UNIT-1:-

Introduction to C Language

C Language Overview:-

The C-programming language is general purpose and high-level language that was originally developed by Dennis M. Ritche's to develop the UNIX OS at Bell Labs.

C was originally first implemented on the DEC PDP-11 computer in 1972.

In 1978, Brian Kernighan and Dennis Ritche produced the first publicly available description of C, now known as K&R standard.

C has now belome a widely used proffesional language for various reasons.

(i) ~~Easty~~ to learn

(ii) Structured language

(iii) It can handle low-level activities.

(iv) It can be compiled on a variety of computer platforms.

Facts about C:-

(i) C was invented to write an OS called UNIX.

(ii) The 'c' language was formalized in 1988 by ANSI (American National Standard Institute).

(iii) The UNIX OS was totally written in C in 1973.

(iv) Now a days, C is widely used and popular system programming language.

(v) Today's most popular linux OS have been written in C.

Uses of C Language:-

(i) OS , (ii) Language Compilers (iii) Language Interpreters (iv) Assemblers ,(v) Text Editors , (vi) Print Spoolers, (vii) Network Drivers , (viii) Modern Programs , (ix) Databases, (x) Utilities.

Basic Structure of C Language:-

```
Documentation Section
Linking Section
Definition Section
Global declaration Section
Main function Section
     {
          Declaration Section
          Executable Section
     }
Sub program or function Section.
```

Algorithm:-

An algorithm is a problem solving technique. It can be defined as a setep by step procedure to solve a particular problem or task or an activity each sted is called instruction.

Characteristics of an algorithm:-

(i) Input , (ii) Output , (iii) Definateness , (iv) Finiteness, (v) Effectiveness.

Ex-1:-

Algorithm to find sum of two numbers.

Step 1 : Start
Step 2 : Accept / Read 2 numbers from the used a,b.
Step 3 : Calculate the sum , c = a+b
Step 4 : Display the value of c.
Step 5 : Stop.

Flow Chart:-

A flow chart is a graphical representation or pictorial representation of an algorithm.

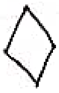
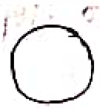
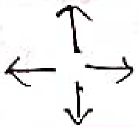
(or)

The diagrametical representation of way to solve the given problem is called flow chart.

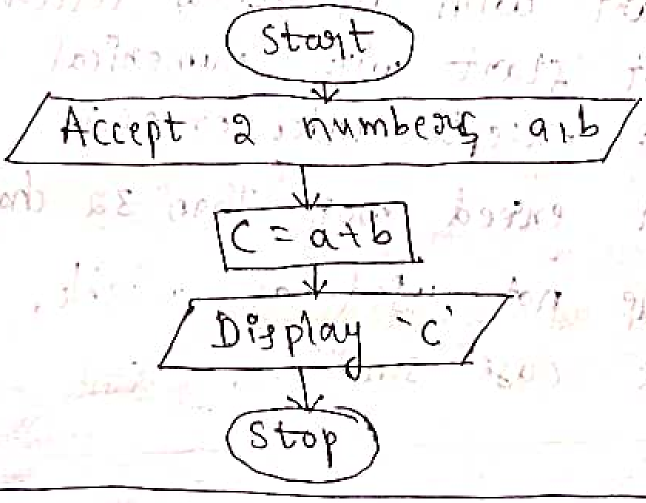
Advantages of Using flow chart:-

(i) Communication, (ii) Effective analysis,
(iii) Proper documentation, (iv) Efficient Coding,
(v) Proper debugging, (vi) Efficient program Maintenance.

Given symbols are used to draw a flowchart:-

	oval	start / stop
	parallelogram	Input / Output.
	Rectangle	Process
	Diamond	Decision
	Circle	Connector
	Arrows	Flow

Ex-1:-
Draw a flowchart for sum of 2 numbers.



C language elements:-

Tokens, Comments, Keywords, Identifiers, Constants, String literals, Punctuation and special characters.

Programming in C-Elements:-

Keywords:-

Keywords are the words whose meaning has alread been explained to the C compiler. The keywords cannot be used as variable names.

There are only 32 keywords available in C.

Key Words

- auto
- break
- case
- char
- const
- continue
- default
- do
- double
- else
- enum
- extern
- float
- for
- goto
- if
- int
- long
- register
- return
- short
- signed
- sizeof
- static
- struct
- switch
- typedef
- union
- unsigned
- void
- volatile
- while.

