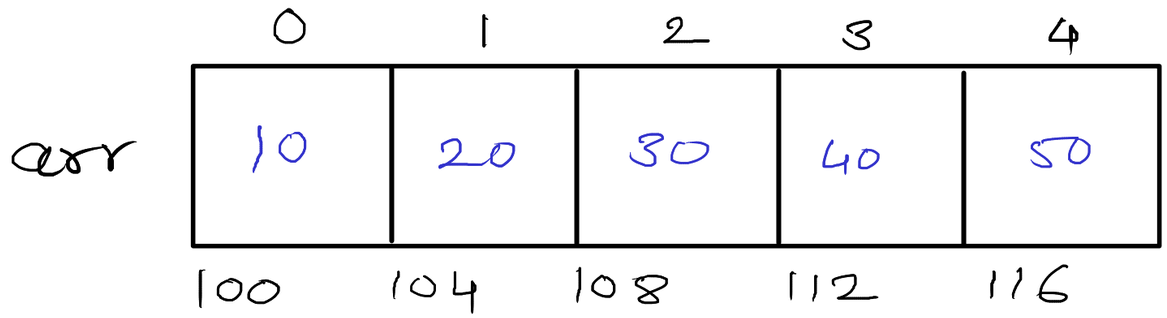


arr[0] = *(arr+0)
arr[1] = *(arr+1)
arr[2] = *(arr+2)
⋮
arr[4] = *(arr+4)

arr[i] = *(arr+i)

Array Notation — Pointer Notation

arr - base address (100) → address of first address

&arr - address of (100) whole array

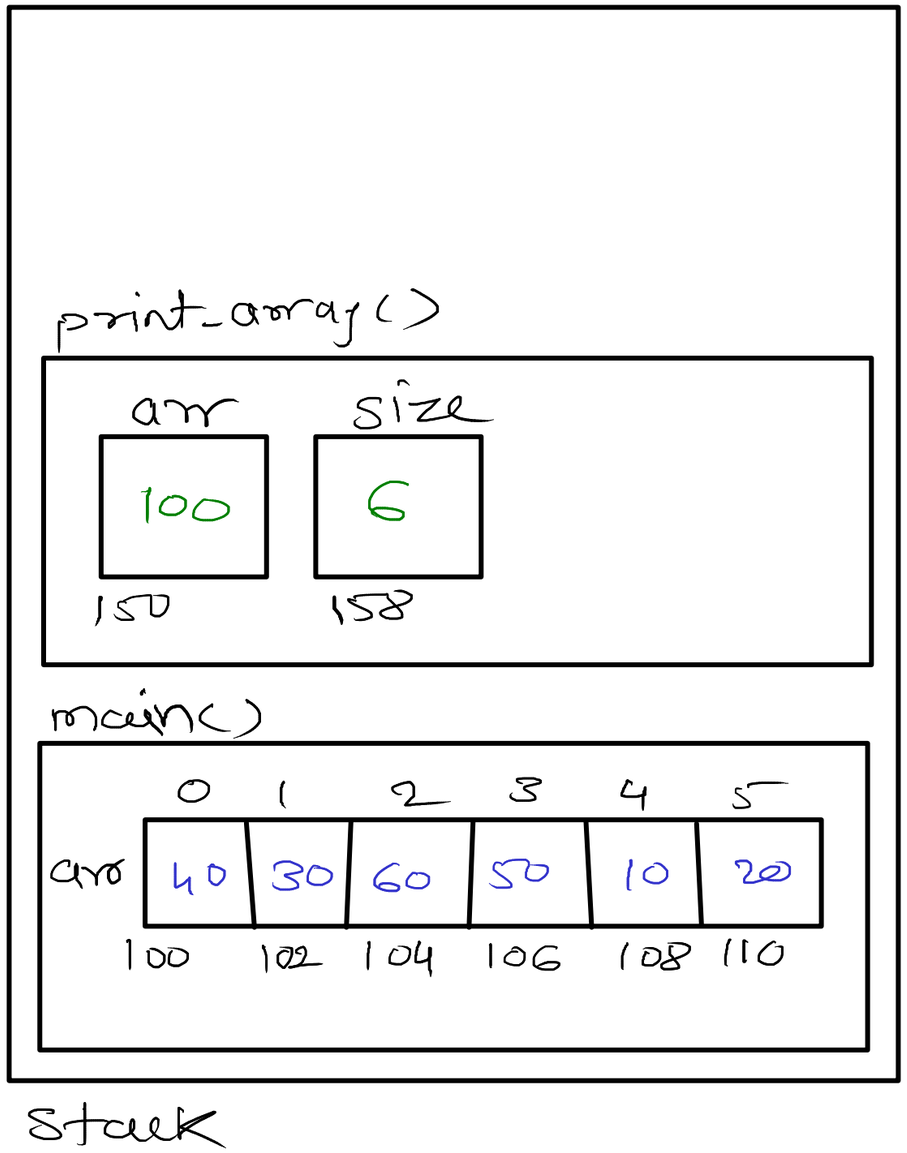
*(arr+i) <=> *(i+arr)
arr[i] <=> i[arr]

e.g arr[3] = 40
*(arr+3) = *(100+3)
= *(112) = 40

*(arr+i) → value
arr+i → address } ith element

&arr[i] => ~~&~~*(arr+i)

