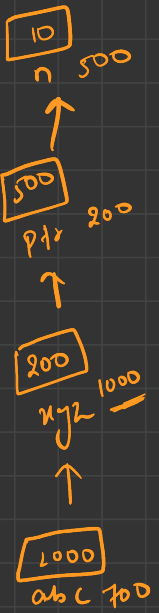
Double Pointer




```
int  n = 10;
int  *ptr = &n;
int  **xyz = &ptr;
int  ***abc = &xyz;
```

Made with Goodnotes

```
int n = 10;
int *ptr = &n;
int **xyz = &ptr;
```

*ptr = *ptr + 4;
= 10 + 4
= 14

*ptr

**xyz = *+xyz + 4;

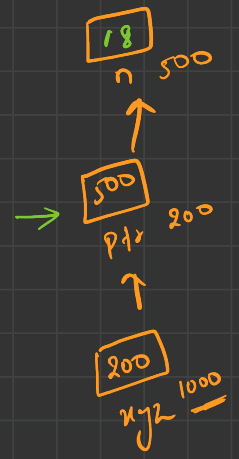
**xyz = 14 + 4

**xyz = 18

**xyz

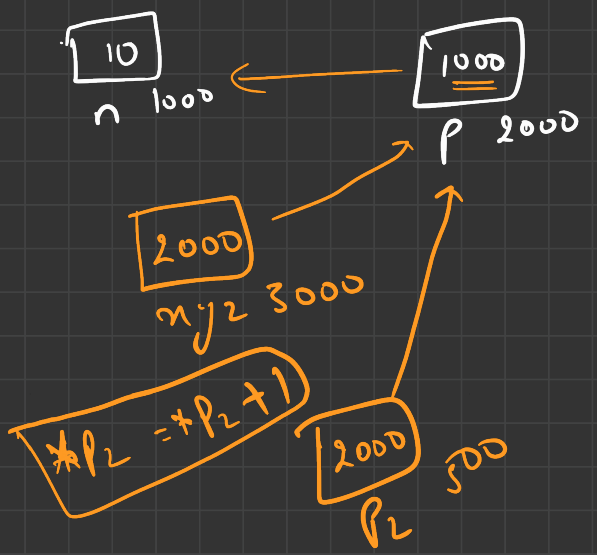
*500

18



Made with Goodnotes

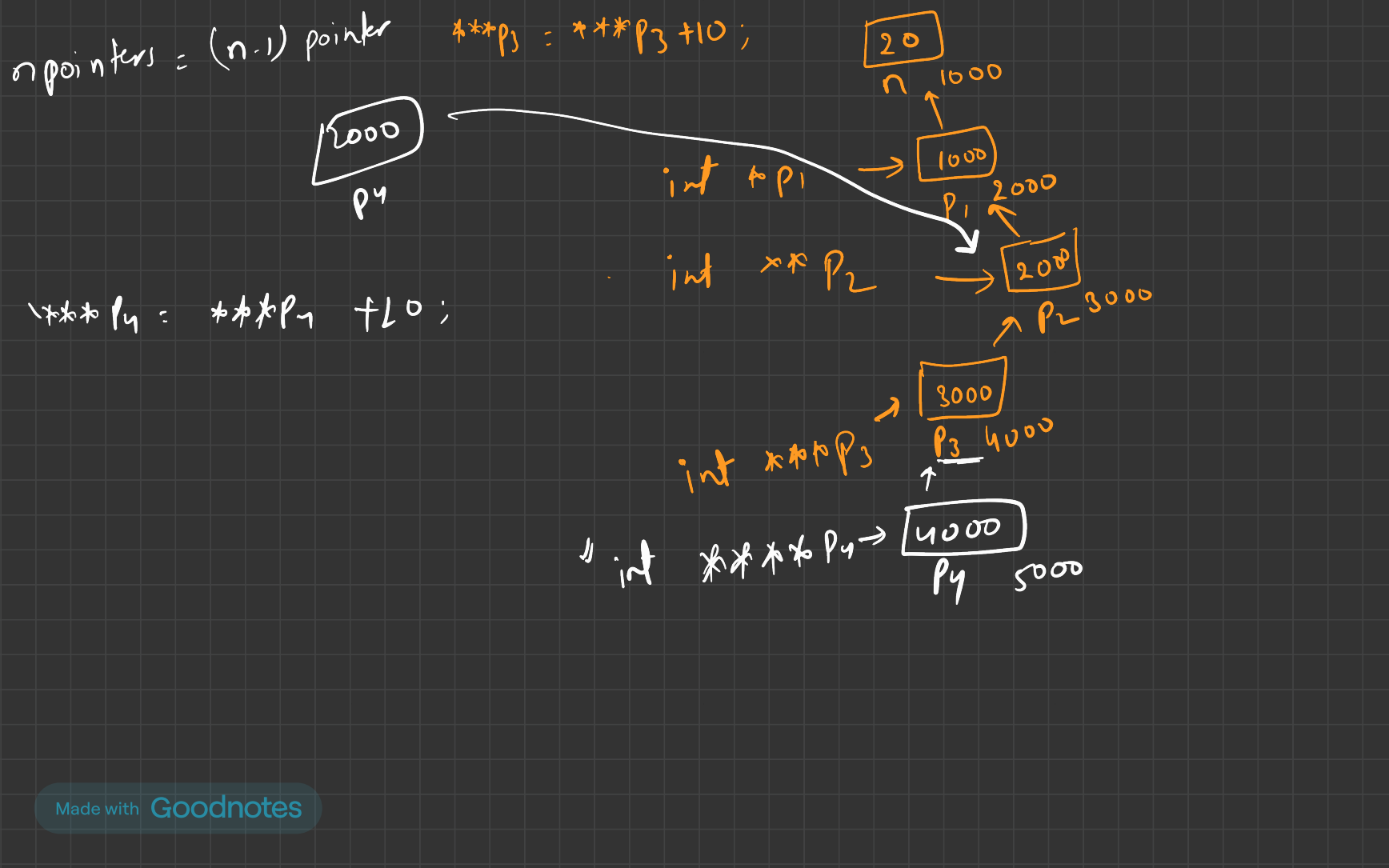
```
void fun(int **p2){
    *p2 = *p2 + 1;
}
int main(){
    int n = 10
    int *p = &n;
    int **xyz = &p;
    fun(xyz);
    cout << p;
}
```



