CS & IT ENGINEERING

Programming in C

Control Flow Statements

Lec - 05

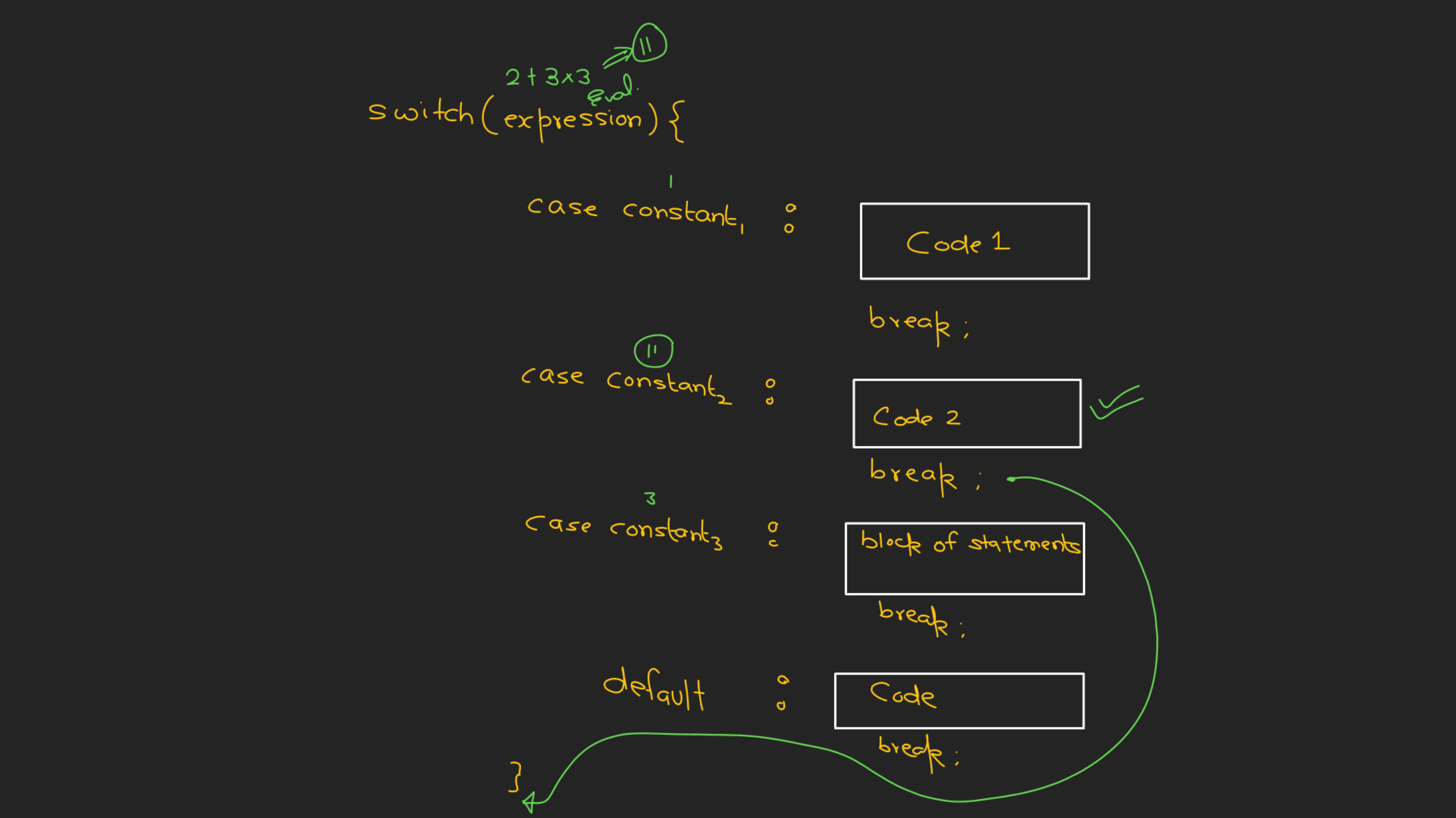
By- Pankaj Sharma sir

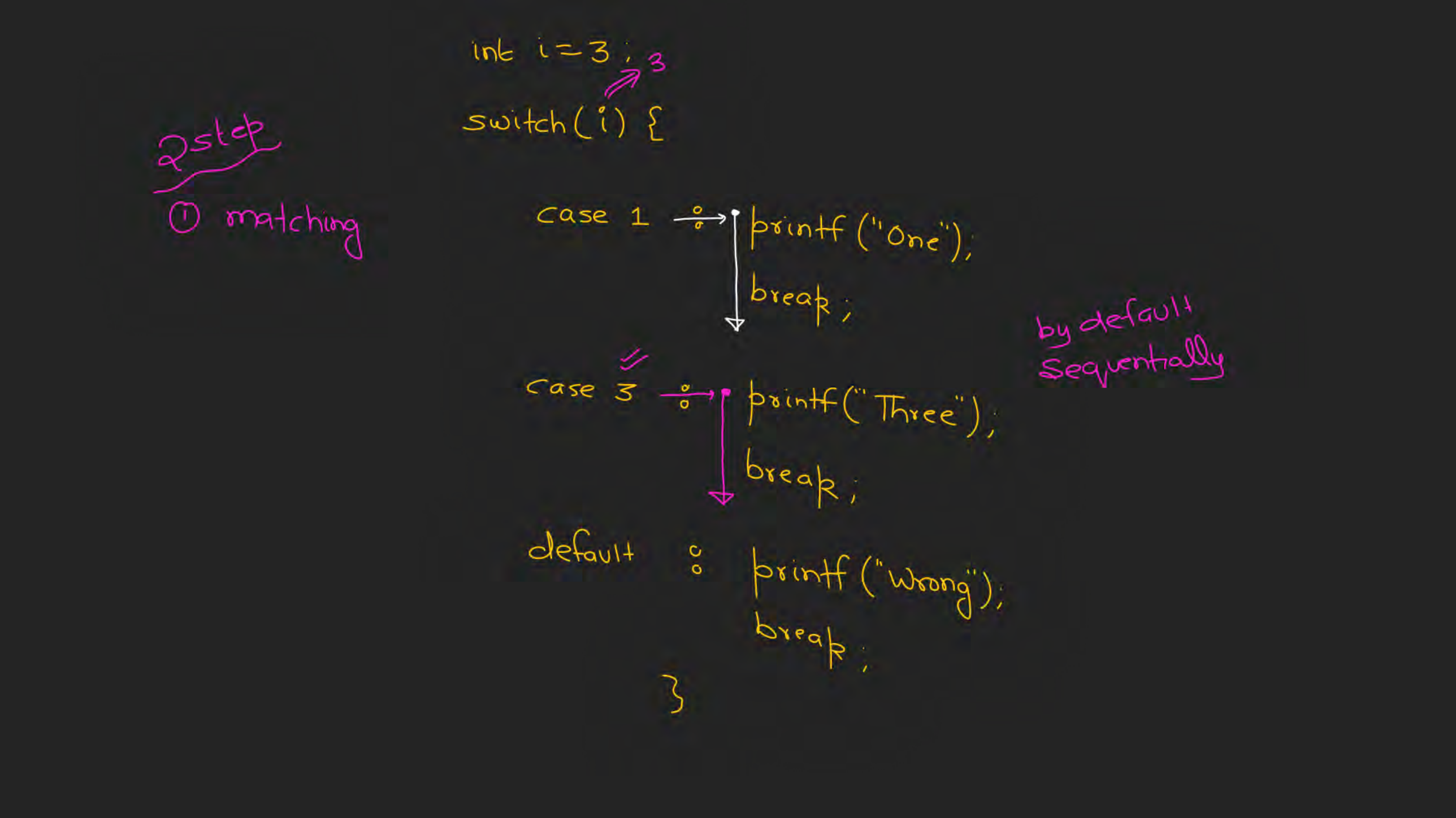

TOPICS TO
BE
COVERED
Switch Statements

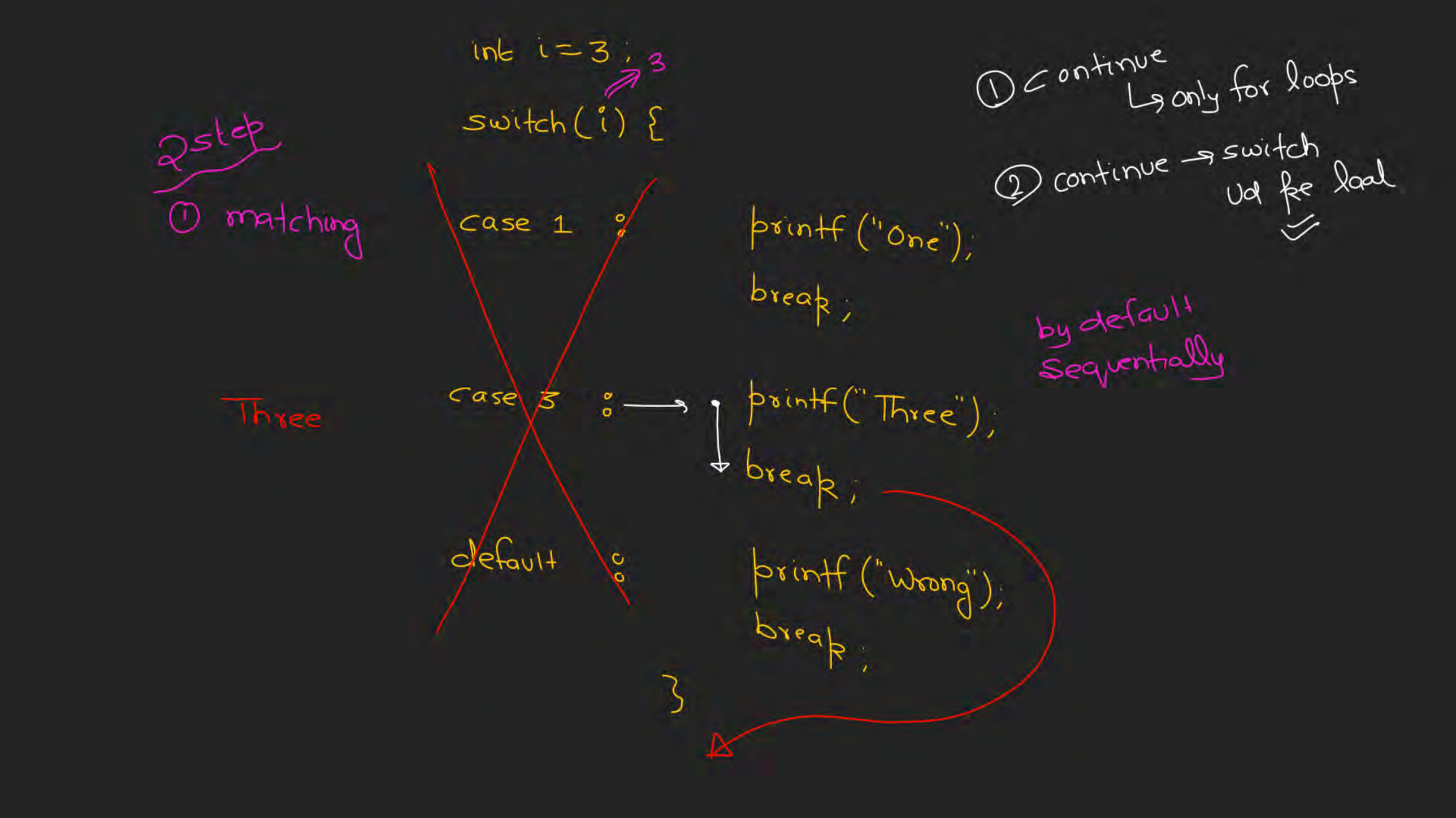
