CS & IT ENGINEERING

Chapter 02

Programming in C

Control Flow Statements
Lec- 04

By- Pankaj Sharma sir

Switch-Case Statements

Switch statement

1) switch : Keyword used to provide selection statement with multiple choices.

It provide multiple choice using another keyword : `case`

```
switch(n) {
  case 1 :  Code we want to execute if the value of
            n is 1
            break;

  case 2 :  Code we want to execute if the value of
            n is 2.
            break;
  default :  Code we want to execute when the value of
             n does not match with any case label.
             break;
}
```

2 + 4 x 3 → Evaluate

```
switch (expression) {
    case constant1 :
        block of statements
        break ;
    case constant2 :
        block of statements
        break ;
    case constant3 :
        block of statements
        break ;
    default :
        break ;
}
```

Ex1

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

Eval ⇒ (3)

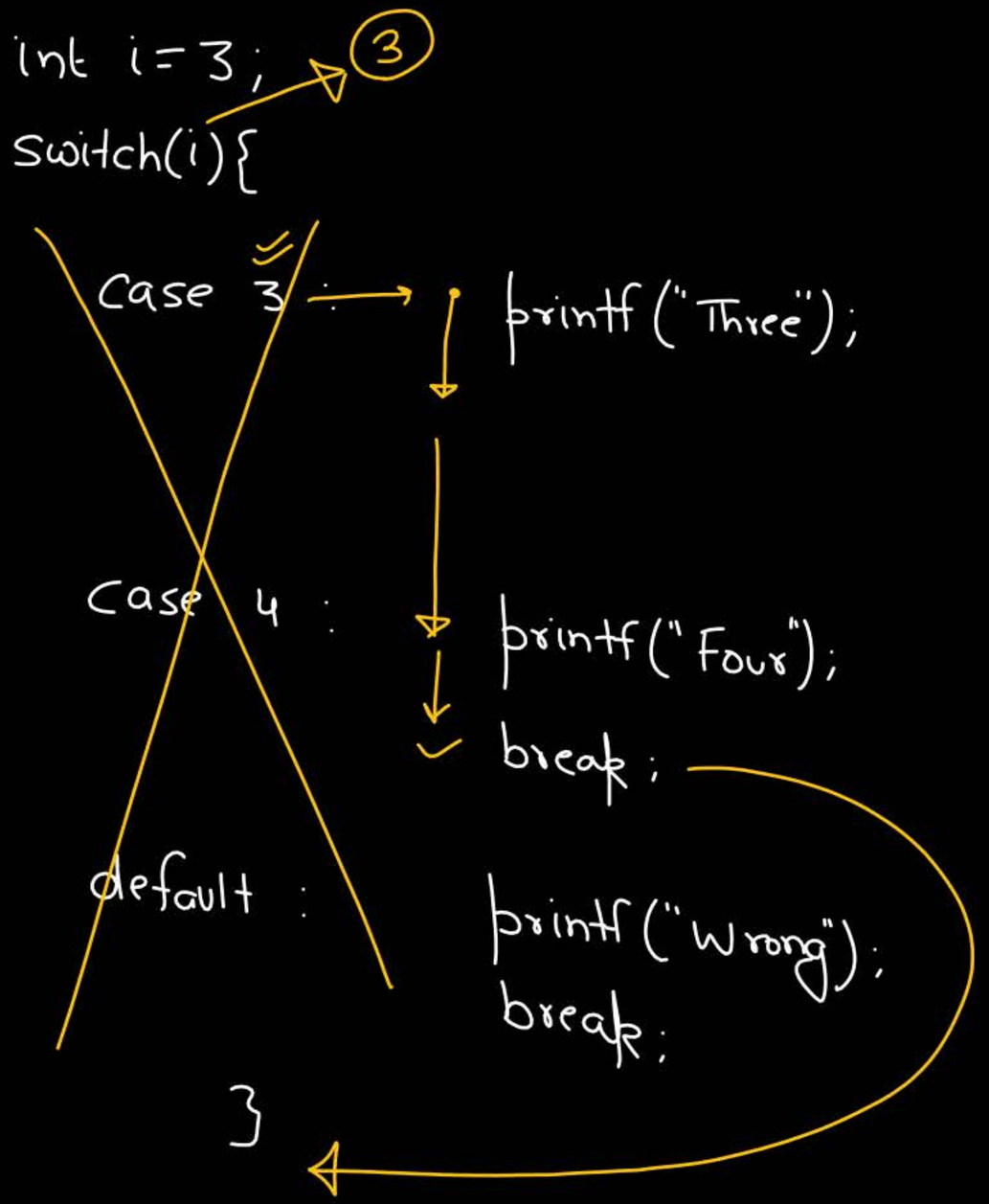
2 Steps

(i) Matching

Sequentially

Three

Ex2:



Three Four

Falling through cases

```
switch(i){
Case 1 :
        
        break;
Case 2 :
        
        break;
default :
        
        break;
}
```

Evaluated to be integer value

```
switch(expression)
{
 _
 _
 _
 _

}
```

(i) 2+3

(ii) 4/2

(iii) 1<2

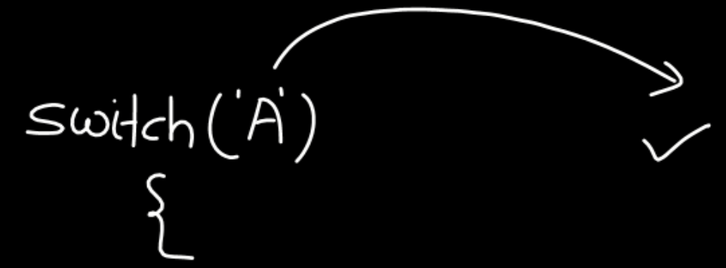
(iv) 2 && 7

```
int a=1,b=2
```

(v) a*b+1

(vi) printf("Pankaj") ✓

(vii) 17.38 ✗ Ud ke baat Marega



≲ 'A' + 2

value

67

case 65 :

=

Case 68 =

default : =

}

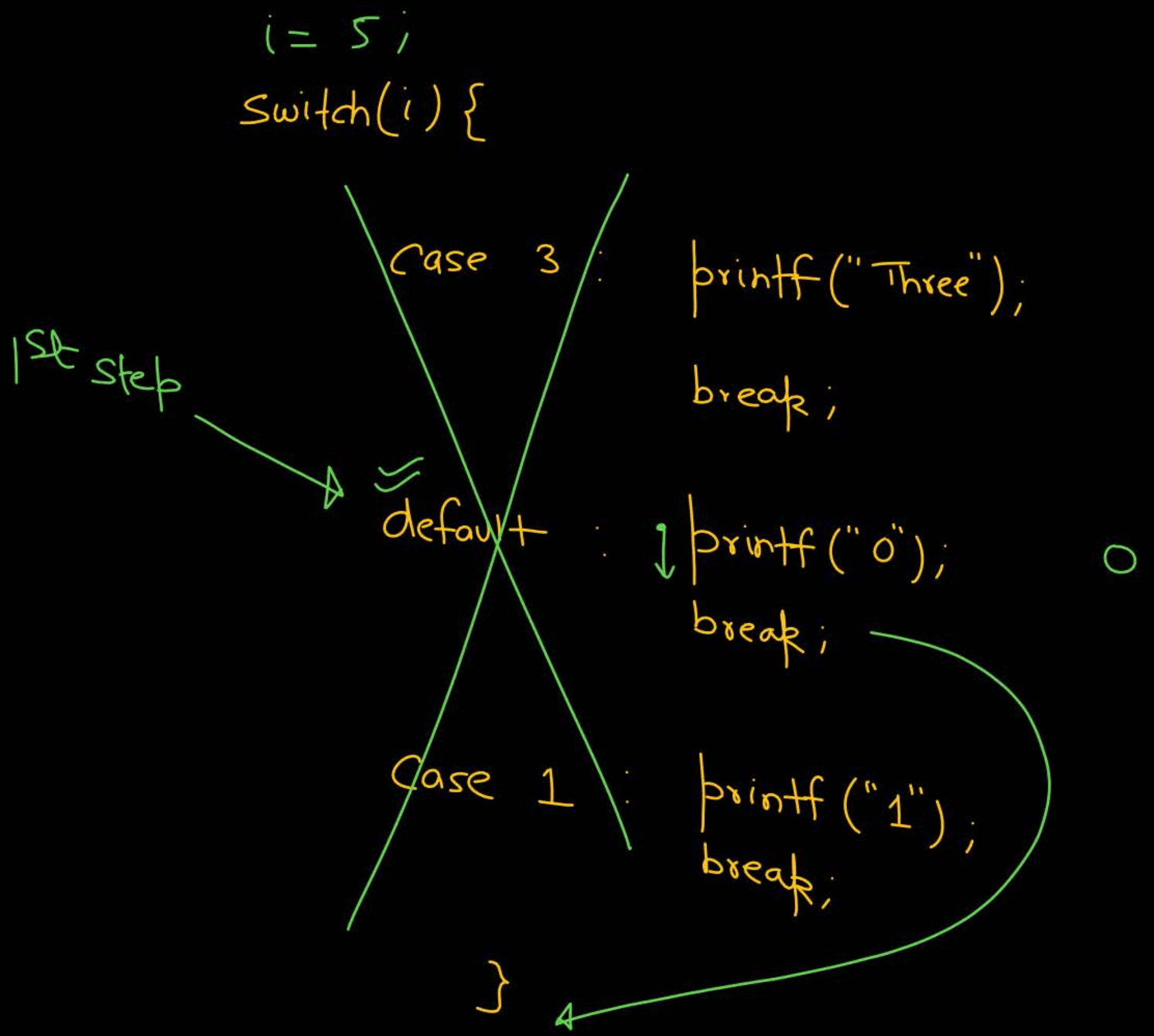
```
int i=3; 3
switch(i){
    case 5 : printf("five");
             break;

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

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

    default : printf("Wrong");
              break;

    }
```

