TOPICS TO BE COVERED

Iterative Statements
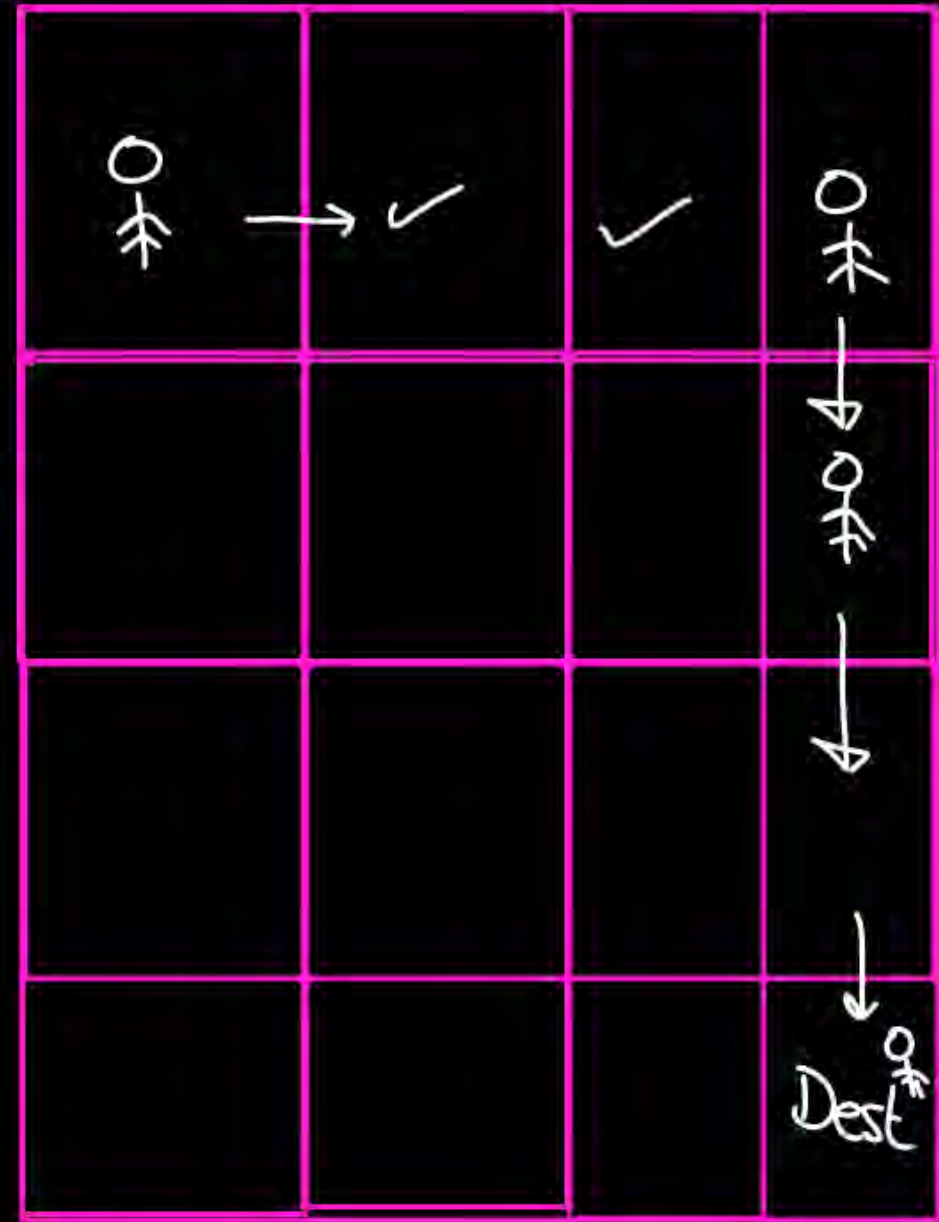
Iterative statements (Repeatition)

