**Structure of c language**

Documentation Section

Link Section

Definition Section

Global Declaration section

Main() Funciton Section {

Declaration Part;

Executable Part;

}

Sub program Section

{

Function 1;

Function 2;

Function 3;

.

.

Function N;

}

**C HAS 32 KeyWords**

**Auto, double, int, struct brae else long switch case enum register typedef char extern return union continue for signed void do if static while default goto sizeof volatile const float short unsigned**

**Identifiers**

Identifiers are names of variable, functions and arrays

Rules:

It must be a sequence of letters,digits and underscore

It must begin with a letter

Uppercase and lowercase are different in cc

No special Chars or punctions are allowed  
No Successive underscores are allowed

Keywords cant be used as identifiers  
  
**Variables**The Quantity that changes during the execution of the program is calle a variable

Th variables represent a particular memory location where data can be stored

**Primary Dataypes**

**Integral Data type**

**Integer types**

**Char Types**

**Floating point datatype**

**Int – integer - %d**

**float – float - %f**

**double – long float - %lf**

**char – character - %c**

**String – array of chars - %s**

**Operators**

Unary Operators: - ,++,--

Binary Operators : +,-,\*,/,% arithmetic

<,<=,>,>=,~,^ relational operator

&&,||,! Logical

&, | , <<,>>,`,^ bitwise

=,+=,-=,\*=,/+,% assignment

? :ternary operator

**Type Conversion**

It is the method to convert one data type into another data type

**2 types**

Implicit / Explicit

**Implicit**

Char, short int -> int -> unsigned int > long int> unsigned long int > float > double > long double

**ASCII**

**A 🡪 65**

**Z 🡪 90**

**.a 🡪 97**

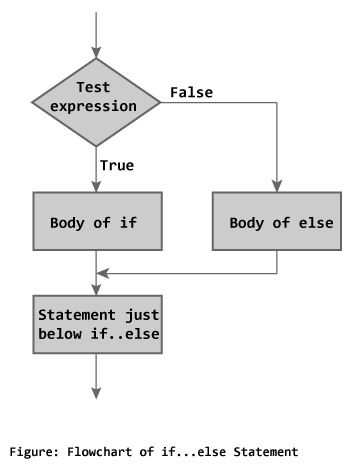
**.z 🡪 122**

**0 🡪 48**

**9 🡪 57  
  
  
  