How

Made with Goodnotes

n pointers = (n-1) pointer



Made with Goodnotes

```
int main()
{
        char c[ ]  = "GATE2025";
        char *p =c;
        cout<<p +p[3]-p[1];
}
```

cout<<c;

2024

Add 'E' - 'A'

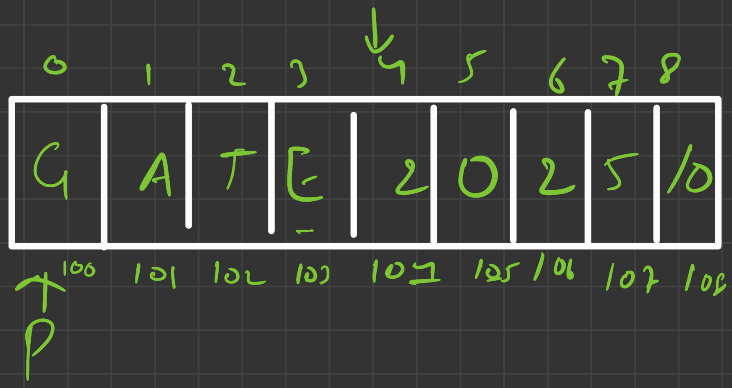
100 69 65

100 + 4

= 104

0	1	2	3	4	5	6	7	8
G	A	T	E	2	0	2	5	10
100	101	102	103	104	105	106	107	108

P



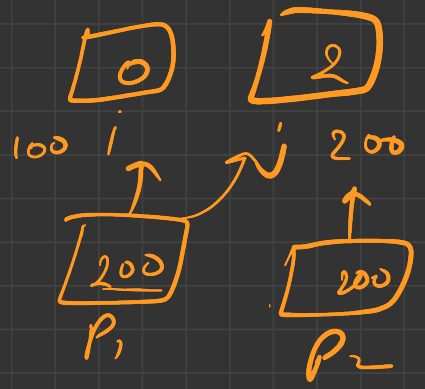
Add + value × size of data type

100 + 4 × 4

104

Made with Goodnotes

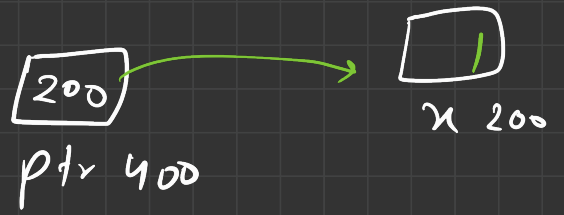
```
void see (int *p1, int *p2) {
    p1 = p2
    *p1 = 2;
}
int main () {
    int i=0, j=1
    see (&i, &j)
    cout << i << j;
}
```