Order of case labels does not matter.

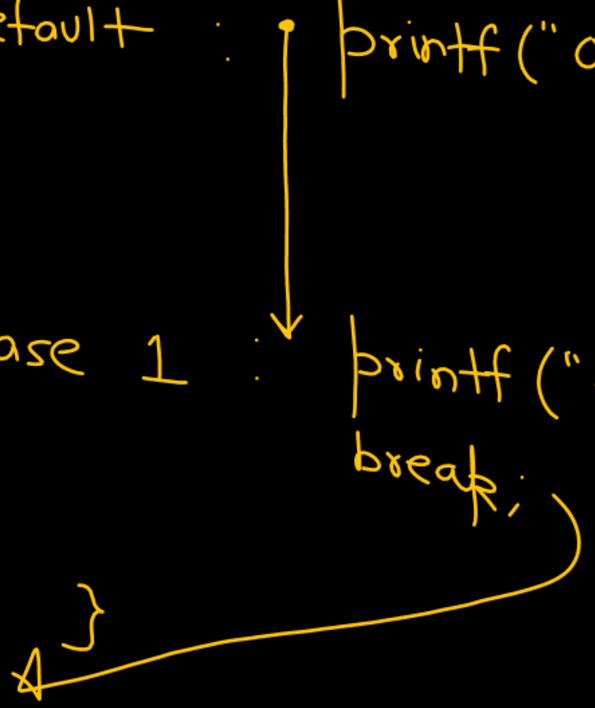
```
int i=3;3
Switch(i){

    case 4 : printf("Four");
             break;

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

}
```

default is optional

0 + 0.0 ⇒ floating type, int


i = 5;
switch(i) {
case 3 :
printf("Three");
break;
1st step
default :
printf("0");
break;
0
case 1 :
printf("1");
break;
}

```
i = 5;
switch(i) {

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

o/p: 01       default :  printf("0");

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