Control Statements**

**If , Else , If else, If else if ladder, nested if, switch statements**

**They help you to make decisions based on certain conditions.**



IF STATEMENT

**If(Condition)**

**{**

**//true statements**

**Statement 1;**

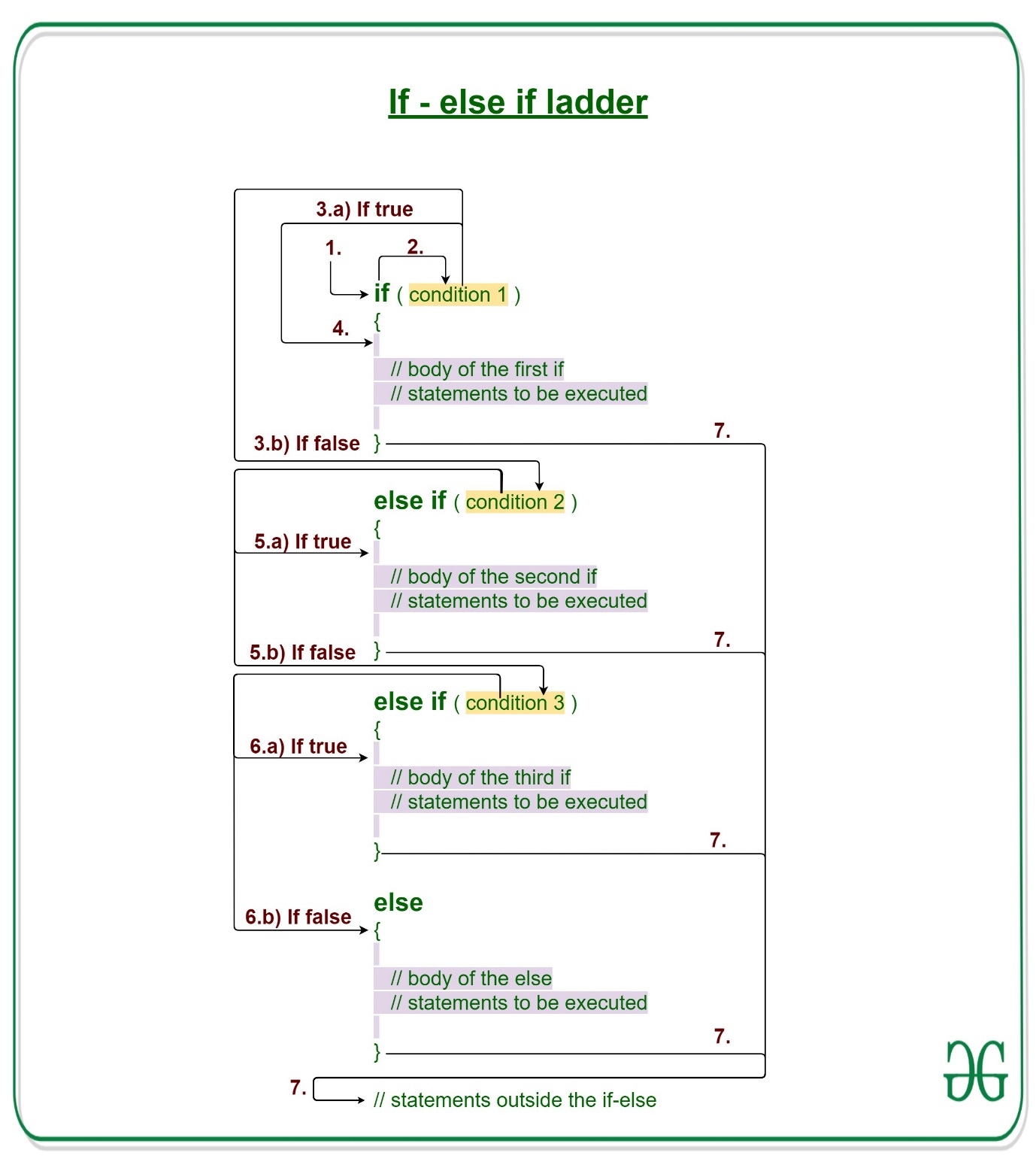
**Statement 2;**

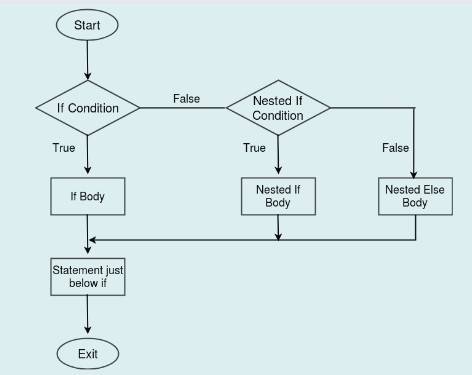
**.**

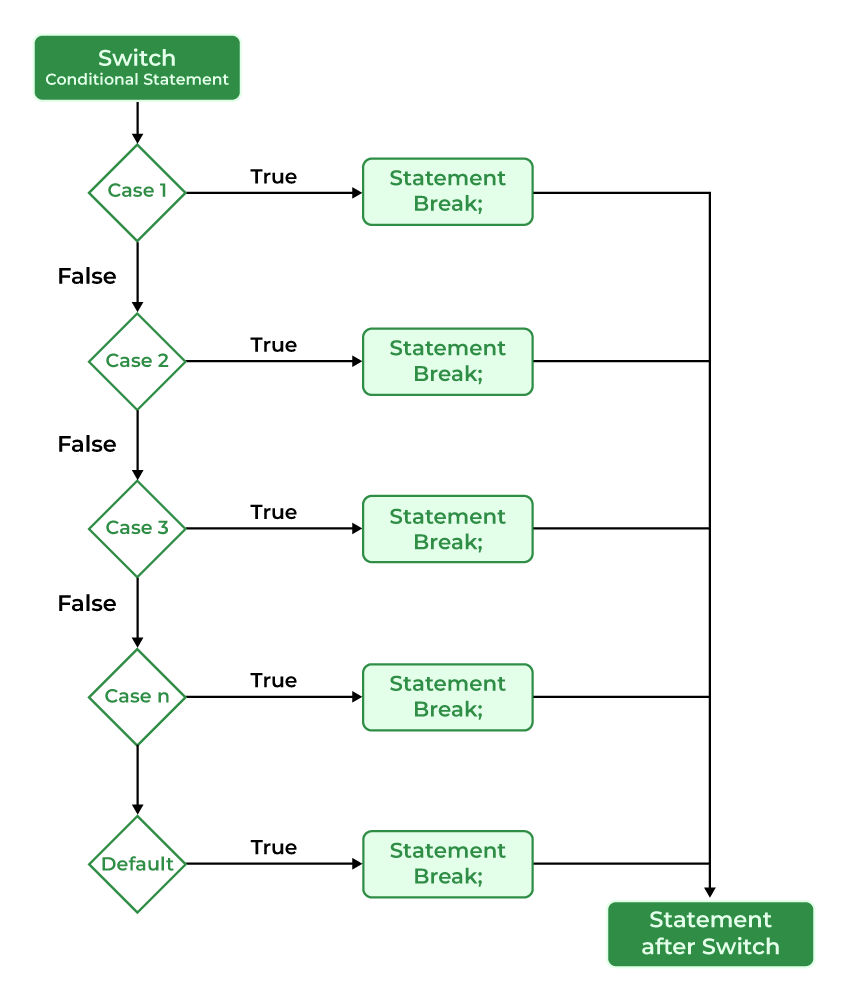
**.**

**.**

**}**



**NESTED IF**

**Switch**

**Loop**

**A statement is repeatedly executed until the condition is satisfied.**

**DO while**

**Syntax:**

**Do {**

**//Code**

**//updation**

**}**

**While(condition);**

**Arrays**

**An arrays is a collection of similar data types in which each element is unique one and located into separate memory locations;**

