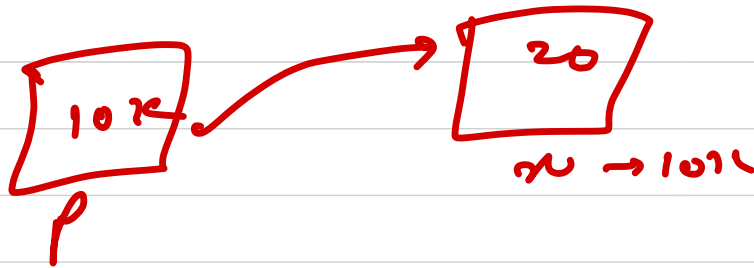


Types Of Pointers :

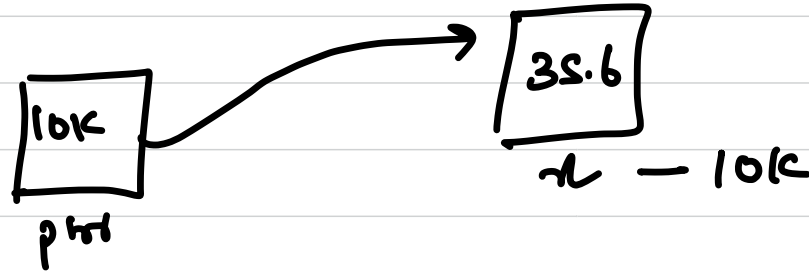
1) Wild Pointer → Uninitialized Pointers . Because of that they can have any random value, inside them that can later cause problems

int *p; ← wild pointer

2) Dangling pointer \rightarrow A pointer pointing to a memory location that was previously available but now has been deallocated.



3) Void Pointer →



float* ptr = &x;

→ what-type of bucket it is pointing

to.

void * ptr



It doesn't have a specific type associated.
The type of data can be anything.

↳ They cannot be dereferenced

Deferring

if $pr = 8x4$

how to
init
pointer



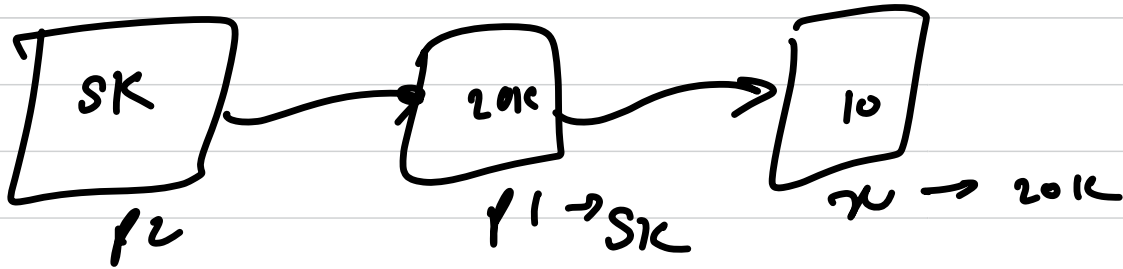
int y = *pr
└─┘
└─┘ without union
↓
deferring operators

cout << *pr << "\n";

int x = 10;

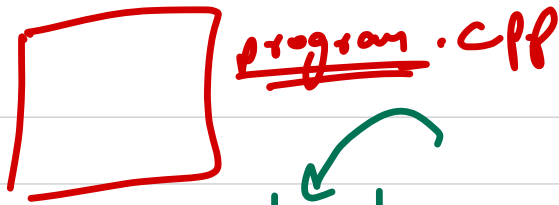
int * p1 = &x;

int ** p2 = &p1;