Switch statement

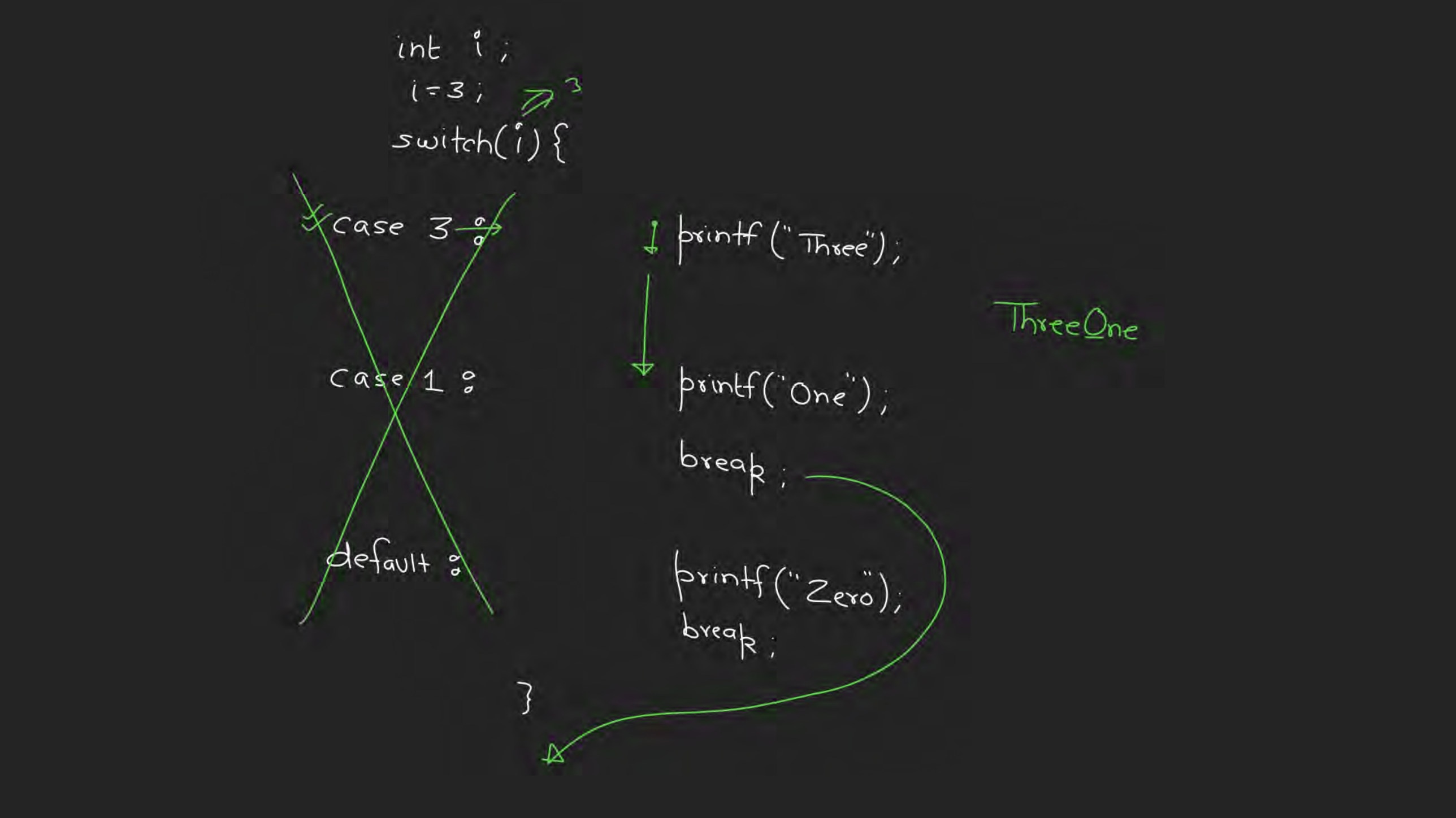
* switch is a keyword
* Used to provide selection statement with multiple choices.
* Multiple choices are provided using keyword : case

```
switch(n){
  case 1 :   Code to be executed when the value of
             n is 1.
             break;

  case 2 :   Code to be executed when the value of
             n is 2.
             break;

  default :  Code to be executed when n is not 1 or 2
             break;
  }
```


