(Dynamic Memory Allocation) (Pointers)

(%d) Pointer: → It is a variable which stores the address or reference of another variable. (0x1000) int *ptr = &a;

%p %x

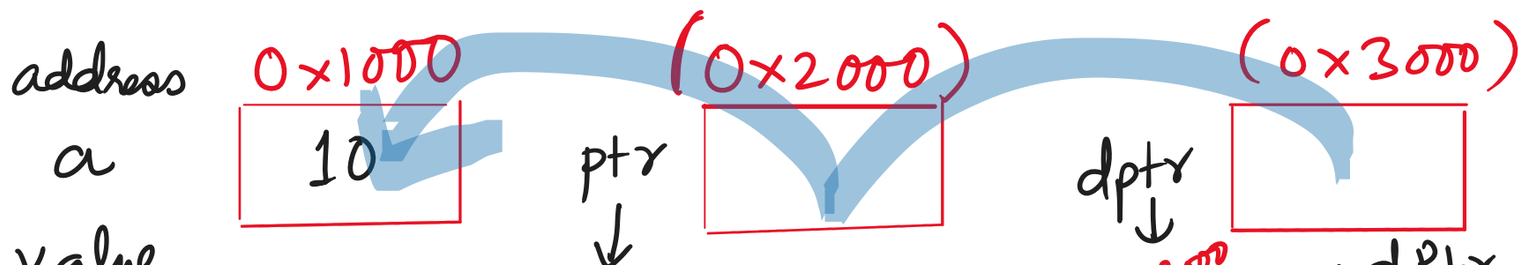
Address operator (&a)

int a = 10;

variable or identifier → value

(0x1000) address of a

(dynamic) (runtime) (compiler allocated)



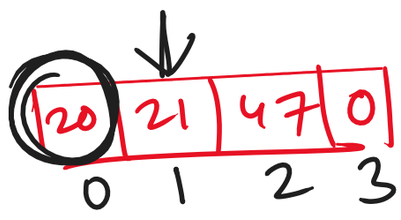
*ptr = 10
ptr = 0x1000

*dptr = Junk
dptr = 0x2000

int a = 10;
int *ptr = &a;
int **dPtr = &ptr;

**dptr = 10

(pointer to a pointer) (double pointer)



arr[i]
i=1

&ptr = 2000
dptr = 2000

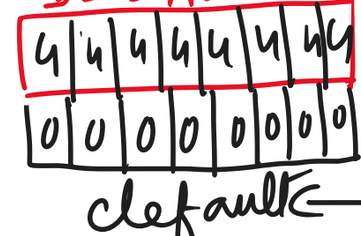
Dynamic Memory allocation: → (Asking the user how much (one single) (Block) memory he/she needs and then allocating it, rather the wasting.

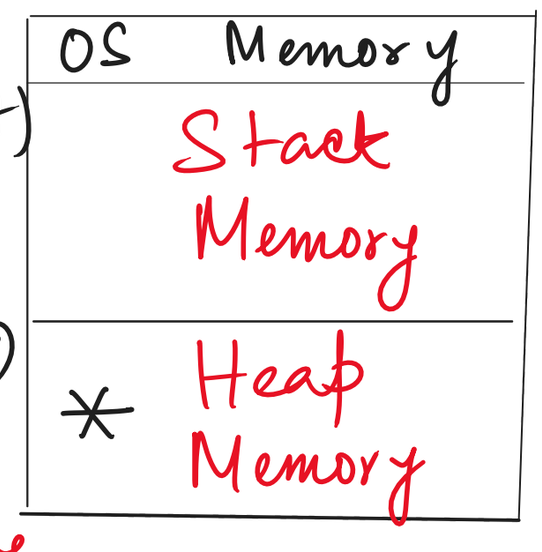
#include <stdlib.h>

(I) *malloc(size);
(II) calloc(n , size);
(III) realloc(ptr, new size);
(IV) free(ptr);

32 bytes

garbage





These two functions return "void pointer". We need to perform type casting.

* In case of C++ (malloc, calloc, realloc & free are optional & not mandatory)
* If we want to use them: We use #include <cstdlib>
* But, to make the life of developers simple, C++ uses:

(new → object) Java ***

(I) new → for memory allocation
(II) delete → for memory de-allocation