**DAY 06**

* **Tools in linux :**

gcc ,make ,ctags ,cscope ,splint ,gdb, valgrind , gproof, cunit, git, gcov(coverage tools)

* **c tags:**

It is just like a book marking

here we can find which file function is declared

* installing ctags

sudo apt install universal -ctags

ctags -R.

vi file1.c

set tags = <path of tag file>

ex: project directory

if I am in source file

set tags = ../../tags

ctrl[] ---- >navigates to mod1 function

return backward ctrl -t

* **cscope:**

c scope is similar to ctag

sudo apt install cscope is used for installation

it stores data in file called cscope.out

cscope ‘find. -name “.\*[ch]” . means current working directory

(or)

Find .-name “.\*[ch]”>cscope.files

Ctrl d is used to come out of cscope

* **gprof**

this gprof allows to measure percentage of time is spent on different functions

compile

gcc -o application -pg main.c

-pg means we are profiling in the main .c

* **Valgrind tools:**

>> valgrind -v – tool =memcheck --leak-check =full –show -reachable =yes

--log-file =valclient2 ./a.out

(a.out is the application name)

This command will produce the valgrind report

>>vi valclient1

>>vi main.c

>>gcc -0 app main.c

>>

Valgrind -v –tool= memcheck—leak-check=full –show -reachable =yes –log file =valclient2 ./app

>> vi valclient2

>>valgrind -v –tool=memcheck—leak -check=full –show -reachable =yes –log file =valclient2 ./app

* **gdb:**

It is an gnu debugger

To go for next (next(or)N) is used for debugging

To go step (or)s

* **C Programming :**

**Basic structure of c program**

Documentation section: (details like when and why it is created)

Description :

[example :

Modules add,sub

Int add (int val1,val2): add is doing addition of val1 with val2 and return the result to the called (main)}

Author:[ initials of the author]

DOC/DOM:[if it was created by us then that date but if it is a modified then it will be authors date]

Version:[0.1v]

[1.0 stable version🡺 baing of the project\*/

Link section (linking header files includes macros)

Definition section(prototyping of function defining modules)

Global declaration section

Function section

Main()

{

Declaration part--- declaration with initialization

Executable part--- which performs an action

}

* **Constants in c:**

The constants in C are the read-only variables whose values cannot be modified once they are declared in the C program. The type of constant can be an integer constant, a floating pointer constant, a string constant, or a character constant. In C language, the **const**keyword is used to define the constants.

const *data\_type var\_name* = *value*;

* **variables in c:**

A variable in C is a memory location with some name that helps store some form of data and retrieves it when required. We can store different types of data in the variable and reuse the same variable for storing some other data any number of times.

data\_type variable\_name = value; // defining single variable  
 or  
data\_type variable\_name1, variable\_name2; // defining multiple variable

* **Datatypes in c**

Primitive Data Types 🡪 int, char, float, double, void

[Derived Types](https://www.geeksforgeeks.org/c-derived-data-types/) 🡪 array, pointers, function

[User Defined Data Types](https://www.geeksforgeeks.org/c-user-defined-data-types/) 🡪 structure, union, enum

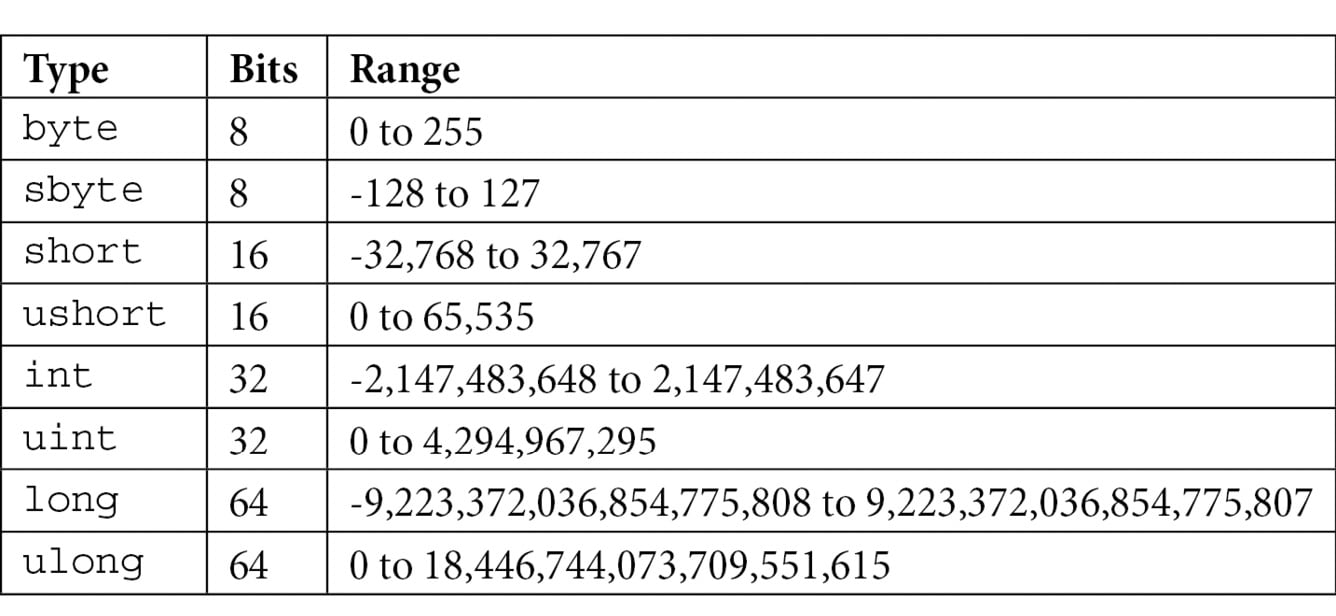
* **C tokens:**

1. Keywords
2. Identifiers
3. Constants
4. Strings
5. Special Symbols
6. Operators

* **Types of Operators in c :**

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operators
6. Other Operators

* **Type modifiers in c :**
* [short Modifier](https://www.geeksforgeeks.org/data-type-modifiers-in-c/#short-type-modifier-in-c)
* [long Modifier](https://www.geeksforgeeks.org/data-type-modifiers-in-c/#long-type-modifier-in-c)
* [unsigned Modifier](https://www.geeksforgeeks.org/data-type-modifiers-in-c/#unsigned-type-modifier-in-c)
* [signed Modifier](https://www.geeksforgeeks.org/data-type-modifiers-in-c/#signed-type-modifier-in-c)
* C supports eight integer types that represent various ranges of integral numbers. The bits and range of each of them are shown in the following table:

****

* **Types of Increment Operator**

There are two types of increment operators – pre increment and post increment.

**Pre (Prefix) Increment Operator**

In an expression, the pre-increment operator increases the value of a variable by 1 before the use of the value of the variable.

**Post (Postfix) Increment Operator**

In an expression, the post-increment operator increases the value of a variable by 1 after the use of the value of the variable.

* **Types of Decrement Operator in C**

**1. Pre-Decrement Operator**

The pre-decrement operator decreases the value of the variable immediately when encountered. It is also known as prefix decrement as the decrement operator is used as the prefix of the operand.

**2. Post-Decrement Operator**

The post-decrement happens when the decrement operator is used as the suffix of the variable. In this case, the decrement operation is performed after all the other operators are evaluated.