Variables:-

Variable is an identifier that is used to represent a single data item.

```
ex:  int a,b,c;        // integer variables
     char d;           // character variable
```

Rules for declaring a variable:-

(i) It should start with alphabetical letters only.

(ii) It should not start with numerical values, special symbols except underscore (_).

(iii) It should not exceed more than 32 characters.

(iv) Keywords are not used as variable.

(v) Variables are case sensitive.

Datatypes and Sizes:-

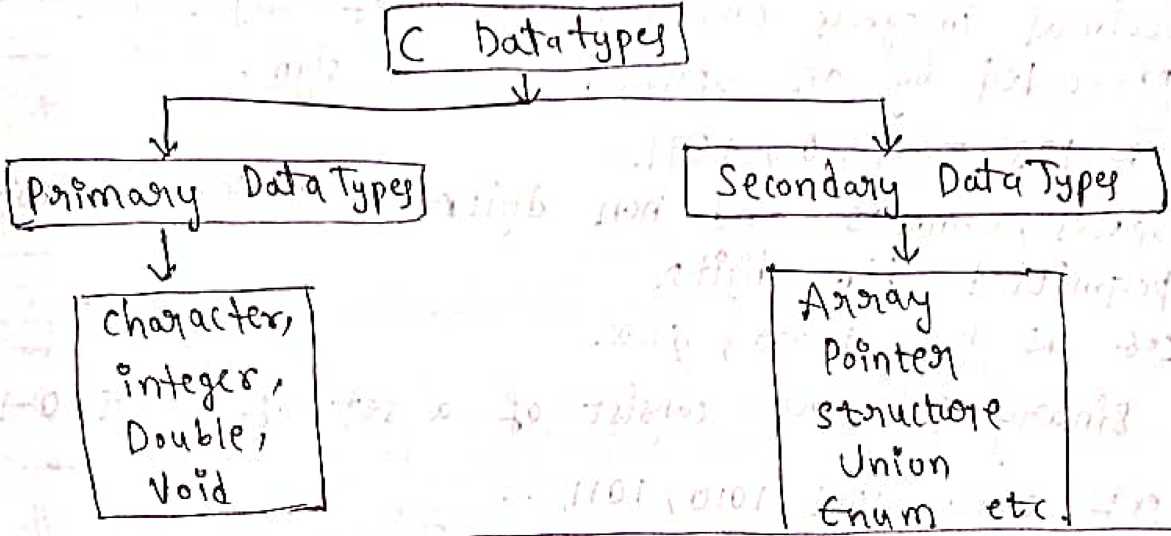
There are 4 data types in C language. They are

(i) Basic datatype (or primary (or primitive datatype.
⇒ int, char, float, double

(ii) Enumeration datatype.
⇒ enum

(iii) Derived datatype (or Secondary datatype
⇒ pointer, array, structure, union

(iv) Void data type
⇒ void.



Storage size of data types:-

Sl. No.	C-datatype	Storage size in (bytes)	Range
①	char	1	-127 to 127
②	int	2 (or 4	-32767 to 32,767.
③	float,	4 (or 8	1E-37 to 1E+37. with six digits of precision.
④	double	8	1E-37 to 1E+37 with ten digits of precision

Escape sequence Codes:-

			\v	⇒	Vertical Tab
\\	⇒	\ character	\?	⇒	? character
\'	⇒	' character	\a	⇒	Alert or bell
\"	⇒	" character.	\b	⇒	Back space

\f → Form feed

\n → New line

\r → Carriage return.

\t → Horizontal tab

\v → Vertical tab

\ooo → Octal number of one to three digits.

\xhh → Hexadecimal number of one or more digits.

Numerical Constants:-

(a) Integer Constants:-

Decimal, Binary, Octal, Hexadecimal.

⇒ Decimal integers consist of a set of digits 0-9 preceeded by an optional - (or) + sign.

ex:- 123, -321, 0, +591...

Spaces, commas and non digits character are not permitted b/w digits.

ex:- 15 150, 20,000, $501...

⇒ Binary integers consist of a set of digits 0-1.

ex:- 0110, 1001, 1010, 1011,...

⇒ Octal integers consist of a set of ~~digits~~ 0-7. with leading '0'.

ex:- 037, 0551, 0435, 0...

