



'A'	'B'	'C'	NVL '0'	
65	66	67	0	

'ABC'

Q = = = '0'

'0'

0	
---	---

''

int const \* p1  
const int \* p1

p1 = &n; ✓  
\*p1 = 10; X

int \* const p2 = &n;

p2 = &n; X <sup>Cond  
fill</sup>  
\*p2 = 10; ✓

int const \* const p3;  
const int

p3 = &n; X <sup>Cond  
fill</sup>  
\*p3 = 10; X

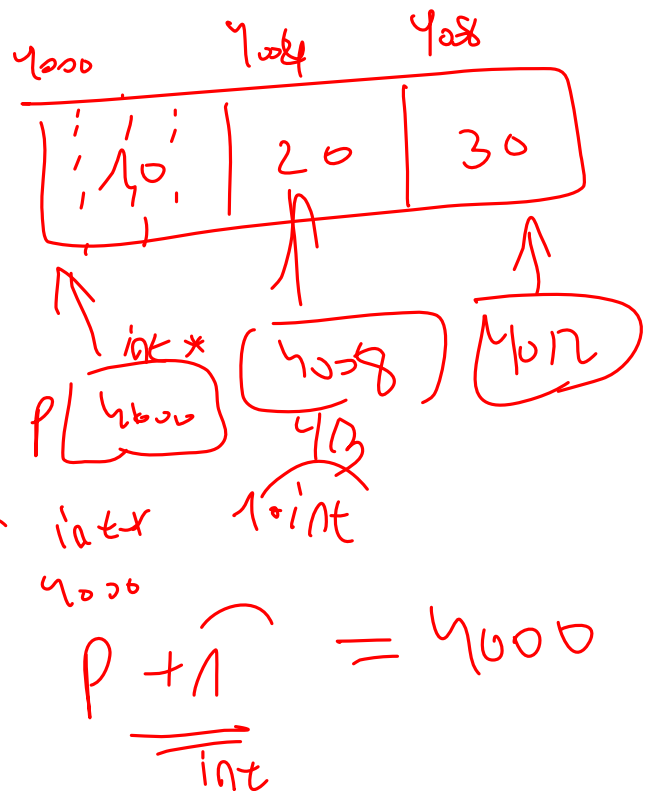
int a[3] = {10, 20, 30};

int \*p = &a[0];

int \*p = 0;

array  
decay  
into  
pointer

++p



2000

$$\overbrace{P} + 3$$

$$\Rightarrow 4012$$

$$\underbrace{3 \cdot 413}_{12}$$

$$p[i] \rightarrow * (p + i)$$

$4000$ 
 $1 \cdot 4$

$4004$

arr

arr[3]

4004
10

arr+i

↓

\* (arr+3)

&arr[3]

&\*(arr+i)

&arr[i]

~~&\*~~ (arr+3)

&arr[0]  $\rightarrow$  ~~&\*(arr + 0)~~  $\rightarrow$  arr

char arr[5] = "abcd";

char arr[5] = {'a', 'b', 'c', 'd', '\0'};

char arr[5] = {97, 98, 99, 100, 0};

4000 4001 4002 4003 4004

arr 

97	98	99	100	0
----	----	----	-----	---

(char\*)  
4000  
arr

4001  
arr+1

(char (\*)[5])  
4000  
arr

4005  
arr+5

```
printf("%i", *arr);
```

```
printf("%i", arr[0]);
```

```
printf("%s", (??)arr[2]); ← ?
```

char

$arr[0] \Leftrightarrow *(arr + 0) \Leftrightarrow *(arr) \Leftrightarrow *arr$



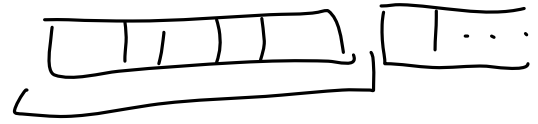
```
printf("%i", *arr);  
char (*p)[5]
```

char s [5] printf("%i", p)

char \*p2

char \*a[5]

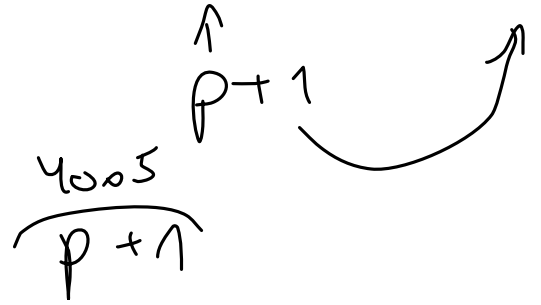
char \* p (5) printf("%i", a)



char str[5] = "abcd";