```
switch(i) {
```

Dummy

valid ✓

```
}
```

```
switch(i);
```
✓

```
switch( );
```
; Compiler ud fe laat marega

Case labels

```
int a=1, b=2;
switch(3+5){
    case a :
        break;
    case b :
        break;
}
```

Can not be variable

(constant/literals)

label :

printf(—) ✗

variable ✗

a + b x 2 ✗

int: constant/literal

2+3 ✓

2 ✓

3x4+2 ✓

2+a ✗

int a=1;

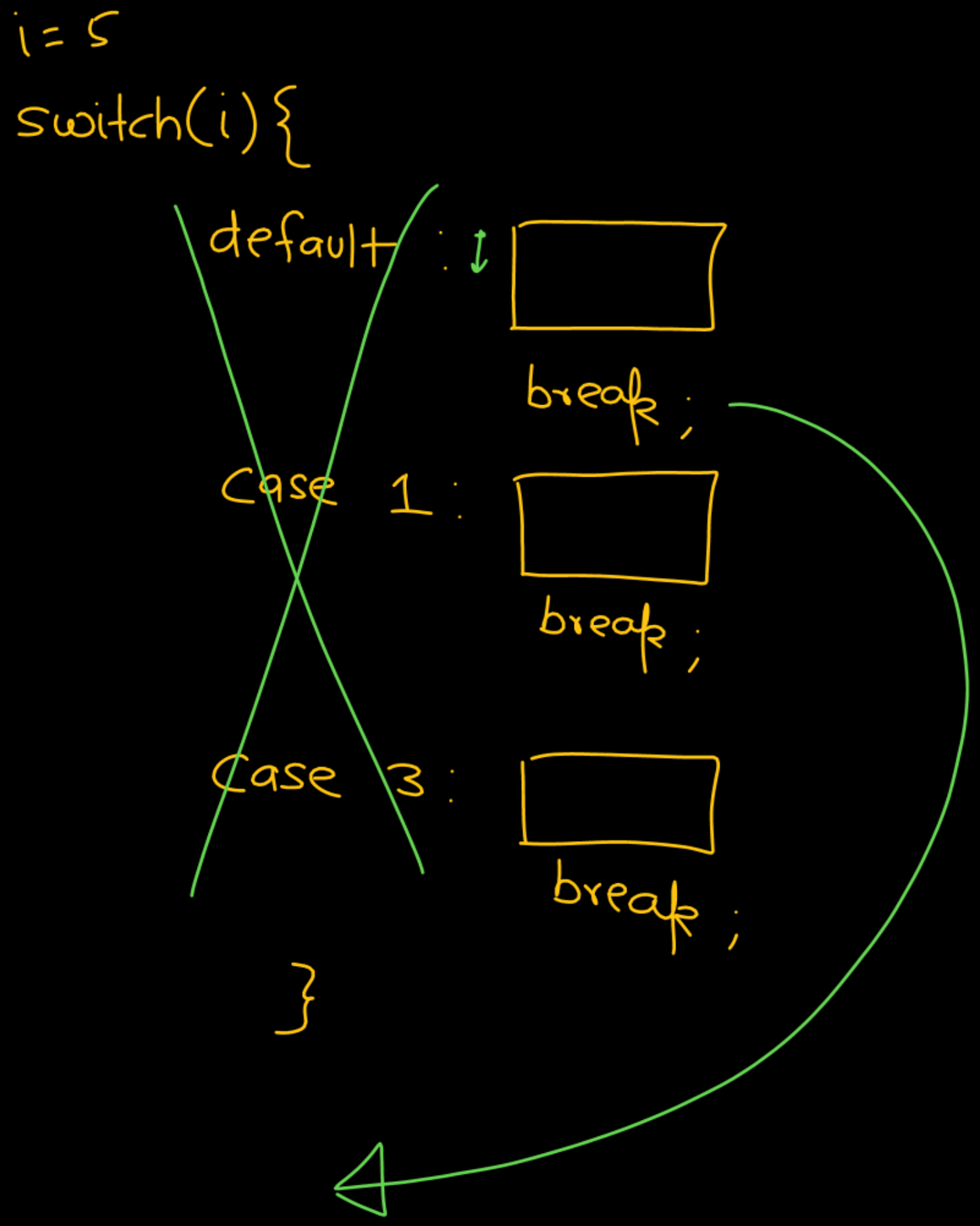
① case 'A' ✓

② case 'a' + 2 ✓

③ case 3+4×2 ✓

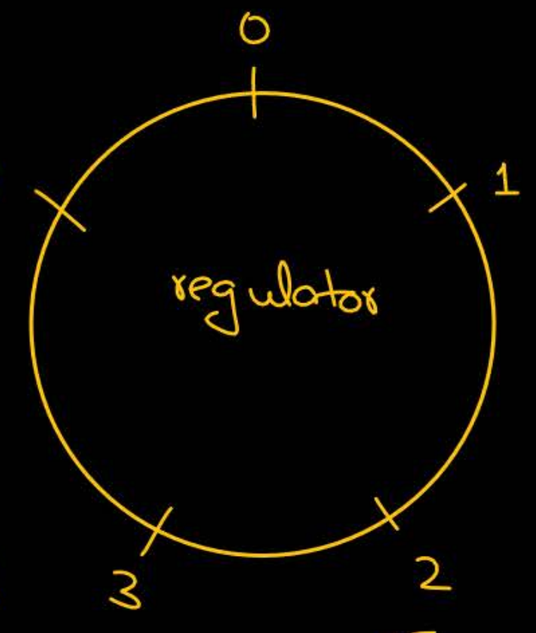
④ case !2+4 ✓

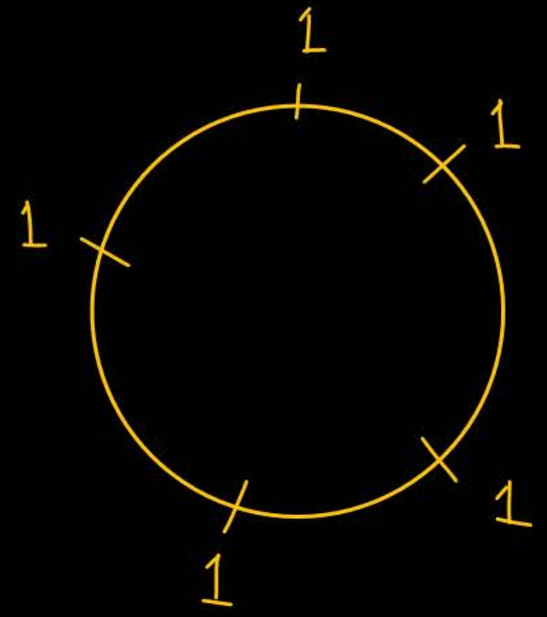
⑤ case printf("Hello")+2 ✗


i = 5
switch(i){
default :
break ;
case 1 :
break ;
case 3 :
break ;
}

