~~ loop

Chanchal

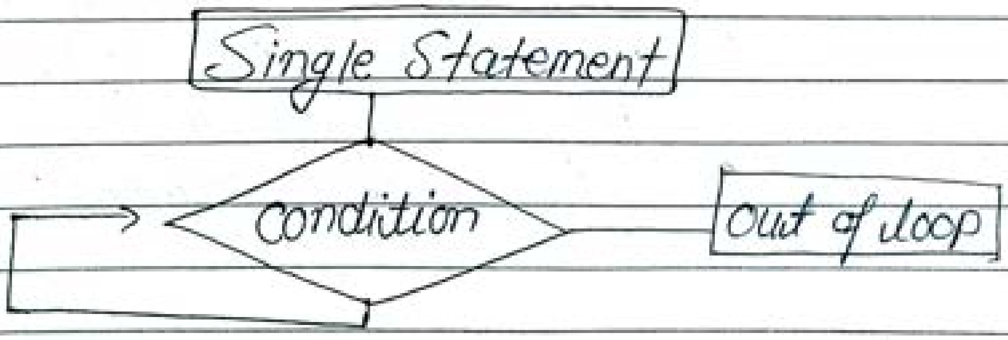
```
 # include<stdio.h>
# include<conio.h>
   main( )
{
  int n;
  n= 1;
 while  (n<10)
{
 printf ("n=%d,"n);
   n= n+1;
}
  getch( );
}
```

Do while loop-

In do while loop is Single Statement will be executed without testing condition and then condition will be avaleble.

Syntax.

Block Digram of do while loop.



Program to understand Do while loop.

```
# include <stdio.h>
# include {conio.h>
     main( )
{
   int n;
  n=1
do
  {
   Printf ("n=%d,"n);
  }
  while (n<11)
 {
   n = n+2;
 }
  getch( );
 }
```

For loop-

For loop contains three sections first section is initialization second is condition and third is increament or Decreament.

Program to understand for loop

```
# include <stdio.h>
# include <Conio.h>
   main()
{
   int i;
for (i=1; i<11; i++)
{
 printf ("i=%d," i);
}
   getch();
}
```

```
Output
   1
   2
   3
   4
   5
   6
   7
   8
   9
   10
```

Program to print (10 to 1) from for loop.

```
#include<stdio.h>
#include<Conio.h>
  main()
{
  int n;
  for (n=10; n<10; n--)
{
  printf("n=%d,"n);
}
   getch();
}
```

```
Output
 10
 9
 8
 7
 6
 5
 4
 3
 2
 1
```

Break and continue statement.

Break statement is used to change loop in the program.

Syntax.

```
        Break;
```

Program to understand Break statement.

```
#include<Stdio.h>
#include<Conio.h>
  main()
{
  int n=1;
 while (n<11)
{
 printf("I am understand use of Break")
   break;
}
  getch();
}
```

Output.

```
        I am understand use of Break"
```

Continue Statement :-

This is well defind statement by which we Continue a loop for execution.

Syntax:-

```
        Continue;
```

Program to understand Continue Statemen

```
# include<stdio.h>
# include<conio.h>
 main()
{
   int n=1;
 While (n<11)
{
 printf("n=%d",n);
      n=n+1;

   Continue;
}
   getch();
}
```

output

```
1
2
3
4
5
6
7
8
9
10
```

If else statement :-

If else statement is one conditional control statement if condition is true then itblock is executed otherwise it goes into else block.

Syntase :-

```
        if (condition);
     {
        Statement;
     else (Condution);
   {
        Statement;
```

Program to understand if else statement

```
      #include<stdio.h>
     #include<conio.h>
   void main()
  {
    int a=5;
    if (a<10)
{ printf("I am doing C Program");
}
 else
{ printf("I am doing C++ Program");
}
    getch();
}
```

Output :- I am doing C Program

Swich Sitatement :-

Swich statement is ~~muld~~ multidimentional conditional statement which gives us multi chalce syntase

```
      Swich (expression)
   {
      Case 1 :
      Statement ;
      Case 2 :
      Statement ;
```