**p2

→ 10



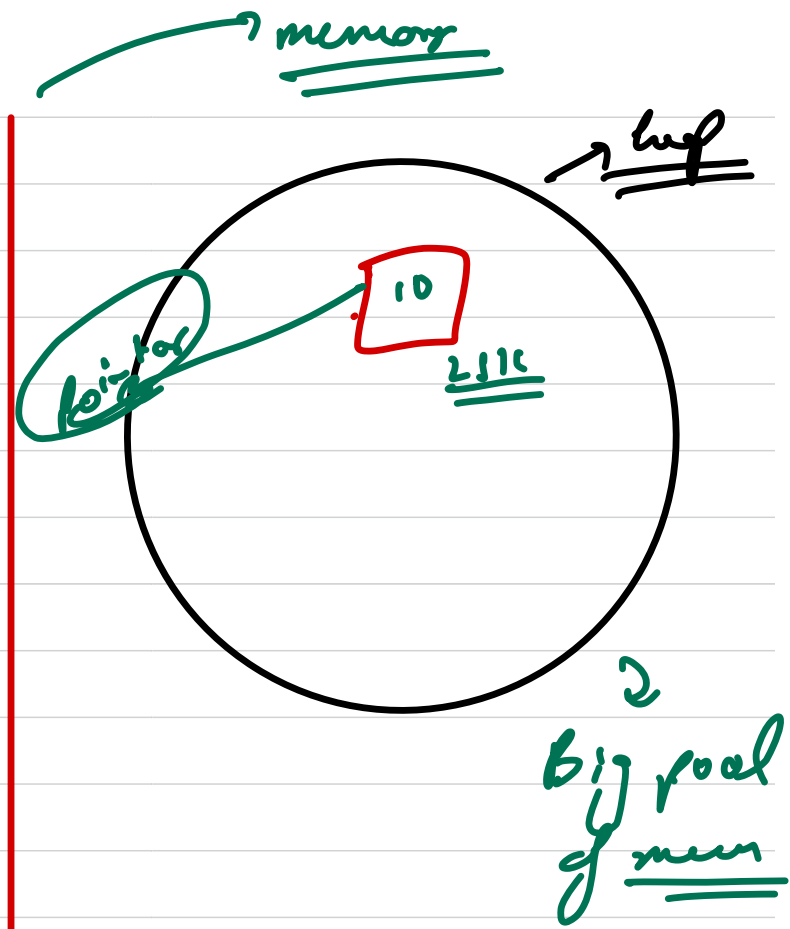
LIFO

Stack

linear
memory
space

content
line 13

name
func



whenever we call a funcⁿ

main() \rightarrow func \rightarrow gen C) ...


```
#include <iostream>
```

```
void fun() {
```

```
std::cout << "fun" << "\n";  
}
```

```
int main() {
```

```
→ int x = 10;
```

```
→ fun(); ←  
return 0;
```