1st way

{
Go right
Go right
Go right

Go down
Go down
Go down
}

Right, down

2nd way

Go right 3 times

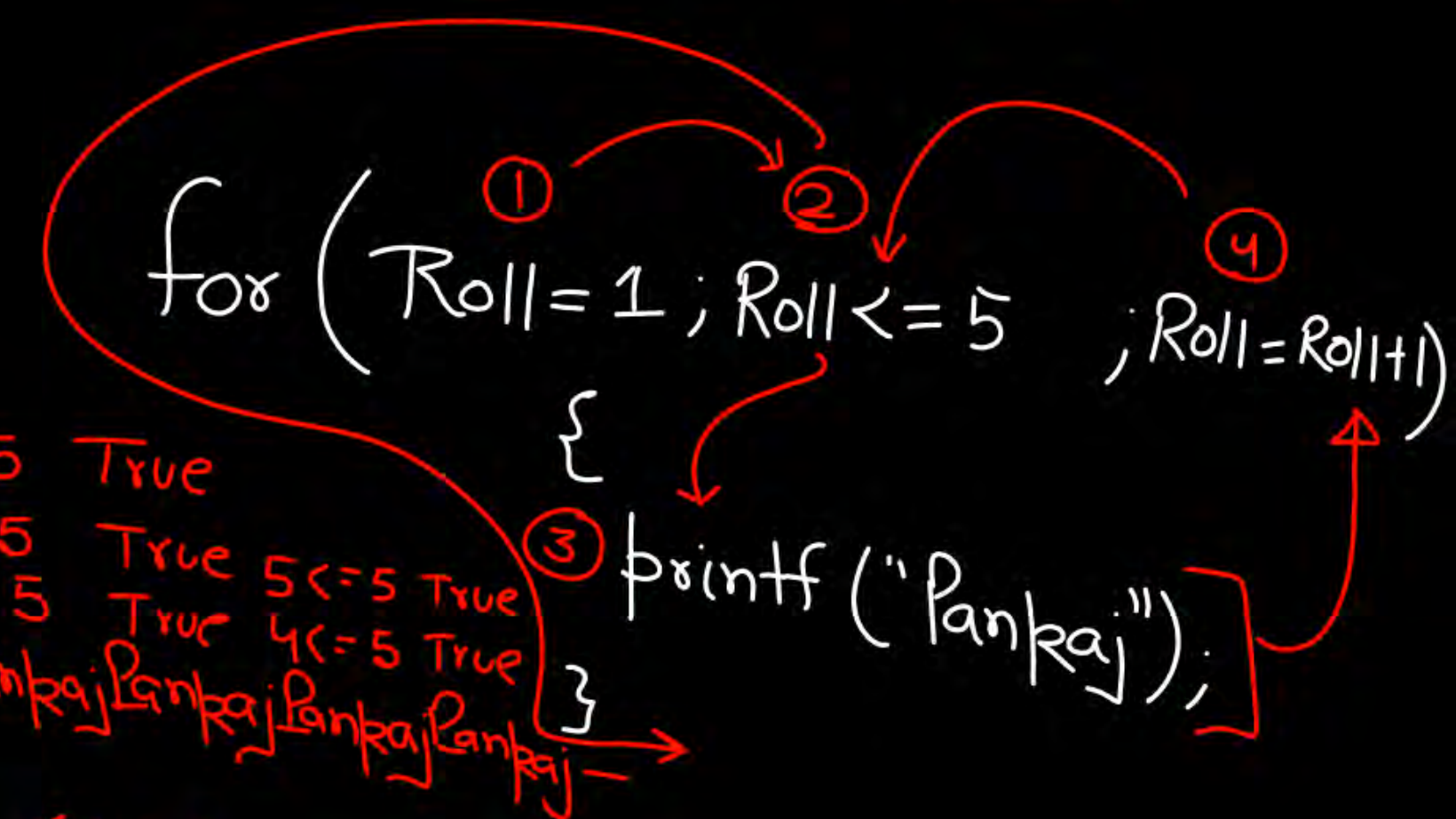Go down 3 times

print my name → 1000 times   loops

1st way

1st time

printf("Pankaj");
printf("Pankaj");

Ex: Roll = 1

Roll
5 3 4
6 1 2

1) Let us C
2) Dennis Ritchie
3) C in Depth

for (Roll = 1 ; Roll <= 5   ; Roll = Roll+1)
①   ②   ④
                        {
                  ③ printf("Pankaj");
                        }

1 <= 5   True
2 <= 5   True   5 <= 5 True
3 <= 5   True   4 <= 5 True
Pankaj Pankaj Pankaj Pankaj Pankaj →

6 <= 5 → False

for (initialization ; condition ; inc/dec)
① ② ④
{
③
code
}

True

Roll
1

Roll = 1, 2, 3, 4, 5, 6
✓ ✓ ✓ ✓ ✓ ✗

Books

1) Roll = 1

2)

False

for (Roll = 1 ; Roll <= 5 ; Roll ++)
① ② ④
{

③
printf ("Pankaj");