**They are stored in consecutive memory locations,**

**Each element in an array is referenced by a subscript enclosed in a pair of square brackets.**

**Types of Arrays:**

**1.Single dimensional Array**

**2.MultiDimensional Array**

**The dimension is determined by the number of subscripts present in the given array**

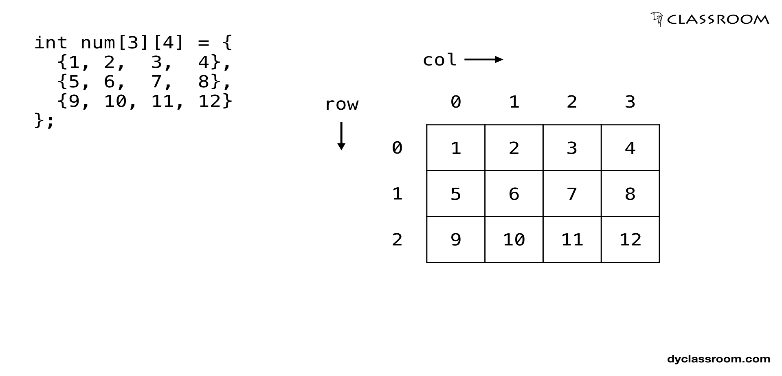
**For example,**

**2 Dimensional Array:**

**Data items arranged in the horizontal direction called rows and that in vertical direction are referred as columns**

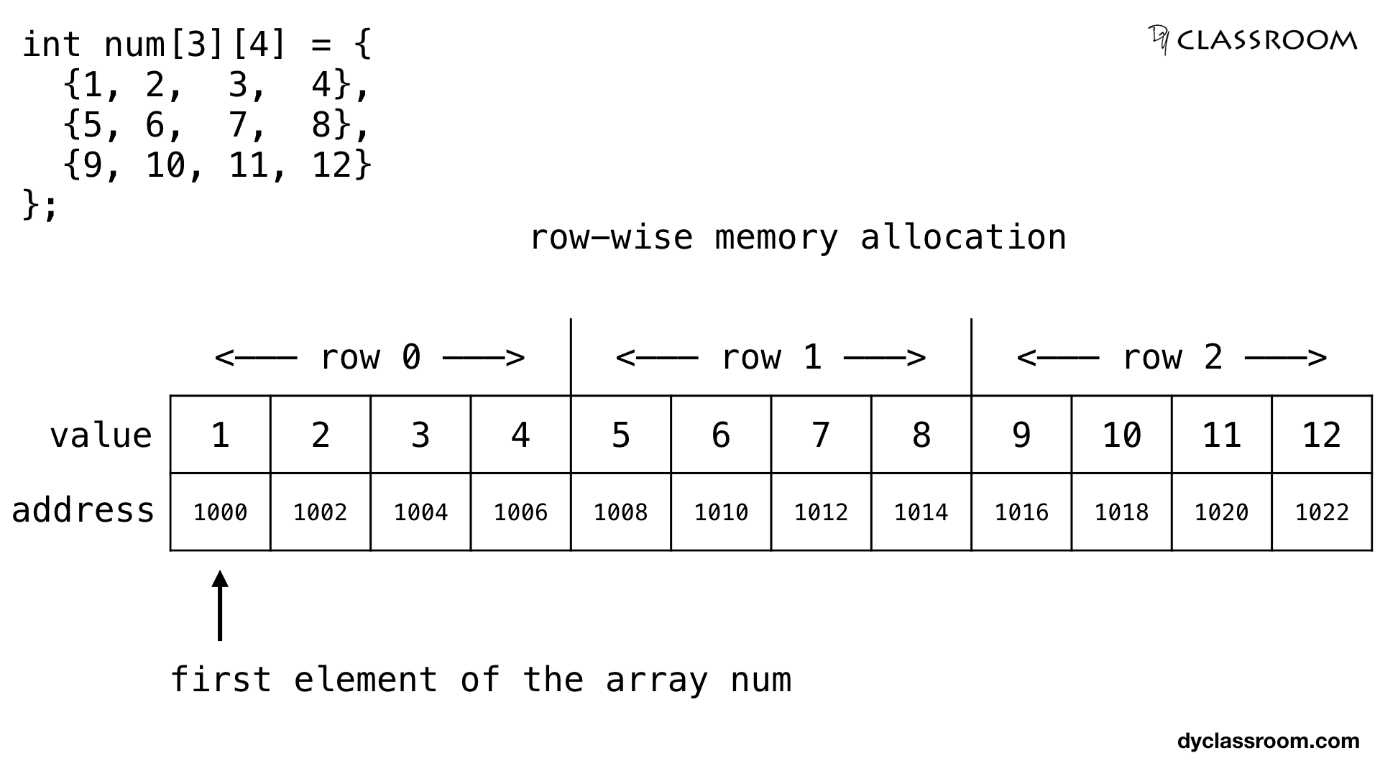
**Declaaration of 2D array**

**Syntax**

**Datatypes array\_name[row][columns];**

**Example:**

**Int arr[3][3];**



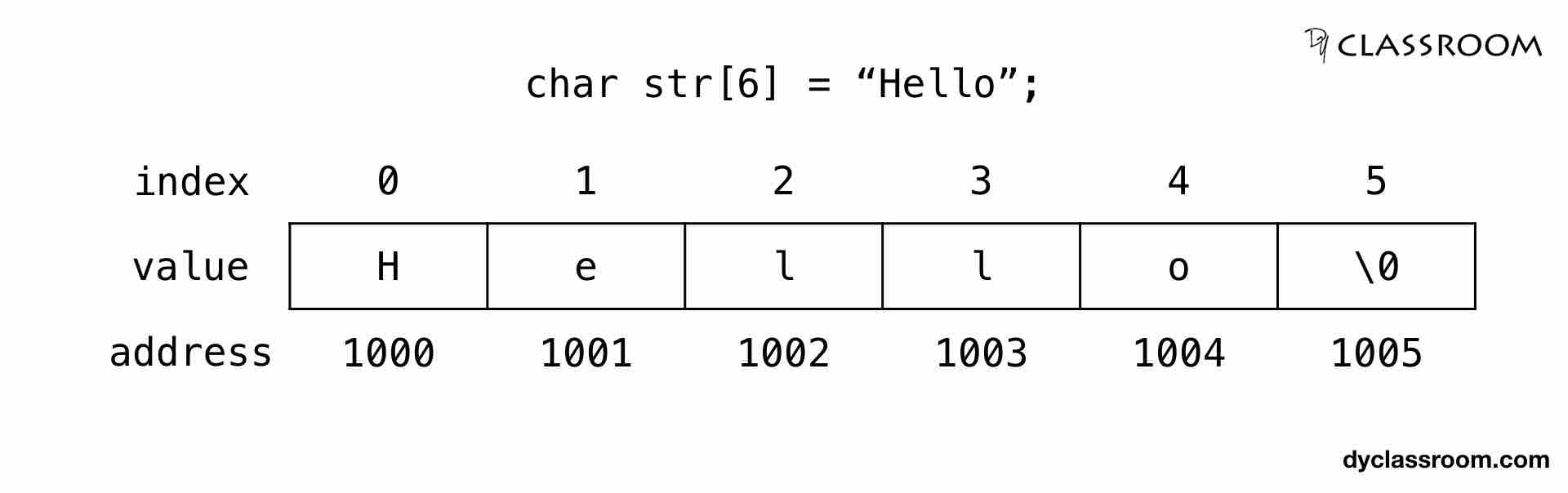
**Strings:**

In c language the group of characters digits and symbols enclosed within quotation marks called as string

The string is always declared arrays are called string

Every string terminated with \0 NULL character

The NULL character is a byte with all bits at logic zero. Hence its decimal value is zero.