2+3×3
eval.
11
switch(expression){
1
case constant1 :
Code 1
break;
11
case constant2 :
Code 2
break;
3
case constant3 :
block of statements
break;
default :
Code
break;
}


int i=3;
3
switch(i) {
2step
① matching
case 1 :
printf("One");
break;
case 3 :
printf("Three");
break;
default :
printf("Wrong");
break;
}
by default
Sequentially


int i=3;
3
switch(i) {
2 step
① matching
case 1 :
printf("One");
break;
case 3 :
printf("Three");
break;
default :
printf("Wrong");
break;
}
Three
① continue
↳ only for loops
② continue → switch
Ud fe laal
by default
sequentially


int i;
i=3;
3
switch(i){
case 3 :
case 1 :
default :
printf("Three");
printf("One");
break;
printf("Zero");
break;
}
ThreeOne

switch(expression)

Expression that
Eval. to be int value

{

}

10

10 + 20

10 && 30

10 < 10

10 < 20 && 3 > 5

10 == 20

int a=1, b=2 a+b*3

17.38 → Undefined float

printf("Pankaj");

'A'

'A'+2

'A' ✓

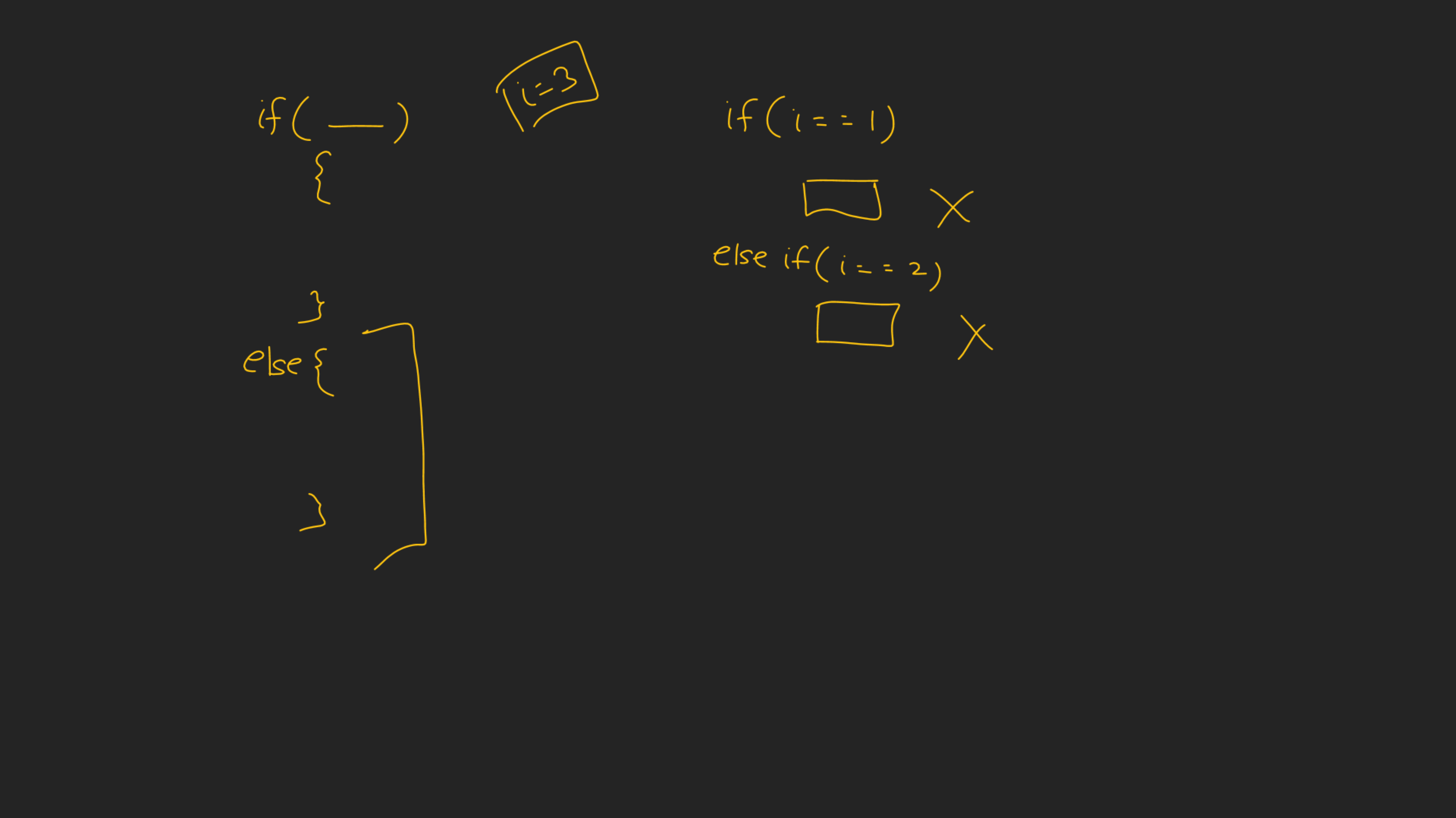
```
switch('A'){
Case 67 : Code
          break;
Code 65 : code
          break;
default : ___
          break
}
```

```
switch( printf("Anushka"))
{

case 1 : printf("1");
         break;

default : printf("Hello");
          break;

}
```