}

1)
for (Roll = 6 ; Roll <= 10 ; Roll ++)
① ② ④
{
③ printf ("Pankaj");
}

$11 \rightarrow$ $11 <= 10 \rightarrow$ False

Pankaj $\rightarrow$ 6, 7, 8, 9, 10

5 times

Roll		
6	6 <= 10 → True →	Pankaj ✓
7	7 <= 10 → True →	Pankaj
8	8 <= 10 → True →	Pankaj
9	9 <= 10 → True →	Pankaj
10	10 <= 10 → True →	Pankaj

for (Roll = 1; Roll < = 5; Roll ++)

$\{$

Code

$\}$

Roll = 1 $\Rightarrow$ Code will execute

Roll = 2 $\Rightarrow$ "

Roll = 3 $\Rightarrow$ "

Roll = 4 $\Rightarrow$ "

Roll = 5 $\Rightarrow$ "

$\boxed{4<=3}$ fals

for (Roll = -1 ; Roll <= 3; Roll ++)

$\{$

Code

$\}$

Roll = -1, 0, 1, 2, 3

1)

scope

```
      ①  ② ④
for ( i=1; i<=10; i++)
   {
   ③ printf ("Pankaj");
   }
```

i			
1	1<=10	True	✓
2	2<=10	True	✓
3	3<=10	True	✓
⋮			
10	10<=10	True	✓
11	11<=10	False	

1 to 10 ⟹ 10 times

2)

```
        ① → ②
for ( i=0; i>5; i++)
   {
   printf ("Pankaj");
   }
```

i		
0	0>5	false

0 times.

1 to 10

Last - First + 1

10 - 1 + 1 ⟹ 10

3)

```
for ( i=10; i<=100; i++)
   {
   printf ("Hello");
   }
```

100 - 10 + 1
= 90 + 1
⟹ 91 times

for (expression1 ; expression2 ; expression3)