$p = \overbrace{\&str}^{4000};$

$p2 = (\text{char}*)p;$



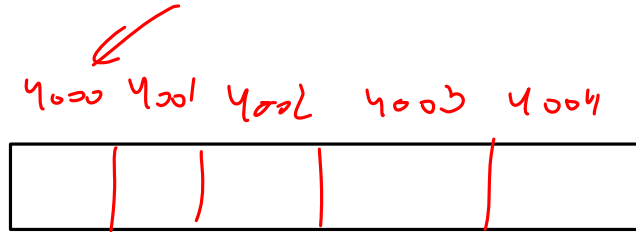
char arr[5] = "abcd";

char \*p = arr;

4000  
arr

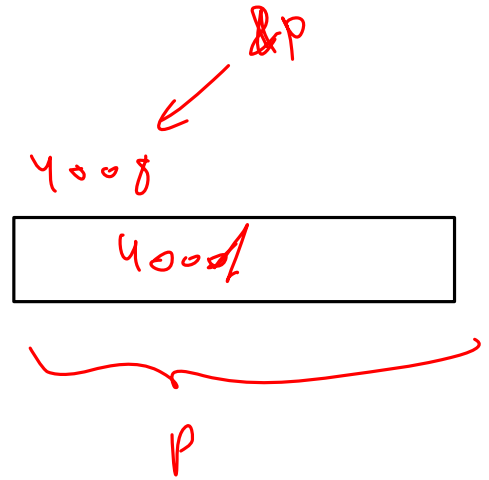
6000  
4000  
p

→ (jibk arr (&arr[0]))



arr

padding  
xxxx



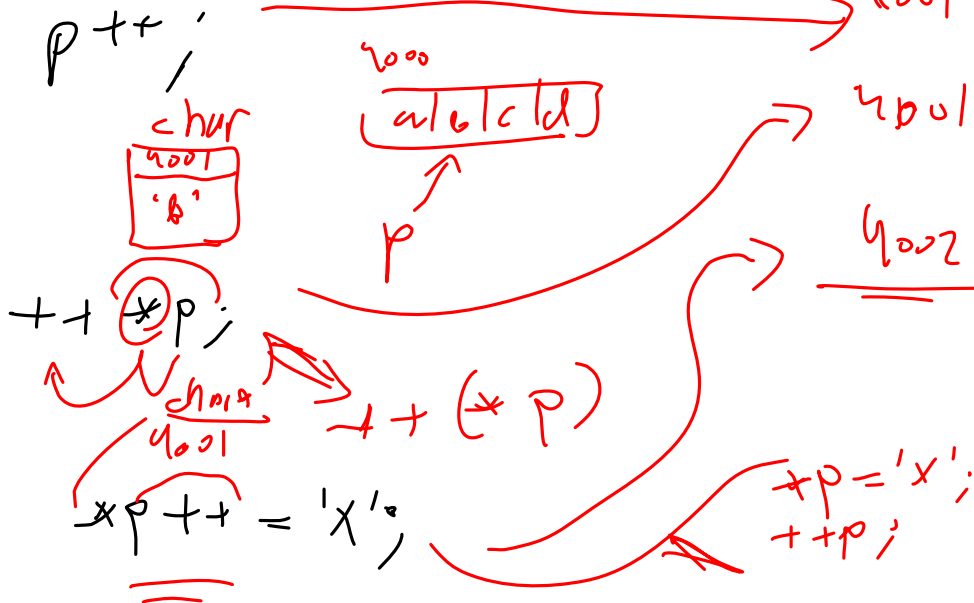
arr + 1  
⇒ arr = arr + 1

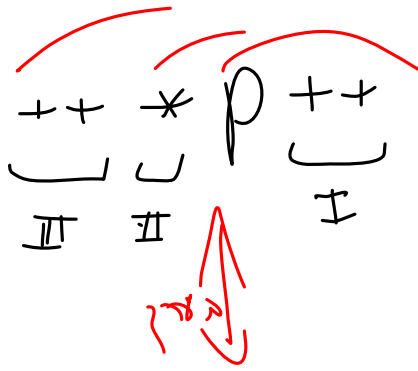
p + 1 ⇒ p = p + 1

char arr[5] = "abcd";

char \*p = arr;

p	arr 0	1	2	3	4
4000	a	b	c	d	0
4001	a	b	c	d	0
4001	a	c	c	d	0
<u>4002</u>	a	<u>x</u>	c	d	0