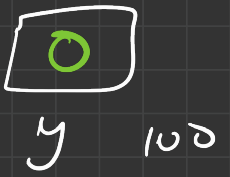


0 2

Made with Goodnotes

```
int *ptr ;
int  x = 0;
ptr = &x;
int y = *ptr;
*ptr = 1;
cout << x << y;
```



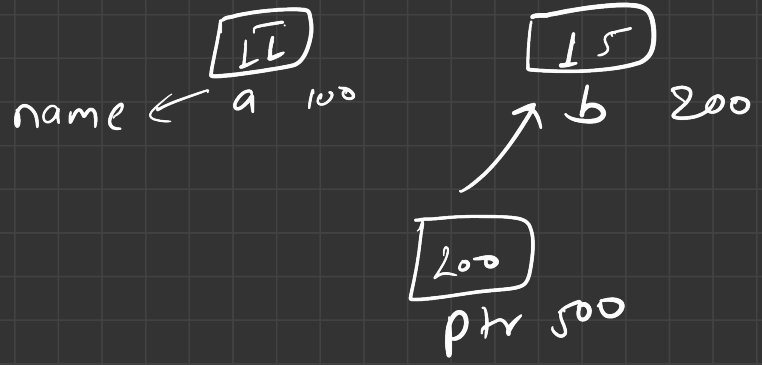


x = 1

y = 0

Made with Goodnotes

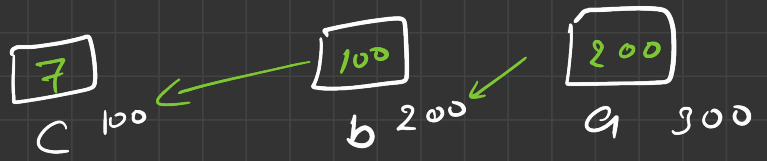
```
int a=5, b=10;
int &name =a;
int *ptr =&a;
(*ptr)++; ≡
ptr =&b;
*ptr = *ptr +5
name +=5;
cout << a <<b;
```

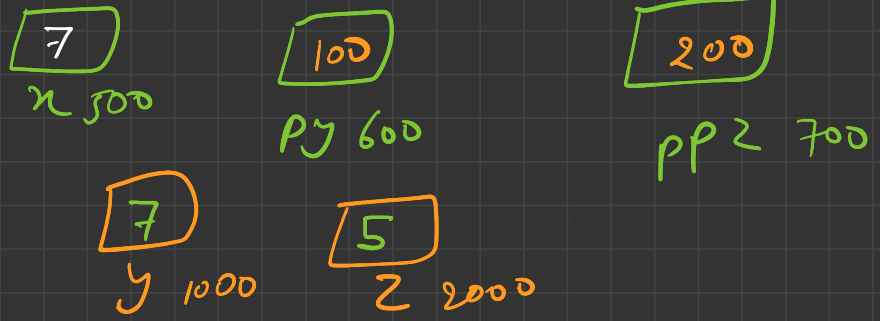


1

Made with Goodnotes

```
int four( int x, int *py, int **ppz){
        int y,z;
        **ppz +=1 ;
        z=**ppz;
        *py +=2;
        y=*py;
        x+=3;
        return x+y+z;
}   int main (
        int c,*b,**a;
        c=4,b=&c,a=&b;
        cout<<four(c,b,a);
  }
```





x + 7 + z

7 + 7 + 5 = 19

Made with Goodnotes

```
void five (char *str1, char*str2){
        while((*str1=*str2)){
        str1++, str2++;
        }
}

int main(){
        char first[ ]= "Mohit";
        char sec[ ]= " Rohan";
        five(first,sec);
        cout<<first;
}
```

→ 10

Str1 = 100+1×1 = 101

Str2 = 200+1×1 = 201

$\boxed{100}$ 500

first

Str1

R	O	H	A	N	6

100 ↑ ↑ ↑ ↑ ↑

sec

R	O	H	A	N	6

200 ↑ ↑ ↑ ↑ ↑

Str2

$\boxed{200}$ 1000

$\boxed{\text{ROHAN}}$

```
while (x=0) {
    cout << "Hi" ;
}
```

x = 1

k = 5

while (5) → non zero True

while (0)

Made with Goodnotes

Made with Goodnotes

Made with Goodnotes

Made with Goodnotes

Made with Goodnotes