7

```
int i = 3;
switch(i+2) {
    case 5 : printf("5");
             break;
    case 1 : printf("1");
             break;
    case 3 : printf("3");
             break;
    default : printf("Hello");
              break;
}
```

```
int i=3;
switch(i+2){

    default : printf("0");
              break;

    case 5 : printf("5");
             break;

}
```


if(—)
{
i=3
if(i==1)
X
else if(i==2)
X
}
else {
}

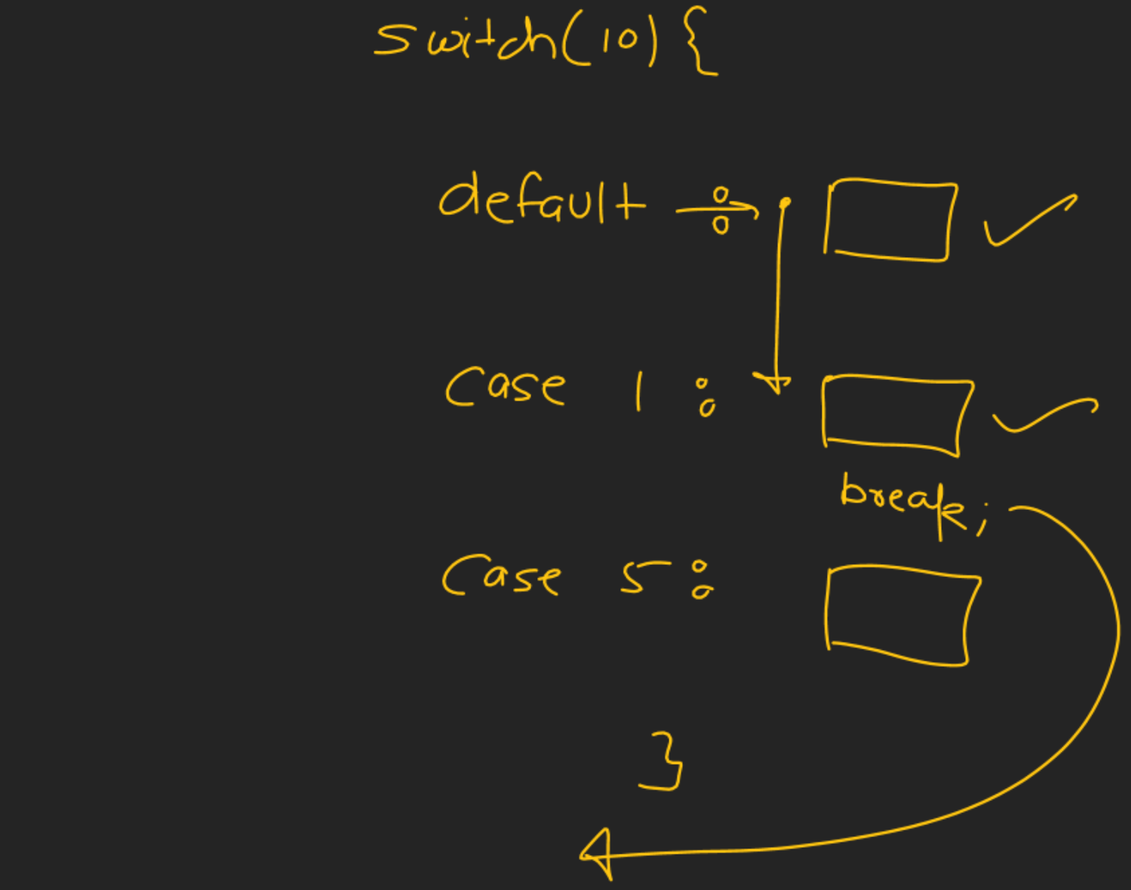
```
int i = 3;
switch(i+2){

    case 3 : printf("3");
             break;

    case 1 : printf("1");
             break;

}
```

```
int i=3;
switch(i) {

}
switch(i);
```