⇒ A sequence of digits preceded by 0x is considered as hexadecimal integer. They may also include alphabets a-f (or) A-F. The letters A through F represents the number 10-15.

ex:- 0x2, 0x9f, 0xbcd, 0x...

(b) Real Constants:-

ex:- 0.0083, -0.75, 43.36, +247.0.

ex:- 215., .95, -.74, +.5...

ex:- 215.65 may be written as 2.1565e2 in exponential notation. e2 means multiply by 10^2. // ↓e2 = E2

Character Constants :-

ⓐ Single character constant :- ↓ 'a' = single quote.

ex :- '5', 'P'. ; character constant within a pair of ' '.

Character constant '5' is not same as the number 5.

Character constant have integer values known as ASCII values.

```
printf ("%d",'a');
```

ex:- the number 97 = ASCII value of a.

ⓑ String Constants :-

A string constant is a sequence of characters enclosed in double quotes.

ex:- "hello!", "1987", "?...!", "5+3", ...

Operators in C :-

The symbols which are used to perform logical and mathematical operations in C program are called C operators.

⇒ Operators, functions, constants and variables are combined together to form expressions.

ex :- a+b * 5

Here, a,b are variables, (+,*) are operators. 5 is ~~a~~ constant.

a+b * 5 is an expression.

Types of C operators :-

(i) Arithmetic Operators

(ii) Assignment operators

(iii) Relational operators

(iv) Logical operators

(v) Bit wise operators

(vi) Conditional operators (ternary operators)

(vii) Increment / decrement operators

(viii) Special operators.

(i) Arithmetic Operators:-

Arithmetic operators are used to perform mathematical calculations like addition, substraction, multiplication, division and modulus in C programs.

⇒ +	Addition	a+b
⇒ -	substraction	a-b
⇒ *	multiplication	a*b
⇒ /	Division	a/b
⇒ %	modulus	a%b

(ii) Assignment Operators:- To assign values for variables.

⇒ Simple assignment operators.

= ;	sum=10 ;	10 is assigned to variable sum

⇒ Compound assignment operators.

+= ;	sum += 10 ;	sum = sum+10
-= ;	sum -= 10 ;	sum = sum - 10
*= ;	sum *= 10 ;	sum = sum * 10
/= ;	sum /= 10 ;	sum = sum / 10
%= ;	sum %= 10 ;	sum = sum % 10
&= ;	sum &= 10 ;	sum = sum & 10.
^= ;	sum ^= 10 ;	sum = sum ^ 10 ;

(iii). Relational Operators:-

Relational operators are used to find the relation between two variables.

> ;	x > y ;	x is greater than y
< ;	x < y ;	x is less than y
>= ;	x >=y ;	x is greater than or equal to y
<= ;	x <=y ;	x is less than or equal to y.

== ; x==y ; x is equal to y.

!= ; x!=y ; x is not equal to y.

(iv). Logical operators:-

To perform logical operations.

&& ; logical AND ; (x>5)&&(y<5)

|| ; logical OR : (x>=10)||(y>=10)

! ; logical NOT ; !((x>5)&&(y<5)).

⇒. True = 1, False = 0

a	b	a && b
1	1	1
0	1	0
1	0	0
0	0	0

Truth Table

a	b	a||b
1	1	1
0	1	1
1	0	1
0	0	0

Truth Table

a	!a
1	0
0	1
1	0
0	1

Truth Table

b	!b
0	1
0	1
1	0
1	0

a	b	!(a&&b)
1	1	0
0	1	1
1	0	1
0	0	1

a	b	!(a||b)
1	1	0
0	1	0
1	0	0
0	0	1

(v) Bit wise operators:-

To perform bit operations. Decimal values are converted into binary values

& ; Bitwise AND

| ; Bitwise OR

~ ; Bitwise NOT

^ ; XOR

<< ; Left shift

>> ; Right shift.

⇒ True = 1 , False = 0

a	b	a&b
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table

a	b	a|b
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table

a	b	a^b
0	0	0
0	1	1
1	0	1
1	1	0

Truth Table

(vi). Conditional or ternary operators:-

Syntax:-

(Condition ? true_value : false_value) ;

ex:- (A>100 ? 0:1);

(vii) Increment and Decrement operators:-

Syntax:-

Increment operator : ++ var_name or var_name ++;

Decrement operator : -- var_name or var_name--;

ex:- ++ i ; i++

-- i ; i--

++ i <5 → 1,2,3,4.

i++ <5 → 1,2,3,4,5.

