→ C TOKENS →

→ C tokens are the Basic Building Blocks in C language which are constructed together to write a C prg.

*). Each & Every smallest individual units in a C prog are known as C tokens.

→ C tokens are of six types they are,

(i) keywords (eg: int, while),
(ii) Identifier. (eg main, total).
(iii) Constants (eg. 10, 20)
(iv) Strings (eg: "total", "hello");
(v) Special symbols (eg: (), {});
(vi) Operators (eg +, /, −, *).

C Token Example prog:-

```
int main().
{
   int x, y, total;
   x = 10, y = 20;
   total = x + y;
   Printf ("Total = %d \n", total);
}
```

→ where, main - Identifier
{, }, (,) = delimiter
int - keyword
x, y, total - identifier
main {, }, (,), int, x, y, total - tokens

→ Do you known how to use C token in real-time applications Programs. where C token is used, you can refer the Below C program,

→ Indentifiers in C Language :-

→ Each program Elements in a C program are given a name called identifiers.

→ Names given to identify Variable, fn and arrays are Example for identifiers, Variables, fn and arrays are Example for identifiers, eg: x is a name given to integer variable is above prog.

→ Rules for constructing identifies Name in C :-

(i) First character should be an alphabet @ underscore
(ii) Succeeding character might be digit @ letter.
(iii) punctuation and Special characters are n't allowed Except underscore
(iv) identifiers should not be keywords

3). Keywords in C language

→ keywords are pre-defined words in a ~~com~~ C Compiler
→ Each keyword is meant to perform a specific fn in a C prg.
→ Since keywords are refered name for compiler, they can't be used as variable name.
→ C language support 32 keywords which are given Below. →

auto, double, int, struct, const, float, short, unsigned
break, else, long, else long, switch continue, for, void (2)
case, enum, register, typedef, default, goto, sizeof, volatile
char, Extern, return, union, do, if, static, while.

Constants:

↳ are also like normal variable, But only difference is, their values can not be modified by the prog once they are defined.

↳ Constants refers to fixed values, they are also called as literals.

↳ Constants may be belonging to any of the data type

Syntax:

```
const data_type variable_name;
(or)
const data_type *variable_name;
```

Types of C constant

Constant type	data type Example
i) Integer constants →	int, (53, 742, -478) unsigned int (5000U, 1000U. etc) long int, long long, int (483, 6477, 147, 483, 680)
Real @ Floating point constants	float (10.456789) double (600.12345678)
Octal Constant	int (Ex: 013 Astarts with 0 */)
Hexadecimal Constant	int (Example: 0X90 A Start with 0X */)

Character constants → char (Example: 'A', 'B', 'C')
String constants → char (Example: "ABCD", "Hai")

Rules for constructing C constant

(i) Integer constants in C

→ An integer constant must have at least one digit
→ It must not have a decimal point
→ It can either be positive @ negative
→ No commas @ Blanks are allowed within an integer constant
→ If no sign precedes an integer constant, it is assumed to be positive.
→ The allowable range for integer constants is -32768 to 32767.

(ii) Real constants in C

→ A real constant must have at least one digit
→ It must have a decimal point
→ It could be either positive @ negative
→ If no sign precedes an integer constant, it is assumed to be positive
→ No commas @ Blanks are allowed within a Real constant

(iii) Character and String constants in C

→ A character constant is a single alphabet, a single digit & special symbol enclosed within single quote
→ The maximum length of a character constant is 1 character
→ String constants are enclosed within double quotes

(iv) Backslash character constant in C

→ There are some characters which have special meaning in C language

→ They should be preceded by Backslash symbol to make use of special fn of them

→ Given Below are the list of special character and their Purpose.

Backslash character	Meaning
\b	Backspace
\f	Form feed
\n	New line
\r	carriage return
\t	Horizontal tab
\"	Double quote
\'	Single quote
\\	Backslash
\v	Vertical tab
\a	Alert @ bell
\?	Question mark
\N	Octal constant (N is an octal constant)
\XN	Hexadecimal constant (N- hexdcml const)

→ The Symbol which are used by
C-operator + Expression

→ The system which are used to perform logical + mathematical operation in a C prog. are called C operators

→ These C operator join individual constants + variables to form Expression

→ Operator, fn, constants and Variable are combined together for form Expression

→ Consider the Expression A+B*5 where +, * are operators A, B are variable, 5, is constant & A+B*5 is an Expr.

