If we want to access the last digit ... % 10.

But to remove last digit ...

/ 10

○ Transfer values:

```
int n1 = 10 ; int n2 = 20;

sout(n1);
sout(n2);
```

n1 = 10 20
n2 = 20 10

```
int n1 = 10;
int n2 = 20;

n1 = n2;
n2 = n1;

sout(n1);
sout(n2);

// 20
// 20
```

```
int temp = n1;
n1 = n2;
n2 = temp;
```

n1 = 10 20
n2 = 20 10
temp = 10

○ Short-circuit

```
if(x1 && x2 && x3 && x4){
    sout("All well");
}else{
    sout("No well");
}
```

```
if(x1 || x2 || x3 || x4){
    sout("Some well");
}else{
    sout("Nothing well");
}
```

629 % 10 → 9

629 % 100 → 29

629 % 100 = 29

9 + 29 =

○ Unary : i) minus
ii) not operator
iii)

○ Pre-increment vs post-increment:

○ In pre-increment.. values is incremented by 1 and then assigned.

○ In post-increment.. value is assigned and then incremented by 1.

○ Logical operator:

I) AND operator (&&) — ampersend sign

II) OR operator (||)

Q. age, name. If age > 18 .. Take license. If age ≤ 18 .. No license.
name is Rohan.. take license

Q: leap year: → Q. User year i/p
2020 → It's a leap year
19 → Not a leap year.

Feb → 28 days
leap year, Feb → 29 days.

I) 2022 → 4 table ... leap year.
II) 1900 → 100 table ... not a leap year.
III) 800 → 400 table ... leap year

200 → ✗
404 → ✓
900 → ✗
2000 → ✓

year = 2040

Q: Second greatest amongst 3:

i/p: n1 = 10 ; n2 = 20 ; n3 = 5
o/p: 10

i/p: n1 = 50 ; n2 = 10 ; n3 = 25
o/p: 25.

```
if((n1 > n2 && n1 < n3) || (n1 > n3 && n1 < n2)){
    sout(n1);
}
else if ( -- ){
}
else{
    sout(n3);
}
```

Q: Sum of last 2 digits of an user gn no:

int num = 39235.
o/p: 8

num = 32146
o/p: 10

num = 392

int v1 = num % 10;

num % 100 ⇒ ? ⇒ 92

629 ⇒ 600 + 20 + 9

600 + 20 + 9