① ② ④ ③

{

    code

}
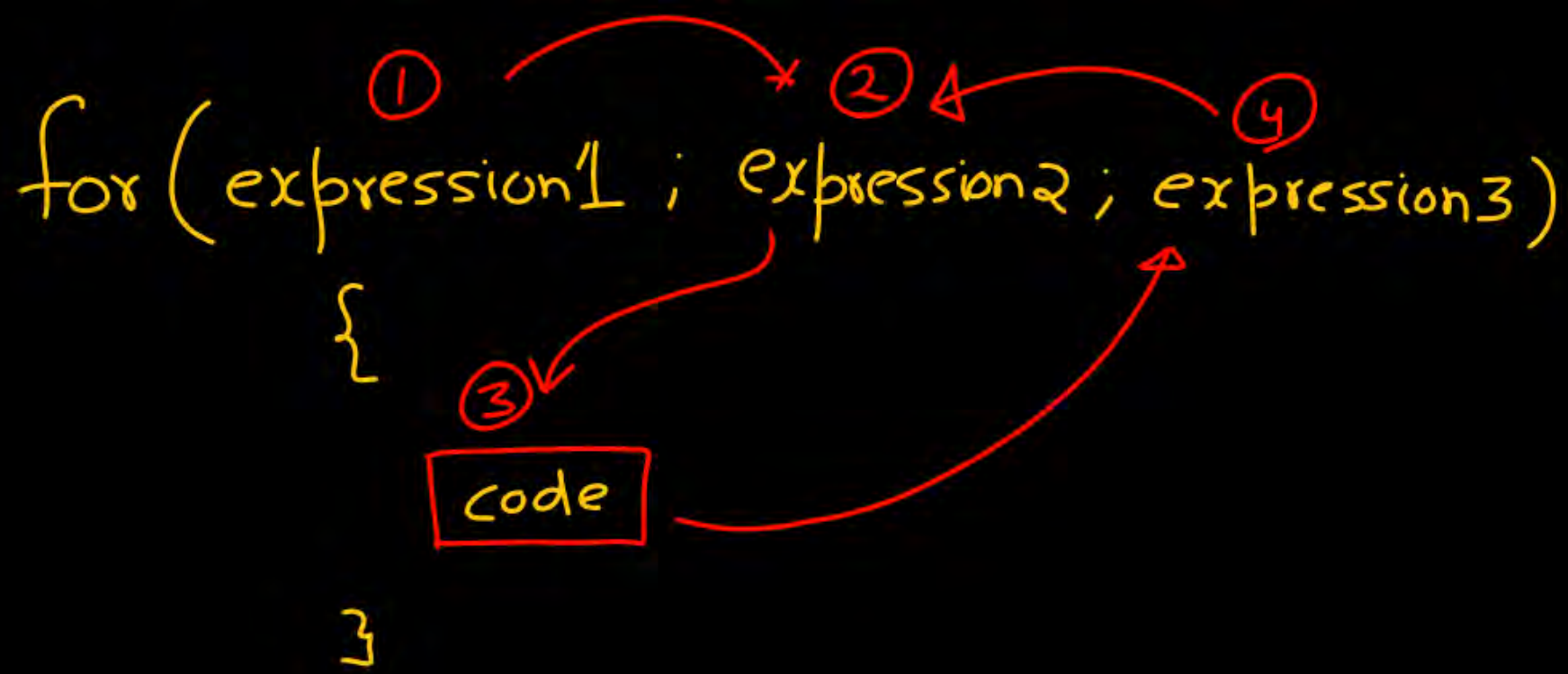
iteration

exp1 : ⟶ 1 time

exp2 ⟶ True ⟶ code ⟶ exp3

expression ⟶ statement with some value

```c
void main(){
    int i=1;
    for ( I0 ; i<=3 ; I2)
    {
        printf("Pankaj");
        i = i+2 ;
    }
}
```

① I0

② Is it a exp. ✓

scope

code

i

5 1̶ 3̶

1<=3   True
3<=3   True
5<=3   False

Pankaj Pankaj

void main(){

char ch = 1;

expression → value
ASCII code

false
0

for (ch = 1; ch; ch++)

printf("Pankaj"); → First semi-colon

printf("END");

}

by default signed

255 time
Pankaj
then END

1 → 127, 128, → -1
255

1, 2 ... 127

-128 → True (non-zero)
-127 → true

-2 0 1 2
-3 -1

-125
-127
-126
-128 127 126

```
void main() {
        char ch;

        for (ch = 1; ch; ch = ch + 2)

                printf("Pankaj");

        }
```

Pankaj will be printed
Infinite
    times

1, 3, 5, 7, ... 127,
-127, -125, ... -3, -1

int i=1;

①

for (printf("1") ; i<=5 ; printf("3"))

②   ④

{

③ | printf("2");
   | i++;