--i <5 → 9,8,7,6.

i-- <5 → 9,8,7,6,5.

(viii) Special Operators:-

& ; &a will gives address of a.

* ; *a is pointer to the variable a

sizeof() ; sizeof (char) will give us 1.

C statements fall into three categories:-

⇒ Selection statements ; if and switch

⇒ Iteration statements ; while, do, and for.

⇒ Jump statements ; break, continue and goto.

Other C statements ⇒ Compound statement, Null statement.

Selection Statements (or Decision Making Branching statements:-

① If ⇒ simple if, if else, Nested if, If else ladder.

② Switch

(1) If :-

if :- Syntax ⇒

```
if (condition)
{
  statements;
}
```

```
if ()
{ =
}
```

if... else :- Syntax ⇒

```
if (condition)
{
  statement1;
}
else
{
  statement 2;
}
```

```
if ()
{ =
}
else
{ =
}
```

Nested if :- Syntax ⇒

```
if (condition1)
{
  ~~statement 1;~~
  ~~else~~
    if (condition 2)
    {
      statement 1;
    }
    else
    {
      statement 2;
    }
  ~~statement 3;~~
}
else
{
  statement 3;
}
```

```
if ()
{
  if ()
  { =
  }
  else
  { =
  }
}
else
{ =
}
```

Ladder if :- Syntax ⇒

```
if (condition 1)
{ statement 1;
}
else if (condition 2)
{ statement2;
}
else
{
 statement 3;
}.
```

```
if ( )
{ =
}
else if ( )
{ =
}
else
{ =
}.
```

(ii) Switch :-

Switch() statement is useful for writing menu driven programs.

Syntax :-

```
switch(expression)
{ case 1 : statements;
           break;
  case 2 : statements;
           break;
  default : statements;
           break;
}
```

Iteration Statements (or) Decision Making Looping Statements :-

There are 3 types of loop control statements in C language. They are,
(i) for , (ii) while , (iii) do-while

for :- Syntax :-

```
for (exp 1 ; exp 2 ; exp3)
{ statements;
}
```

(or)

```
for(i=0; i<10; i++)
{ statement;
}.
```

(or)

```
for(initialization ; condition ; incredment/decrement)
```

⇒ for loop statement comprises three actions. these three actions are (a) Initialize, (b) Counter, (c) Test condition

While:- Syntax:- while (condition)
```
{
    statements ;
}
```

⇒.
```
while ( i<10)
{
    statement ;
    i++;
}
```

Do-while:- Syntax:-
```
do
{
    statements ;
}
while (condition);
```

⇒
```
do
{
    statement ;
    i++ ;
}
while ( i <4);
```

Jump Statements:-

Break:- syntax:- break;

Continue:- syntax:- continue;

goto:- syntax:-
```
{
    .........
    goto case 6; ~~[illegible]~~
    -----
    ......
    case 6 : statement ;
}
```

srikrishna
Indu

G. Srikrishna devarayulu Bhumi [illegible]

శ్రీకృష్ణదేవరాయలు Venu

VKSBI

⇒ 29|12|2021

G. Sri Krishna devarayulu Krishnaveni

⇒ 9:16 PM

Arrays:-

Array is a collection of homogeneous elements which represented by a single variable.

(or)

Array is a group of related data items that share a common name.

The complete set of value is referred to as an array, the individual values are called elements.

One-dimensional Array (or) Single Subscripted variable:-

A list of items can be given one variable index is called "single subscripted variable" (or) "1-d array".

eg:- int a[3];

It contains 3 elemente. The range starting from 0-2 elements. (a[0], a[1], a[2]).

Declaration of 1-d array ⇒

Type variable-name [size];

Initialization of arrays ⇒

Type array name [size] ={List of values};

Two-dimensional array:-

Declaration of 2-d array ⇒

type array name [row-size] [col-size];

eg:-
```
int a[3][3];
  for ( i=0 ; i<3 ; i++)
   for (j=0 ; j<3 ; j++)
    scanf ("%d", &a[i][j]);
```

Multi-Dimensional Arrays:-

Type array-name [s1][s2][s3].... [sm];

Note:- ANSI , C language does not specify any limit for array dimension. However most compilers permit seven to ten dimensions.