3[arr] = *(3+arr) = *(3+100)
= *(112) = 40


print_array()
arr
size
100
6
150
158
main()
0 1 2 3 4 5
arr
40 30 60 50 10 20
100 102 104 106 108 110

Stack

Selection Sort

40 30 60 50 10 20

30 40 60 50 10 20

10 40 60 50 30 20

10 40 60 50 30 20

10 30 60 50 40 20

10 20 60 50 40 30

10 20 60 50 40 30

10 20 50 60 40 30

10 20 40 60 50 30

10 20 30 60 50 40

10 20 30 50 60 40

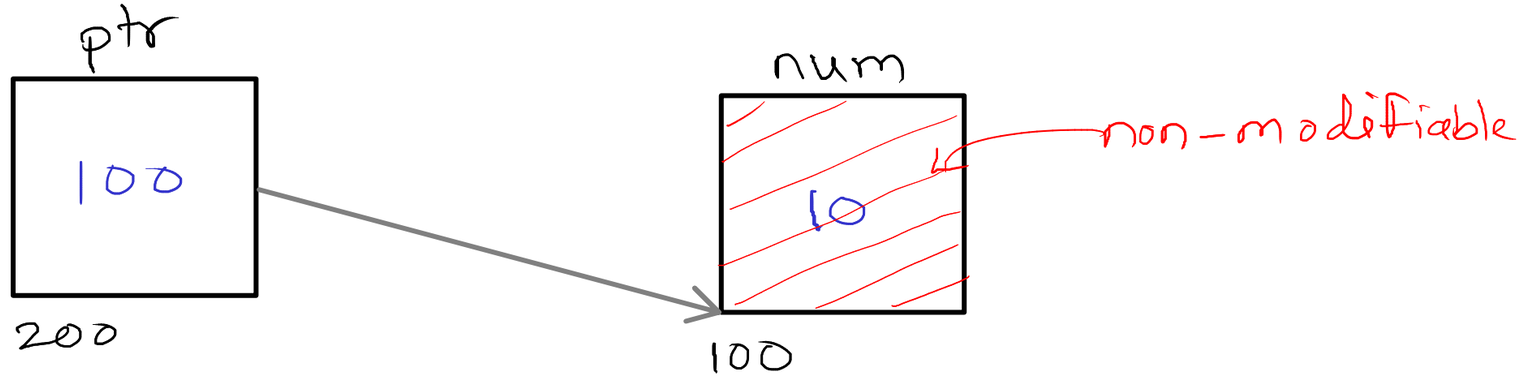
10 20 30 40 60 50

10 20 30 40 50 60

```
size=6
for(i=0; i<size-1; i++)
{
   for(j=i+1; j<size; j++)
   {
      if(arr[i] > arr[j])
      {
         swap(arr[i], arr[j]);
      }
   }
}
```

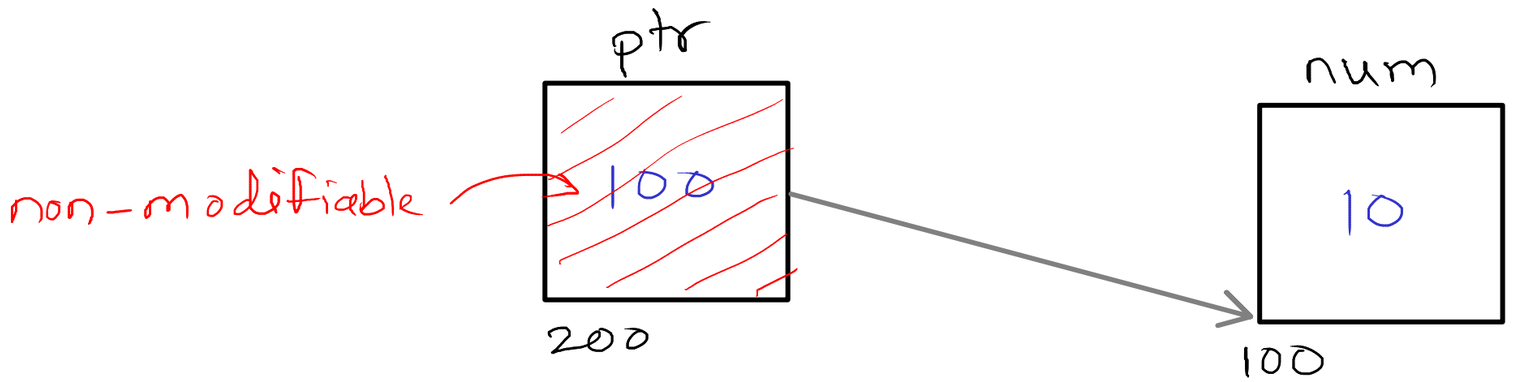
const int *ptr = #
or
int const *ptr = #

