Arrays :→ Pointers :→ The variable which stores the address of another variable is a pointer.

Var : A container / memory location where we store some data.

address operator &a

int a = 10; (1000)

memory block 1000 x [(&a)]

Address :→ Hexa Addr → %p
Base Addr → %x

int * ptr = &a; (1000)

x = 10 (1000)

int * ptr = &a; (ptr = 2000, &a = 1000)

*ptr

a | 10 (1000)

*ptr | 1000 (2000)

**dptr | 2000 (3000)

Pointer to another pointer (Double Pointer)

int ** dPtr = & ptr; (2000)

function {
x = 20 (2000)
}
main {
ptr = 2000 *ptr = 20
}

Storage Classes :

(Dangling Pointer)

* Static → global — file
* extern → global — folder
auto → local
register → RAM memory

no pointer

Reva - DS

static x
extern y
f1.c
f2.c
x
y

* When we try to access data from different scopes, it is called a dangling situation.

ADA
DAA

* Arrays :→ It is a homogeneous collection of data.

int arr[] = { 1, 9, 8, 6, 5, 4 }; (values)
1 2 3 4 5 6 → position
0 1 2 3 4 5 → index

data = [1.2, 1, True, "Hi"]; Python (Heterogeneous)

1D, 2D
↓
Matrix

(index = pos - 1) (pos = index + 1)

int arr[10] = {1, 2}; (8 zeroes) ** 8 x 4 = 32 bytes

(Runtime / Execution) (memory = money) (DMA) → Runtime

* (malloc) (size) 32 bytes garbage pointer
* (calloc) (n, size) 0 0 0 0 0 0 0 0 int 4 default
* realloc (ptr, new size)
* free (ptr)

<Stdlib.h>
void pointer
(int *) (char*)

C++ (new delete) (DMA)

C (cstdlib)

* malloc calloc realloc free
* Struct & union
* strings

* Recursion - Interview Questions