**MEMORY REPRESENTATION OF A STRING.**  
  


**FUNCTIONS:**

**A functions is a self contained block one or more statement that perform a special task when called .**

**The c language support two types of functions:**

**1.Library functions**

**2.user defined functions**

**Library Functions:**

**The library functions are pre defined functions.**

**A user cannot understand the internal working of the standard functions and cannot be modified but can only use all standard functions**

**The user must include the prototype declarations to use the standard library functions.**

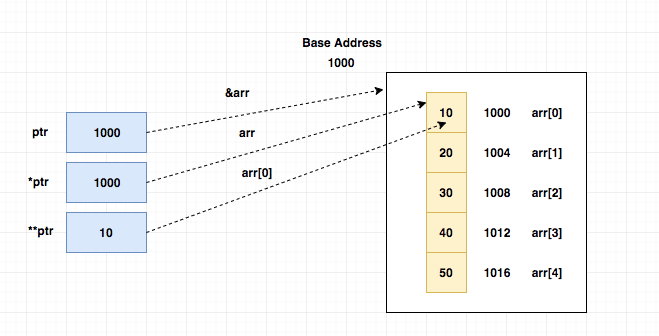
**The example of standard functions are scanf(), printf(),sqrt(),abs etc….**

**User defined functions:**

**User defined functions are the functions defined by the user according to the requirements.**

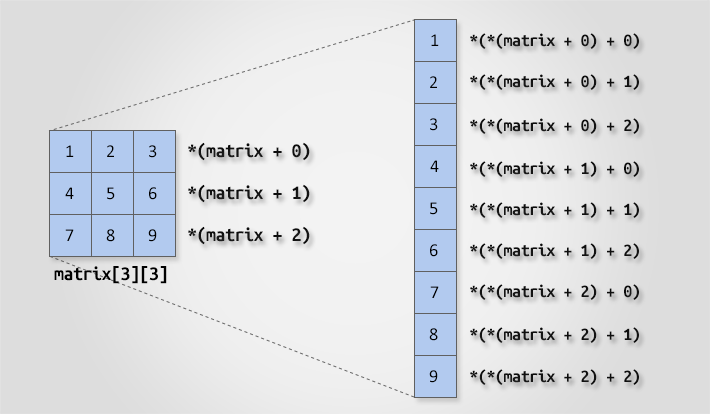
**The user understands the internal working of the function.**

**The main() functions is also a user defined functions except that the name of the function, the number of arguments , and the arguments**

**POINTERS AND ARRAY**

**ARRAY OF POINTER**

**Like an array of variables we can also use array of pointers in c**



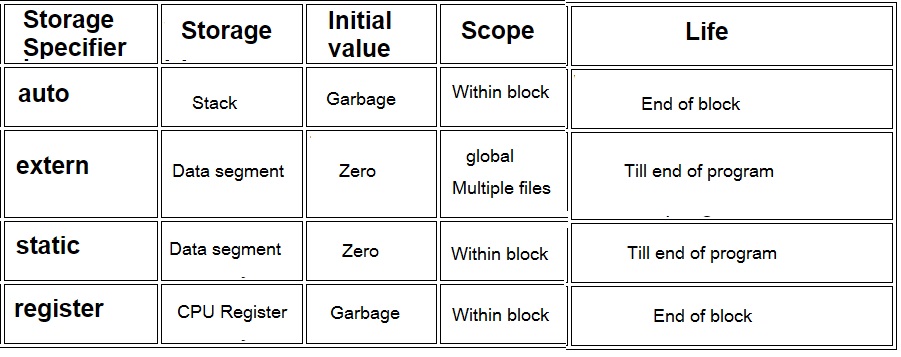
**POINTER TO POINTER**

**PASSING POINTER TO FUNCTION  
we can also pass the pointer variable to function**

**In other words we can pass the addresso of a variable to the function instead of varibale value**

**So if we change the argument value in the function it will modify the actual value of a variable**

**STORAGE CLASS IN C**



**D – M – A**

**Dynamic memory allocation :**

**in c memory cllocation can statically (during compile-time) automatically or dynamically**

**the memory allocation for global and static )variables is allocated on the stack and is done automatically during function calls**

**1.malloc() used to allocate a specified number of bytes as a block of memory from the free store.**

**2.calloc() used to allocate a specific no of bytes from the free store and also initialize them to zero**

**3. realloc used to increases or decreases the size of the specified block of memory**

**4. free() used to release the specified block of memory to the system**

**The MALLOC()**

**The malloc function is used to allocate memory during execution. It takes only one integer argument which represent the number of bytes to be allocated.**

**The malloc function returns the base address to the block of memory allocated on the heap**

**The return type of that address is void\*, hence in the code we need to typecast**

**That void\* type to the primitive data type of the pointer variable.**

**(void \*) malloc(number of bytes);**