`++*p;`

`p++;`

`temp = p`

`++p`

`++ *temp`

```
printf("%i", *arr);  
char arr[5] = "abcd";
```

```
char *p = arr;
```

```
p++;
```

```
printf("%s", p);
```

bcd

python

s1 = "ABC"

ABC

s1 = "hello"

hello

s1 += "world"

hello world

print(s1)

C

char s1[1024] = "ABC";

strcpy(s1, "hello");

strcat(s1, "world");

puts(s1);



printf("%s\n", s1);

&s1[0]

strcpy(s1+6, "world");

int n1 = 10, n2 = 20;

int const \* p1;

p1 = &n1; ✓

p1 = &n2; ✓

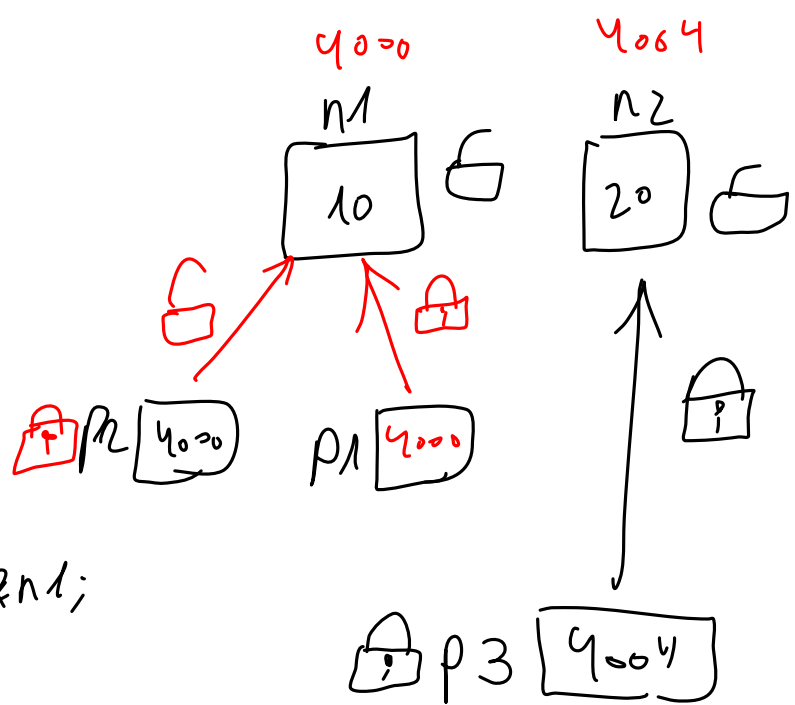
\*p1 = 30; ✗

int \* const p2 = &n1;

p2 = &n2; ✗

\*p2 = 30; ✓

const int \* const p3 = &n2



$\text{int}^* \rightarrow \text{int}$  מערך

$\text{char}^* \rightarrow$  מחרוזת

char\* \*  $\rightarrow$  מחרוזת מערך



UB Fun  $T2^* \xleftrightarrow{\text{error}} T^*$

long a[] = {10, 20};

int ap = (int\*)a;

f(p, a)

f(int \*p1, long \*p2)

strict aliasing

UB  
fun

T2\*

(מכונה)

←→  
(מכונה)

T\*

(מכונה)

→

←

(מכונה)

Void\*

↑  
מכונה  
↓

מכונה

↑  
מכונה  
↓

int a[10];

char\*

char\* p = (char\*)a;

Void\* v;

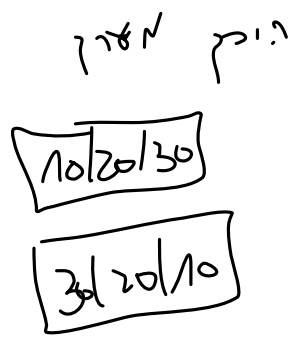
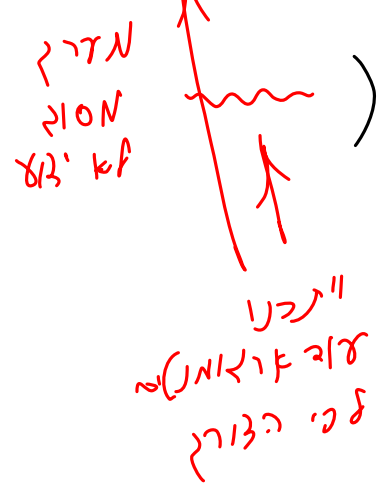
v = a;

p = v

void reverse\_ints (

int \*arr, int n)

void reverse\_arr (



void my\_Mem\_Cpy

void my\_Mem\_Equals