const int num = 10;
or
int const num = 10;



int * const ptr = #

int num = 10;



const int * const ptr = #

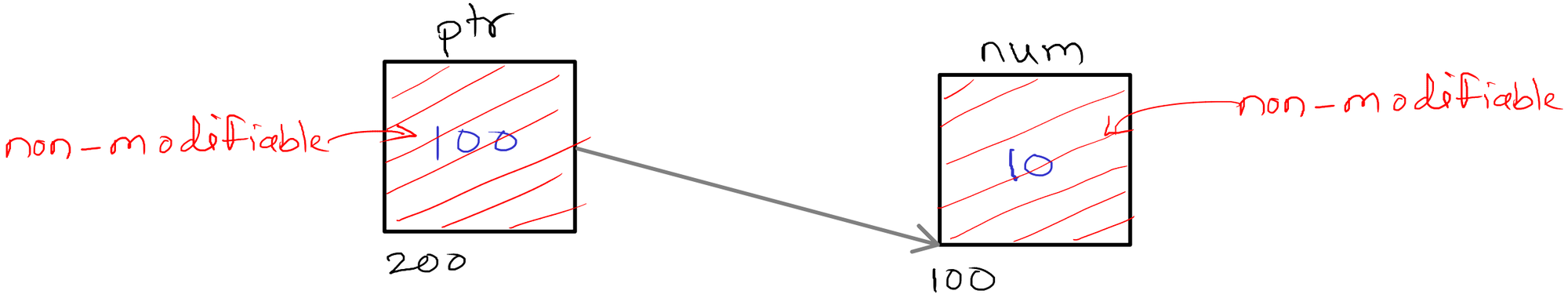
or

int const *const ptr = #

const int num = 10;

or

int const num = 10;

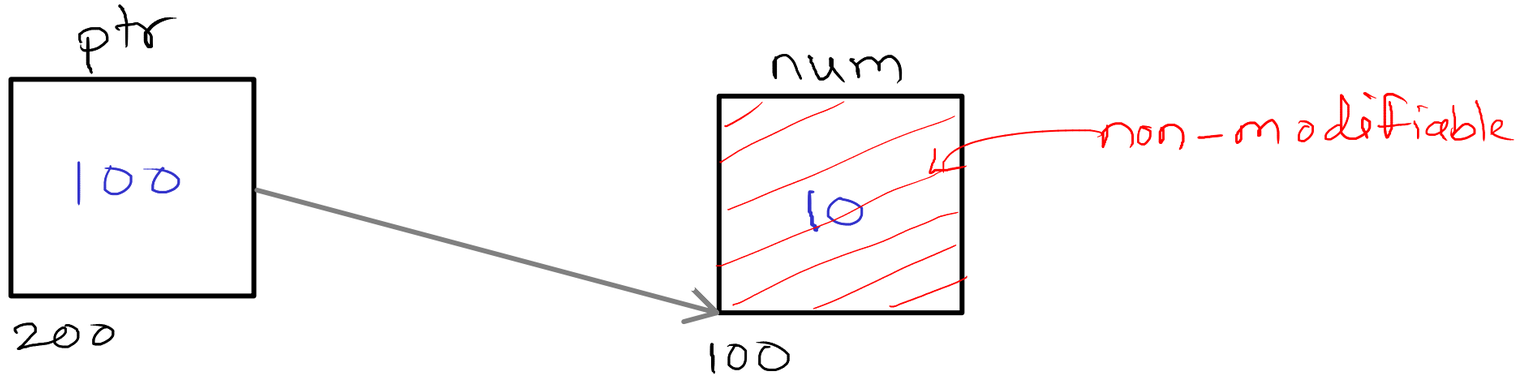


int *ptr = #

const int num = 10;

or

int const num = 10;



const int num
int const num
} — num is constant

const int *ptr
int const *ptr
} — location pointed by ptr is constant

int * const ptr → ptr is constant

int const * const ptr
const int * const ptr
} → ptr and location pointed by ptr, both are constant.