}

output : 1 2 3 2 3 2 3 2 3 2 3

i
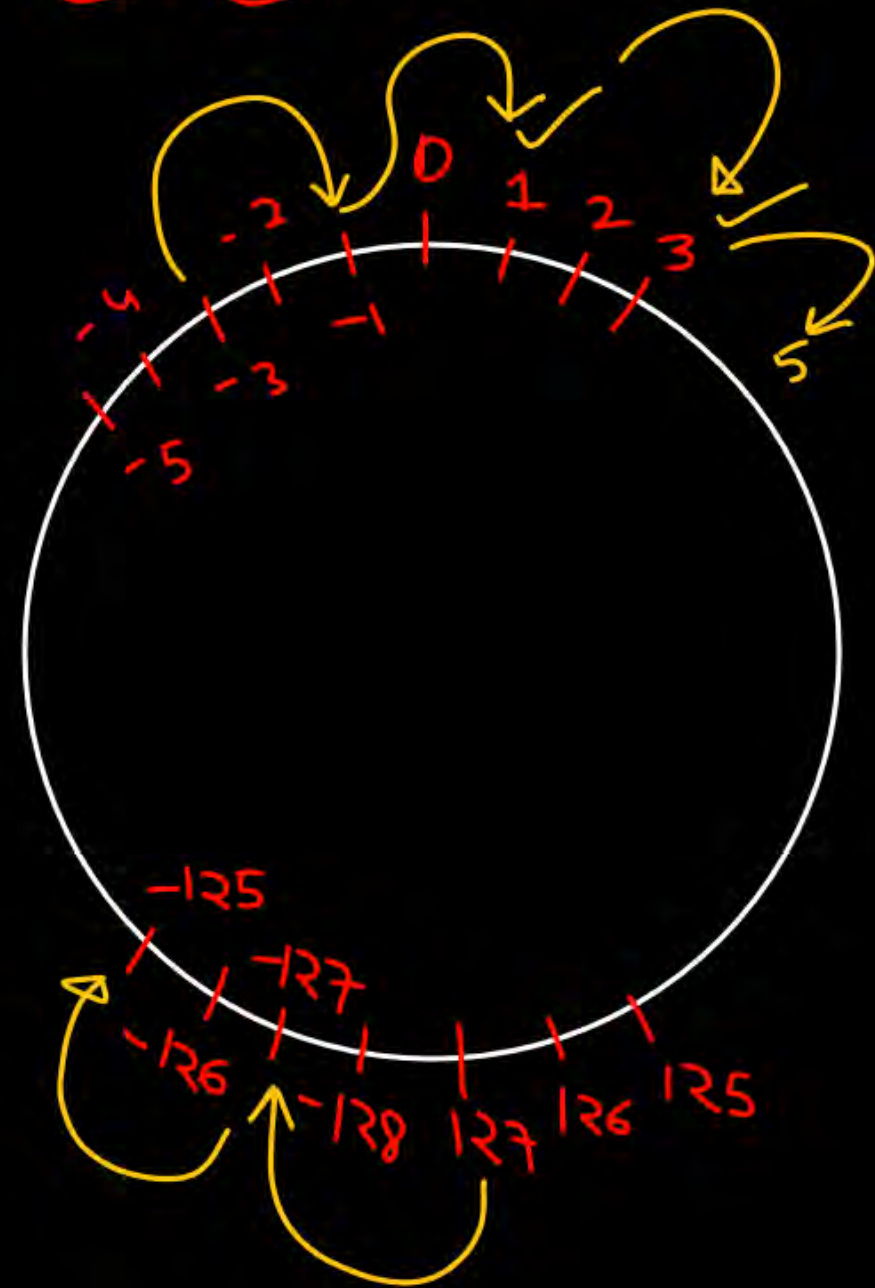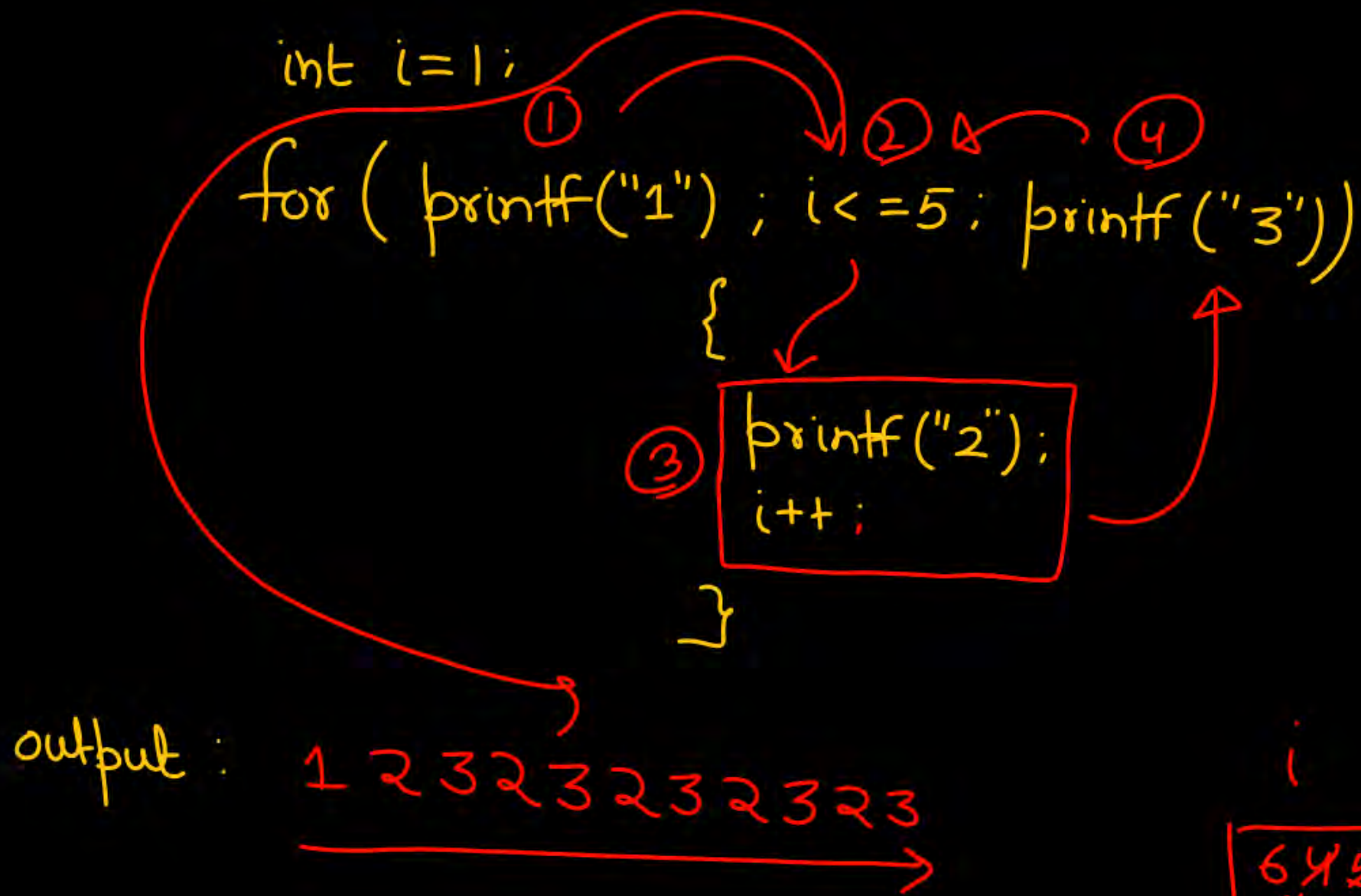
| 6 4 5 |
| 1 7 3 |