Program to understand Swich Statement.

```
  #include<Stdio.h>
  # include<Conio.h>
  main()
{ printf ("enter your chaice");
  Scanf ("%d" &ch);   // (ch meins chaice)
  Swich(ch)
{   case1:
 printf("Honours");
    case2:
 printf(" First devision");
    Case3:
printf(" Second devision");
    Case9:
 printf("wrong choice");
}
   getch();
}
```

If else Statement

Program to understand for even or odd number.

```
#include<stdio.h>
#include<conio.h>
 main()
{ int n=10;
 if (n%2==0)
{
printf("Number is even");
}
else
{
printf("Number is odd");
 }
   getch();
 }
```

output:- Number is even

Program to understand else if Statement.

```
#include <Stdio.h>
#include< Conio. h>
main()
{
   int a=10, b=5, c=9;
    if (a>b)
  {
  printf ("a is biggest Number");
  }
   else if (a>c)
  {
   printf ("a is bigger than b");
  }
   getch();
  }
```

Output:

A is biggest number.

Program of else if statement :-

```
#include <stdio.h>
# include <conio. h>
 void main( )
{
 int  banana=5, Mango=10 ;
 int  a=5 ; b=10
 if ( a <b)
{
 printf ("banana is sweet than mango");
}
 else if ( b<a)
{
printf (" Mango is sweet than banana");
}
getch( );
}
```

Output

```
Banana is sweet than Mango.
```

go to statement:-

This is unconditional control statement when we use. statement flow of control goes into the another part of program without testing any condition.

Syntax:- goto label Name;

- - - - - - - - -

label Name.

Program to understand go to statement.

```
#include<stdio.h>
#include<conio.h>
 Void main()
{
 int n;
printf ("Enter the Number");
     scanf ("%d", ~~Sum~~ &n);

   if (n%2==0)
 {
   go to even;

 }
 ~~en~~ even:
printf ("Number is even");
     go to end;
   odd:
  printf (" Number is odd");
        go to end;
end
       ~~printf ( "%d /n~~
       printf ("\n");
     getch();
    }.
```

Que→

Write a program to detect given is even or odd.

1 - 79
2 - 81
3 - 65

Solve:-

```
    #include<stdio.h>
    #include<conio.h>
  void main( )
{
 int n1= 79, n2=81, n3=65 ;
 if (n1%2 = = 0)
{
  printf ("n1=79 is even number");
}
```

```
else
 {
  printf ("n1 = 79 is odd number");
 }
elseif (n2%2==0)
{
 printf(" n2 = 81 is even number");
}
else if ("n3%2 = = 0")
{
printf ("n3=65 is even number");

else
 {
 printf ("n3=65 is even number");
 }
 getch();
 }
```

Output

```
n1 = 79 is odd number

n2 = 81 is odd number

n3= 65 is odd number
```

What is array ?

An array is collection of similer type of data item
exp->

```
    int arr [10];
    Char arr [10];
```

Dealaration of array.

Array can be dealared as data type, array name [Size];

Exp- int even number[10];

Program to understand array.

```
#inalude<stdio.h>
#inalude<conio.h>
   void main()
{
 int arr [i], i;
 printf ("Inter the number");
  Scanf ("%d", &i);
  for (i=0; i<10; i++)
{
printf (inter the array = %d", arr[i]);
 Scanf ("%d", & arr[i]);
}
for (i=1; i<11 ; i++)
{
 printf ("enter the number=%d",arr[i]);
}
 getch();
}
```

Scanf

```
#include<stdio.h>
  int main()
{
  int values[5]
printf ("Enter 5 integer");
 for(int i=0; i<5; ++i)
{
 scanf("%d", &values[i]);
}
 printf ("Displaying integers:");
 for (int i=0; i<5; ++i)
  {
   printf ("%d\n", Values[i]);
  }
   return 0;
  }
```

String library function :-

Followings are the string library function.

1. Strlen ( ) function
2. Strcmp ( ) function

Strlen ( ) function :-

This function return the length string that is number of charector in String.