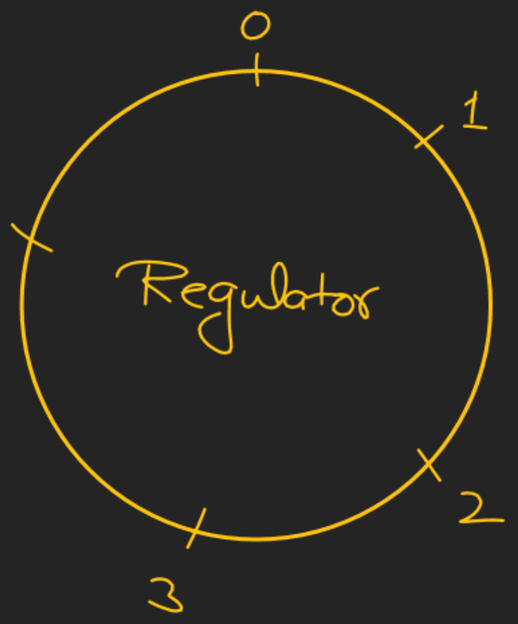
Dummy switch

```
int a=1,b=2;
Switch(3){

    case a :
        ==
        break;

    case a+b :
        ==
        break;

}
```

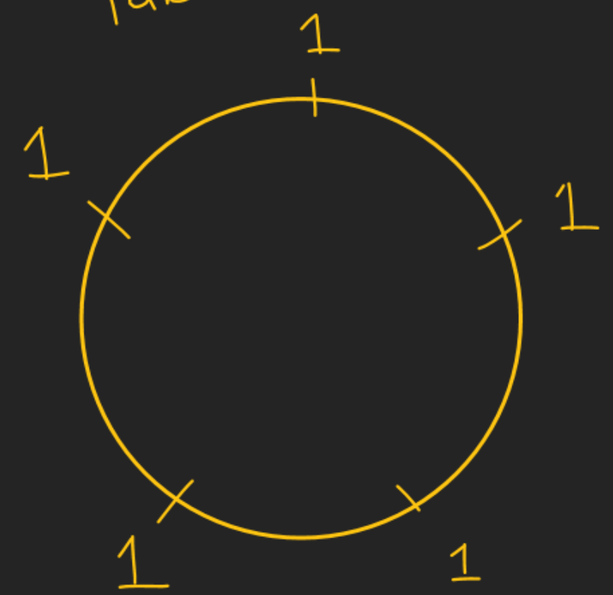
Case labels can not be variable. It can only contain constant/literal