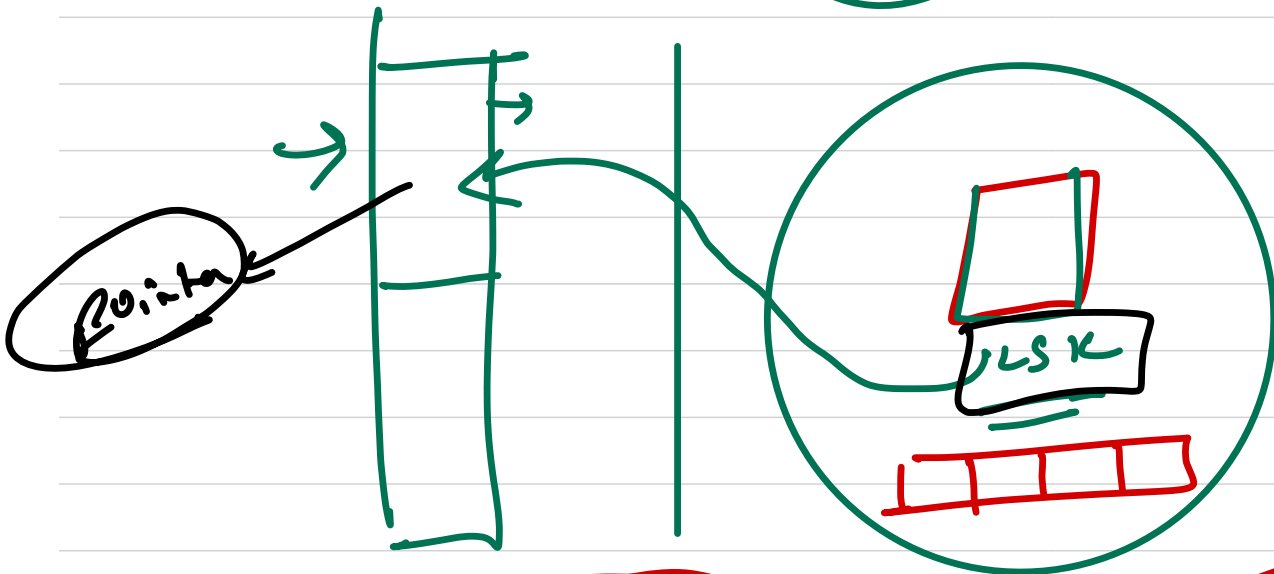
//6

```
}
```

↳ function →

malloc

(size) → 4
4 byte

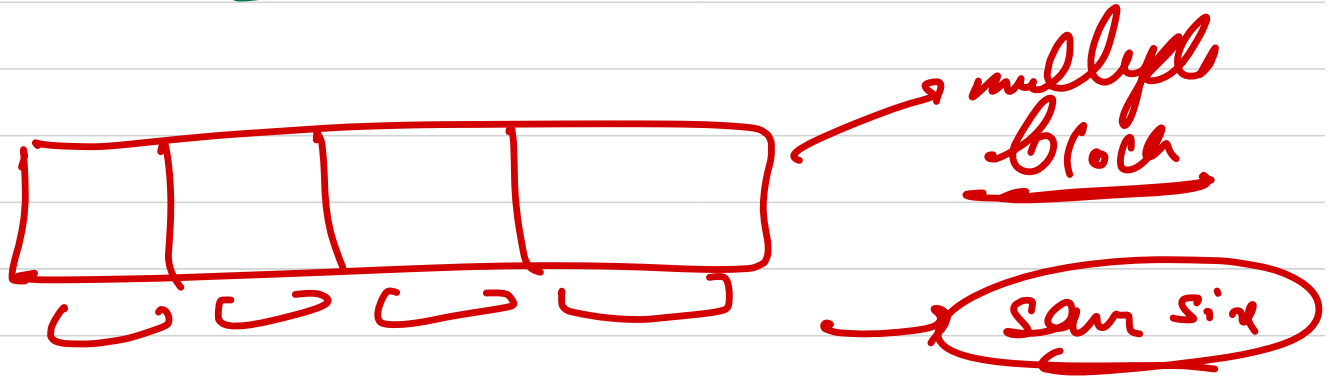
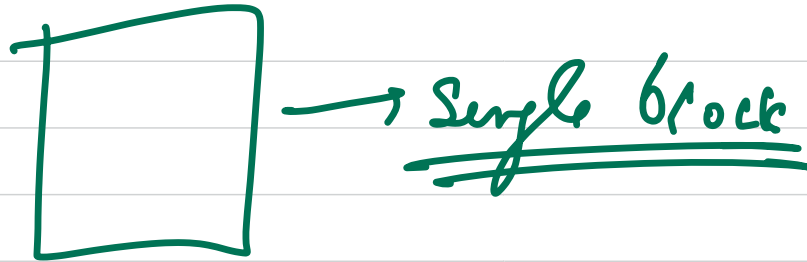


void *

typerant

(int *)

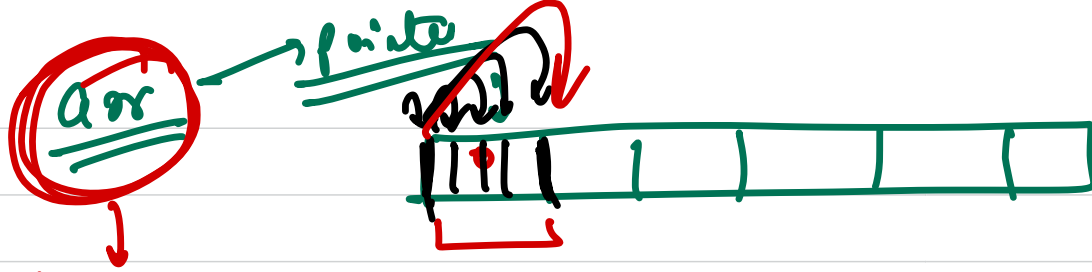
A red-outlined rectangle representing a memory block, with two red arrows pointing towards it from the right.



`calloc(4, 5)`

no of bytes → size of block

The number 4 is labeled "no of bytes" and the number 5 is labeled "size of block", which is underlined.



it stores
address of
base / 0th index

Computer ptr

int arr[10]
label

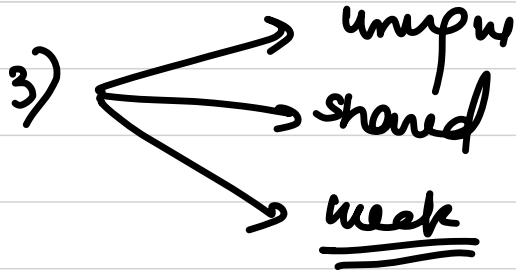
pointer address

int 1 byte

ptr

Smart Pointer

Smart Pointers are an abstract interface to actual raw pointers but with additional benefit of auto memory management.

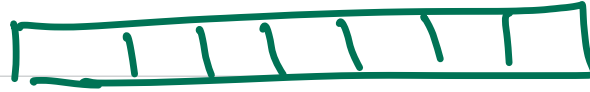


~~Ref count~~

1) Unique Pointer This is a smart pointer that will only permit one owner of the pointer.
i.e. unique_ptr can contain at max only one single raw pointer that points to single memory location

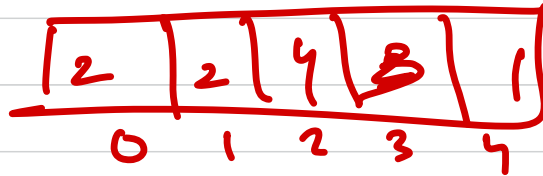
Shared_ptr → The shared_ptr is another choice for a smart pointer in C++. if we want to have multiple owners. This maintains reference count of all pointers which point to the object.

Q



Scanned which are affected by
neighbors

Prefix
sum



differs
value

$a[i] \neq 1$
 $a[i+1] \neq 1$

max sum

0 2
1 0 4 ✓
0 0 3 ✓
2 3 ✓