```
i=2
switch(i+3){
case 4:
    break;
case 3+1:
    break;
}
```

Ud ke baat maanega





{ Duplicate case labels not allowed }

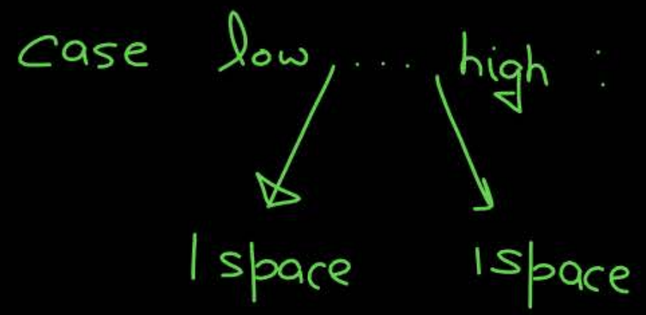
```
if ( i >= 10 && i <= 100)
        {

          code

        }
else if ( i >=200 && i <= 300)
        {

          code

        }
```

range of values ⇒ same code

Not for all compilers ✓ not for Gate

1 → 10 →

case low ... high :




```
switch(i){

case 1 ... 10 : printf("Hello");
                break;

case 11 ... 20 : printf("Pankaj");
                 break;

}
```

```
int i=3;
switch(i){
    case 4 : printf("4");
    case ~5+1 : printf("3");
    case ~-4+1 : printf("1");
    case 2&&3 : printf("0");
}
```

Duplicate case labels not allowed

4

Anna 24 Ghante

Chaukanna

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

$\sim -4 \Rightarrow -(-4+1)$

$= -(-3)$

$= 3$

$\sim -4 + 1$

$3 + 1 = 4$