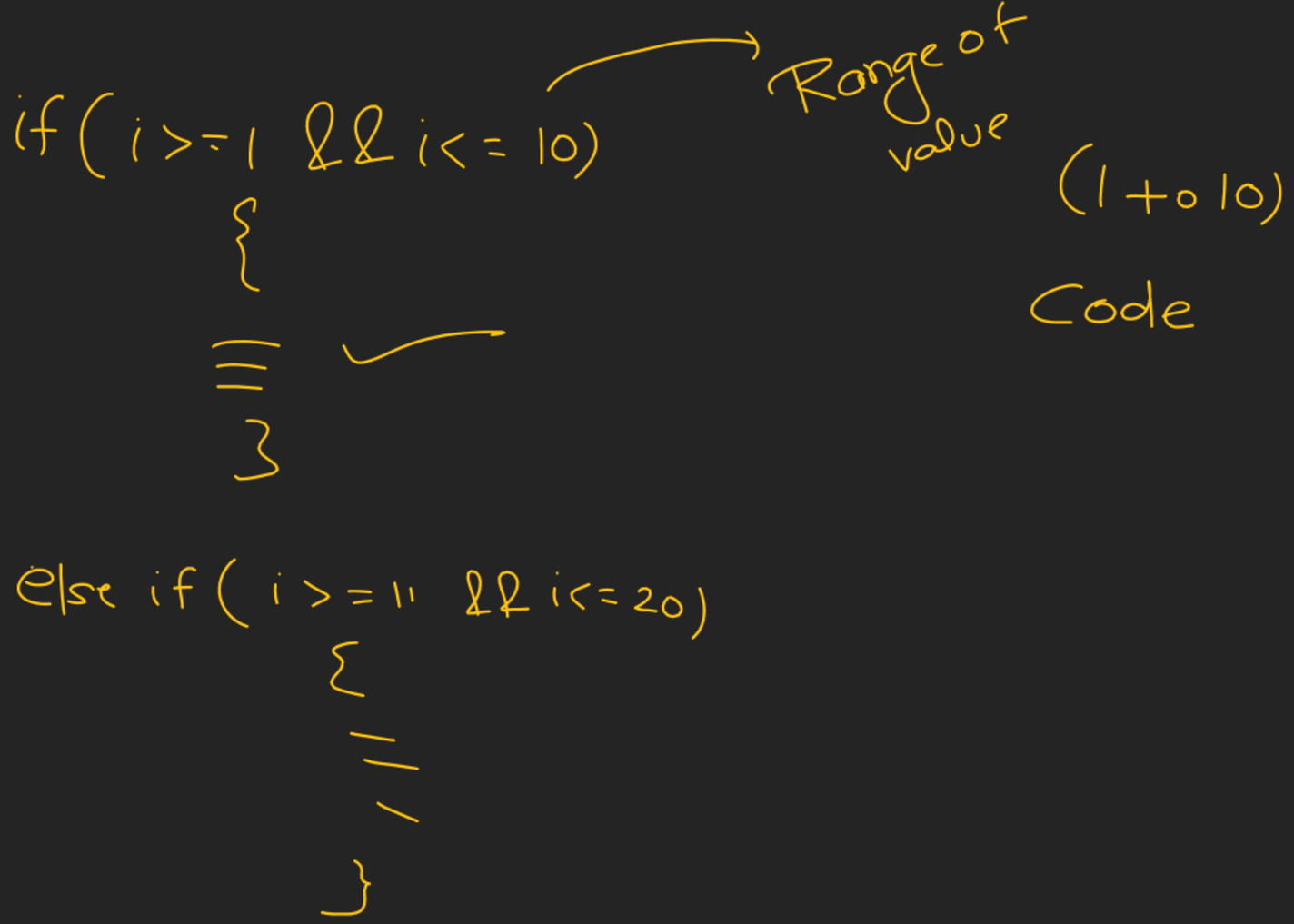

switch(10) {
default :
case 1 :
break;
case 5 :
}

```
switch(i) {

case 4: pf("4");
        break;

case 3+1: pf("four");
          break;
}
```



Duplicate case labels are not allowed.




if (i >= 1 && i <= 10)
Range of value
(1 to 10)
Code
{
}
else if (i >= 11 && i <= 20)
{
}

Not for all compilers

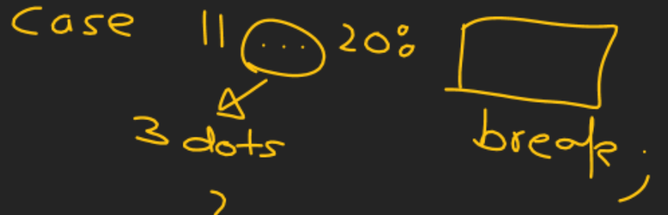
low, high

1 to 10 → code

switch(i){

case 1 ... 10:

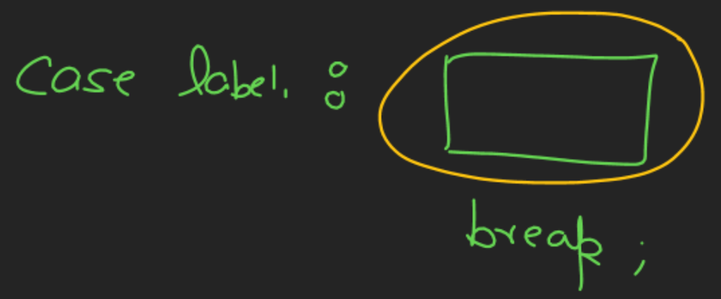
case 11 ... 20:

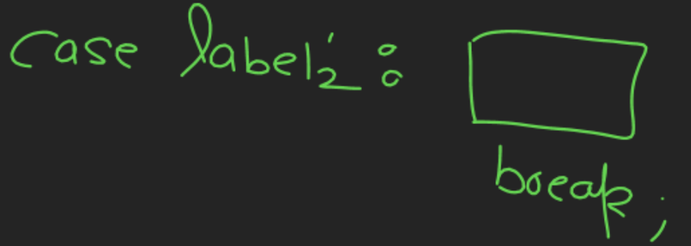


}

```
int i = 2;
switch(i){
    i = i+2;
case 2 : printf("2");
         break;
case 4 : printf("4");
         break;
}
```

ignore





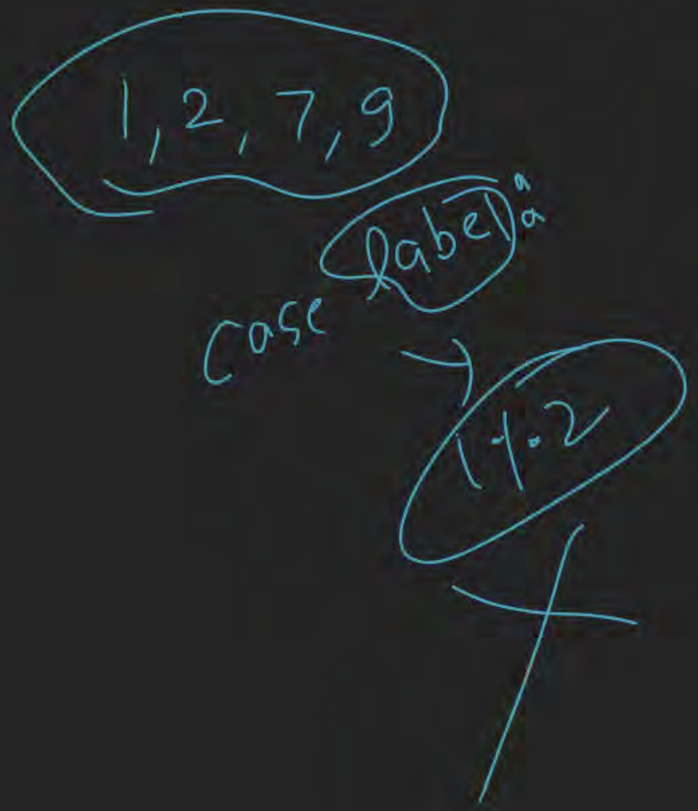
```
int i = 2;
switch(i+2) {
    case 2 : printf("2");
             break;
             printf("Hello");
    case 4 : printf("4");
             break;
}
```