1   1<=5   True

2<=5   True

3<=5   True

4<=5   True

5<=5   True

6<=5   False
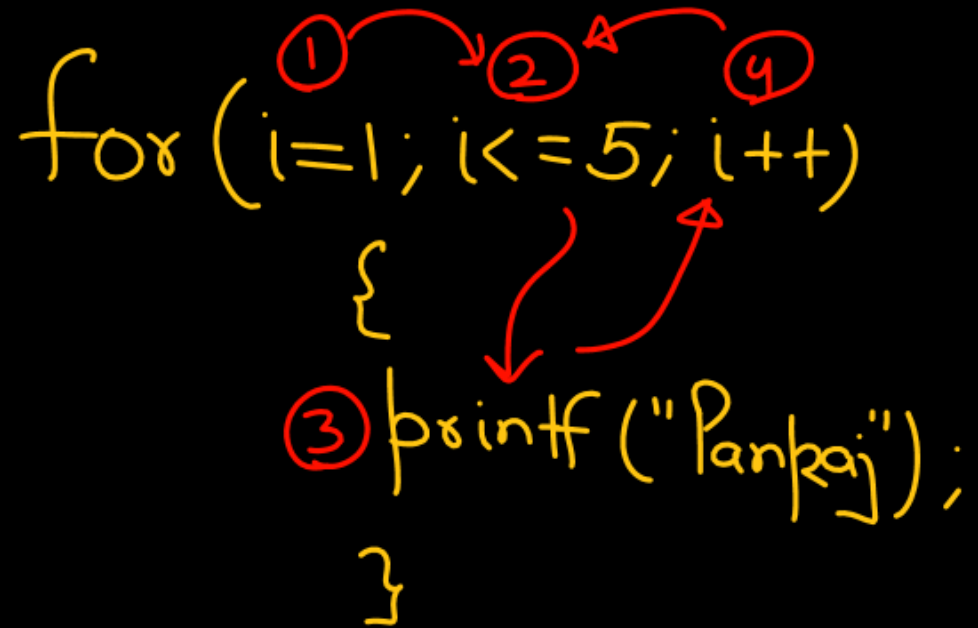
1.

for (①⟲② ④
10; 11; 12)

{ True

③ printf("Pankaj"); ✓

}

∞ times

2.

for (0; 1; 0)

printf("Pankaj");

∞ times

3.

for (①⟲ False
10; 0; 12)
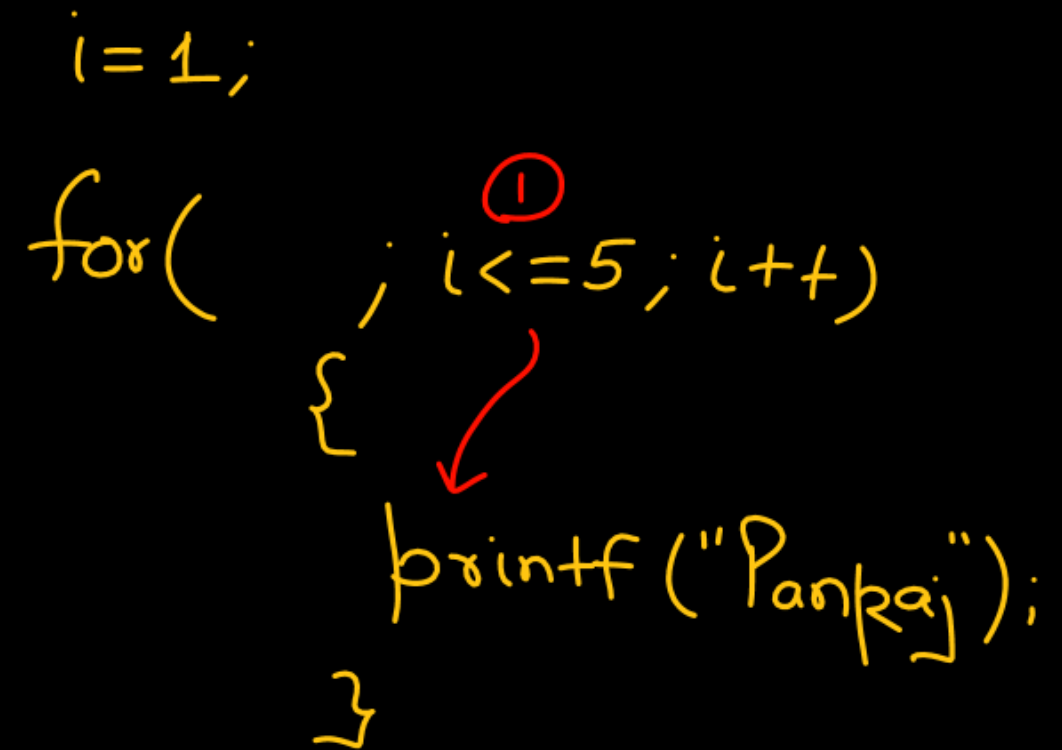
printf("Pankaj");