Type : Arithmetic operator
→ Assignment operator
→ Relational operator
→ logical operator
→ Bit wise operator
→ Conditional Operator (ternary operator)
→ Increament / decrement operator
→ Special Operator (&, *, sizeof)

Arithmetic operator

+ (Addition) → A+B, A-B, A*B, A/B, A%B

```
#include<stdio.h>
int main()
{
  int a=40, b=20, add, sub, mul, div;
    add = a+b;
    sub = a-b;
    mul = a*b;
    div = a/b;
Printf("Addition of a, b is: %d\n", add);
Printf("Subtraction of a, b ---", sub)
Printf("
}
```

Assignment operator

operator	Example / Description
=	sum = 10; 10 is assigned to variable sum
+=	sum += 10; This is same as sum = sum + 10
-=	
/=	
%= , &= , ^=	

Relational operator :- are used to find the relation B/w two variable, to compare the value of two variable in a C prg

operator >, <, >=, <=, ==, !=

```
#include <stdio.h>
int main()
{
  int m=40, n=20;
  if (m==n)
  {
    printf("m and n are equal");
  }
  else {
    printf("m and n are not equal");
  }
}
```

Logical operator are

&& (logical AND)	(X>5)&&(y<5)	(both condition true)
|| (logical OR)	(X>=10)||(Y>=10)	(true, at least one of the condition is true)
! (logical NOT)	!((X>5)&&(y<5))	

Bitwise operators

& - Bitwise AND
| - -"- OR
~ - NOT
^ - -"- EXOR
<< - left shift
>> - Right shift

Conditional operator ✓

→ Conditional operator: Return one value if condition is true and returns another value if condition is false

→ This operator is also called as ternary operator

Syntax : (condition ? true_value : false_value);

Example (A>100 ? 0 : 1);

Increment / decrement operator

→ Increment operator are used to increase the value of the variable by one & decrement operator are used to decrease, the value of the variable by one in C

prog

Syntax Increment operator: ++var_name; (or) var_name++

Decrement operator: --var-name ; (or) var_name--;

Example :

++i , i++;
--i , i--;

Operators	Description
&	This is used to get the address of the variable. Ex:- &a will give address of a
*	This is used as pointer to a variable Ex:- *a where, * is pointer to the variable
Sizeof	This gives the size of the variable. Ex:- size of (char) will give as 1

Ex:- ~~int *ptr~~

```
#include <stdio.h>
int main()
{
int *ptr, q;
q = 50;
/* address of q is assigned to ptr */
ptr = &q;
/* display q's value using ptr variable */
Printf ("%d", *ptr);
return 0;
}
```

Special Symbol in C

→ they have a special meaning which cannot be used for another purpose

(i) Square Brackets [];→ The opening & closing Brackets represent the Single & Multidimensional subscript

→ Simple brackets (); it is used in f^n declaration & f^n calling, for Ex:- printf() is a pre-defined f^n

→ Curly braces {}→ used in the opening & closing of the code, it used in the opening & closing of loops

→ Comma (,) → it is used for separating for more than one statement and for Example, Separating f^n Parameters in a f^n call, separating the variable when Printing the values of more than one variable using Single printf statement

→ #, (Hash) /pre-processor → is used for pre-processor directive, It Basically denotes that we are using the header file

→ Asterisk (*) → this symbol is used to represent Pointers and also used as an operator for multiplication

→ Tilde (~) → It is used as a destructor to free memory

→ period (.) → It is used to access a member of a Structure or a union

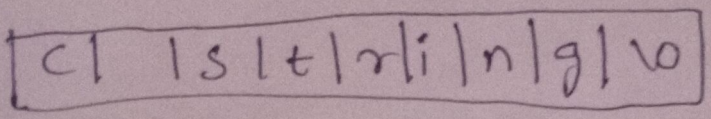
Strings in C

Sequence of character terminated with a null character '\0',
→ stored as an array of character

→ strings are used for storing text/ character
Ex:- "Hello world", is a string of characters

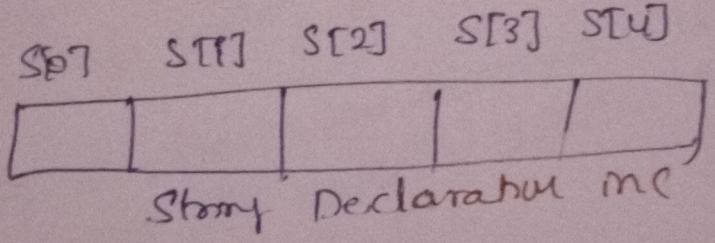
Char c[] = "c string";

m/m diagram



How to declare a string?

Char S[5];



Initialize strings

1) Char c[] = "abcd";
2) Char c[5] = "abcde";

Assigning value to string:

Char c[100]
c = "C programming"; // Error ! array type not assignable

Note: use the strcpy() fn to copy the string instead

→ you can use the Scanf() fn to read a string