ignore

4

set of values 1,3,5,7 ⇒ code

2,4,6 ⇒ code



switch

```
switch(i) {
case 1 :
case 3 :
case 5 :
case 7 : printf("Pankaj");
         break;
```

valid

case 5 % 2 == 0 :

case 4 :

duplicate

case ~5+1 :

update last

case ~-4+1 : 4

$\sim x = -(x+1)$

$\sim 5 = -(5+1) = -6$

$\sim 5+1 = -6+1$

$= -5$

$\sim -4 = -(-4+1) = -(-3) = +3$

$\sim -4+1 = 2+1 = 4$

i=1

a:3, b=2

0

a=((a==b)==0);

switch(i){

ignore a=a+10; X

}

a=(0==0)

a=1

i<<2+1

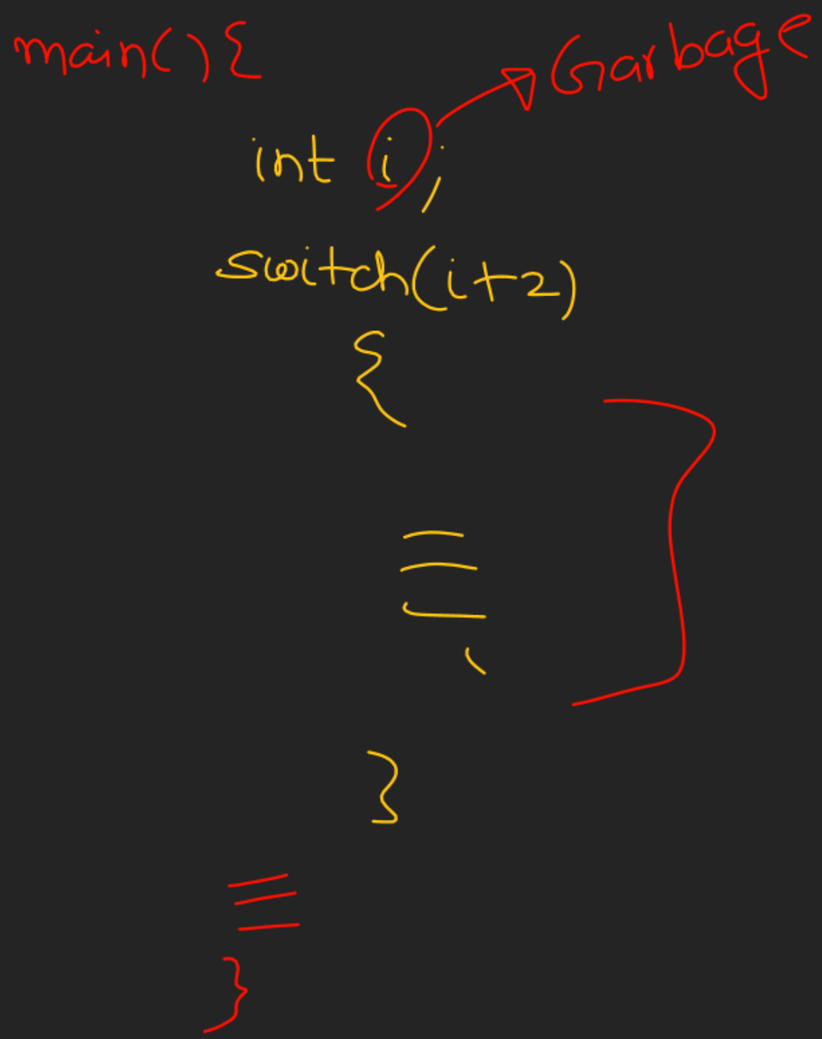
i<<3

1<<3

⇒ 8

==

=


main(){
Garbage
int i;
switch(i+2)
{
}
}

① break is optional.

② Order of case labels does not matter.

③ Position of default does not matter.

④ default is optional.

⑤ switch(i); ✓

int i=3 switch(i) {

✓

}

⑥ switch( ); ✗ Compiler ud fe load

Exp is not optional

⑦ case labels must be constant.

OR

any exp containing all constants

case 1 : ✓

case 1+2 : ✓

case 1<2 && 3>5 : ✓

case a : ✗

case a+2 : ✗

case printf( ) ✗

case 'A' : ✓

case 'a' : ✓

case 'A'+2 : ✓

Anna 24
Ghante
Chaukanna

(8) duplicate case labels not allowed


THANK YOU GW
SOLDIERS !