0 times

All 3 expressions are optional

1

for (i=1; i<=5; i++)
①   ②   ④
{
③ printf ("Pankaj");
}

$\Longrightarrow$

i=1;

for(; i<=5; i++)
          ①
{
printf ("Pankaj");
}

```
for (i=1; i<=5; i++)
    {
        printf("Pankaj");
    }
```

$\Longrightarrow$

```
i=1;
for (   ; i<=5;   )
    {
        printf("Pankaj");
        i++;
    }
```

mandatory

exp2 ⟹ omit

by default ⟶ (True)

for(;   ;)
{

  printf("Pankaj");

}

∞ times

for(i=1;   ; i++)
{

  printf("Pankaj");

}

∞ times

9:30 PM   Nandkishore

i
```
┌──────┐
│  10  │
└──────┘
  2096
```

11:00PM

i x
```
┌─────────────────────────────────┐
│  00000000 00001010                │
└─────────────────────────────────┘
```

```
void main(){

    int i=10;

    ___
    ___
    ___
    ___

    }
```

```
void main(){

    int (x);

    printf("/d",x)
```

→ Garbage

1

Q1.

```
int i = -1;

for ( i++ ; i++ ; i++ )
     {
         printf("Pankaj");
     }
```

0 times

i++ → (i) Use the value
     → (ii) Update

Q/ char $i = -1$: $\emptyset 1$

for ($\overset{-1}{i++}$; $\overset{\emptyset 1 2 3 4 5 6 7}{++i}$; $\overset{8(-1)}{i++}$) ①

{

   printf("Pankaj");

}

-1 $3, 5, 7, \dots -1$

        ∞ times

```
char i = -1;      3^{5-1} ①
            -1          0
for (i++; ++i; ++i)
    {
        printf("Pankaj");
    }

    1, 3, 5, ... - -1,
            ∞ times
```

```
int  i = 1;

for (   ;  ++i < 5 ;    )
    {
        printf("%d", i);
    }

(i)  i = i+1          i → 1 2 3 4 5
(ii)  i < 5
                            2<5 — True
                            3<5 — True
                            4<5 — True
            O/P: (234)      5<5 — False
```