Strcmp ( ) function :-

This function is use to compare of two String

Syntax :-

```
Strcmp (S1, S2)
```

Program to understand Strlen function :-

```
 #include <Stdio.h>
 #include <String.h>
    main()
{
  Char Str [20];
  int lengh;
printf("enter the String");
    Scanf ("%S", & Str);
        length = Strlen (Str);

  printf ("Length of String = %d", length);
      return=0;
  }
```

Topic -
Strcmp() function.

Program to understand Strcmp() function

```
#include<stdio.h>
#include <String.h>
   main()
{
  Char Str1 [10], Str2 [10];
 printf ("Inter the first String");
   Scanf ("%S", & Str1);
 printf ("Inter the second String");
  Scanf ("%S" & Str2);
   if (Str cmp (Str1, Str2) = = 0)
 {
 printf ("Strings are same");
 }
 else
 {
 printf (" Strings are not Same");
 }
    getch();
}
```

Types of Array.

Followings are the type of array

1. One-dimensional. Array
2. Two-dimensional. Array
3. Mult-dimensional. Array

1. One dimensional Array:-

If any array contains only one bracket then that array is known as one dimensional array

Syntax:-

```
int arr[10];
```

2. Two-dimensional Array:-

If array contains two bracket then array is known as Two dimensional array.

Syntax:- `int arr[10][10];`

Multidimensional array:-

If any array contain two or more than two array then that array is known as multidimensional array

```
int = arr [10] [10] [10];
```

Program to understand arry.

```
 #include<Stdio.h>
 #include< conio.h>
    main()
{
   int mat[i] [j], raw, col;
printf ("Inter the raw matrix");
scanf ("%d", & raw);
  printf ("Enter the column of matrix");
  scanf("%d", & col);
 for (i =1; i < row; i++)
  for (j = i; j <col; j++)
   printf ("Enter the raw = %d, Col= %d," i,j)
scanf ("%d", &mat[i] [j]);
 printf (" Matrix: /n");
 for (i=1; i<row; i++)
 for ( j=1; j < col ; j++)
```

Function:-

Funcation is a collection of object which gives us a value in program it solves our problem when we use it in our program.

What is Garbage funcation:-

GCD Garbage Common diviser.
enum, enumerated data Type.

Garbage value is a part of automatic memory management. it is value when we creat objects at run time.

What is dynamic memory allocation

When we allocate memory during execution of C program is Known as dynamic memory allocation

Followings are the function of Dynamic memory allocation.

1 Size of () function
2 Malloc () function
3 Calloc () function
4 free ()
5 Realloc () function.

1. Size of () function:-

Sizeof function is an unary function. This function provide us unary operation it increases size of argument.

//* Program to understand Size of function.

```
# include<stdio.h>
# include <conio.h>
  void main()
{
 Struct
{ Char name [10];
 int age;
 float marks;
}
 rec;
int arr [10];
printf ("Size of Structure=%d", Size of(rec));
printf ("Size of int=%d", Size of (rec));
printf("Size of array=%d", Size of(rec));
     getch();
}.
```

Malloc Function:-

The mallloc function used to alllocate memory space.
The mayllloc function used to reserved memory space and gives starting address of the pointer.

Syntax:-

```
Ptr = (int*) malloc(10);
```

The above malloc memory Size 10 It holds datatype integer and Ptr Store address of the variable.

```
//* Program to understand Malloc functine.
#include <stdio.h>
# include <conio.h>
void main()
{
 int i;
int *p;
        p=(int*)malloc(Size of(int);
}
```

Here p is pointer.

Calloc function :-

Calloc function is used to allocate memory. It allocate multiple block of memory space.

Syntax:
```
ptr = (int*) calloc (5,2);
```

Free Function :-

Free function release memory space when it is used in the program.

Syntax:
```
Free(ptr);
```

inc.

Realloc Function:-

The realloc function change the size of ~~function~~ block of the C-program
It re-allocate the memory space.

Syntax:
```
ptr = R- allocate (Specified Size);
```
पुनः प्रदर्शिकरणा

```
//* Program for Quotient of any Numbe.
#include<stdio.h>
# include < conio.h>
void main()
{
 int a,b,quotient;
 printf("Inter The Two Number");
  Scanf(" %d%d," &a, &b);
 quoatient= a/b;
 printf ("Quotient= %d," quo);
      getch();
}
```