```
switch(exp){
    case label1: Code
                 break;
    case label2: Code
                 break;
    case label3: Code
                 break;
    }
```

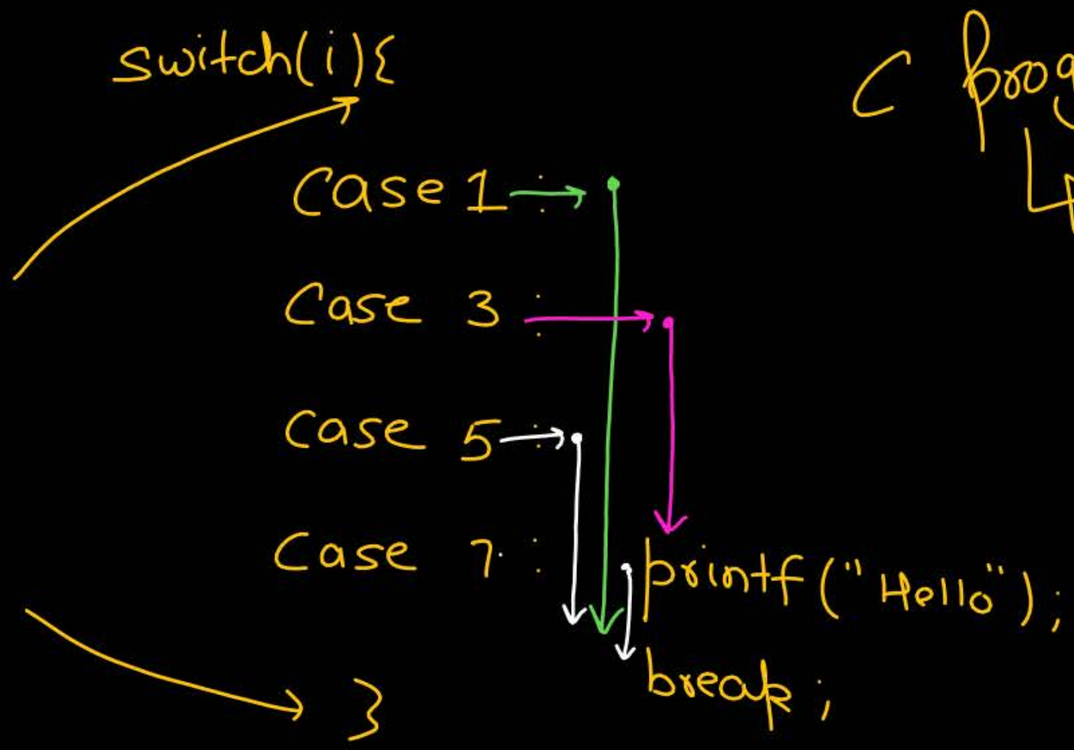
```
int i=2;
switch(i) {
        i = i+2;
Case 2 : printf("2");
         break;

Case 4 : printf("4");
         break;
}
```

ignore this

1, 3, 5, 7 ⇒ same code

2, 4, 6 ⇒ same code

Set of values ⇒ same code



C programming Any complaint

case 1 ... 10 :
8,9,10

Case 8 ... 20
8,9,10

duplicate case label

{
scope
}

printf( ){

}

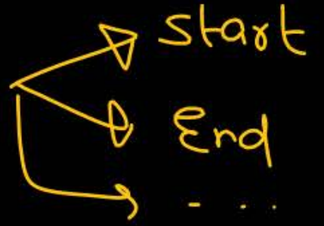
1) break is optional.

2) Expression must be eval. to integer

3) Order of case labels does not matter.

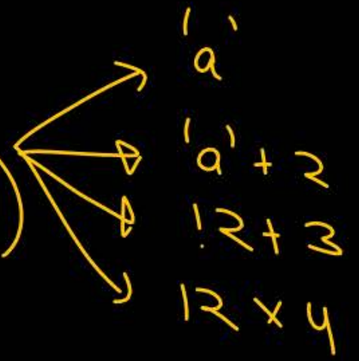
4) default is optional.

5) Position of default does not matter



6)
```
switch(exp);    (Dummy switch)
switch( );  → Error
```

7) Case labels can not be variables (constant/literal)



8) Duplicate case labels are not allowed.


THANK YOU GW
SOLDIERS !