$$i = 1$$

$$\text{for } (\quad ; \quad i++ < 5 \quad ; \quad)$$

(i) $i < 5$

(ii) $i = i + 1$

{
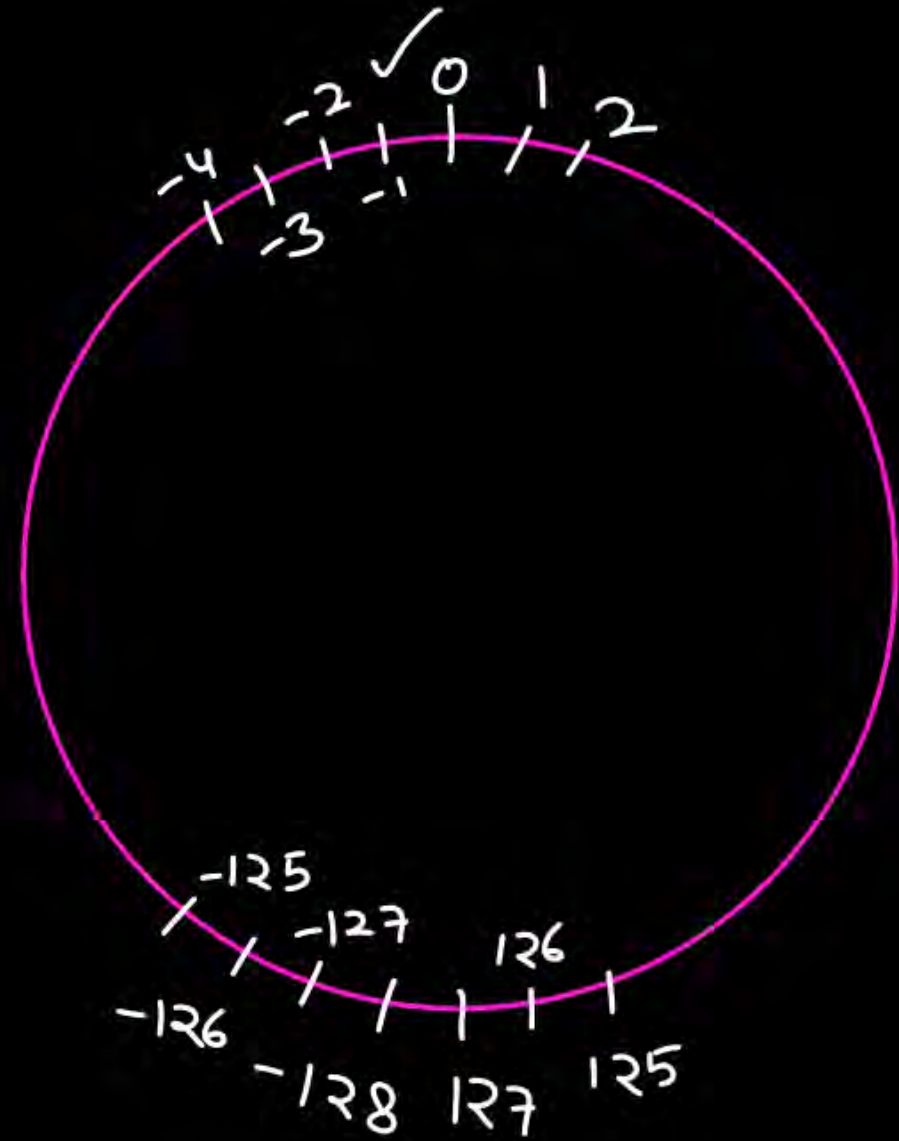
→ printf("%d", i);

}

O/P :   O/P : 2345

① $i = \cancel{1}\ \cancel{2}\ \cancel{3}\ \cancel{4}\ 5$

$1 < 5 \longrightarrow \text{True}$

$2 < 5 \longrightarrow \text{True}$

$3 < 5 \longrightarrow \text{True}$

$4 < 5 \longrightarrow \text{True}$

$5 < 5 \longrightarrow \text{False}$

1.

```
for (i=1; i<=10; i++)
    printf("Pankaj");
```

Ans : 10

2. $n \geq 1$

```
for (i=1; i<=n; i++)
    printf("Pankaj");
```

Ans: n times

3.
```
for (i=1; i<=10; i=i+2)
    printf("Pankaj");
```

$i = 1$ $1<=10$ ✓ $i = i+2$

$i = 3$ $3<=10$ ✓

$i = 5$ $5<=10$ ✓

$i = 7$ $7<=10$ ✓

$i = 9$ $9<=10$ ✓

$i = 11$ $11<=10$ ✗

5 times

4: 　　　for $(i=1; i<=n; i=i+2)$

　　　　　printf("Pankaj");

Exact Answer :　$\left\lceil \dfrac{n}{2} \right\rceil$

$n=10$　　$\left\lceil \dfrac{10}{2} \right\rceil = 5$

$n=11$　　$\left\lceil \dfrac{11}{2} \right\rceil = \lceil 5.5 \rceil = 6$

$n=10 \Rightarrow 10/2 = 5$

$n=11 \Rightarrow 11/2 \Rightarrow 5$

$\longrightarrow$ 6 times

$i = 1,3,5,7,9,11$

$n/2$ can not be right answer

n is

Even $\rightarrow \dfrac{n}{2}$

Odd $\rightarrow \dfrac{n}{2} + 1$

4: for $(i=1; i<=n; i=i*2)$
    printf ("Pankaj");

$n \Rightarrow 100$ | $n = 64$
$i \Rightarrow 64$ | $i = 64,$

$i = 1, 2, 2^2, 2^3, \ldots$ $\boxed{2^k}$ ← last value of i

$i = 2^0, 2^1, 2^2, 2^3, \ldots 2^k$

$\underbrace{\qquad\qquad\qquad}$
How many counting

printf $\rightarrow (k+1)$ times

$2^k <= n$

$\log 2^k <= \log n$

$k \log 2 <= \log n$

$k <= \log_2 n$ | $i <= 10$

$k = \lfloor \log_2 n \rfloor$

$\left(\lfloor \log_2 n \rfloor + 1 \right)$ times