

CS & IT ENGINEERING

Control Flow Statements

Iterative Statements: More about
loops-2

Programming in C

Lecture:06



By- Pankaj Sharma sir

TOPICS TO BE
COVERED

1: More about loops

145

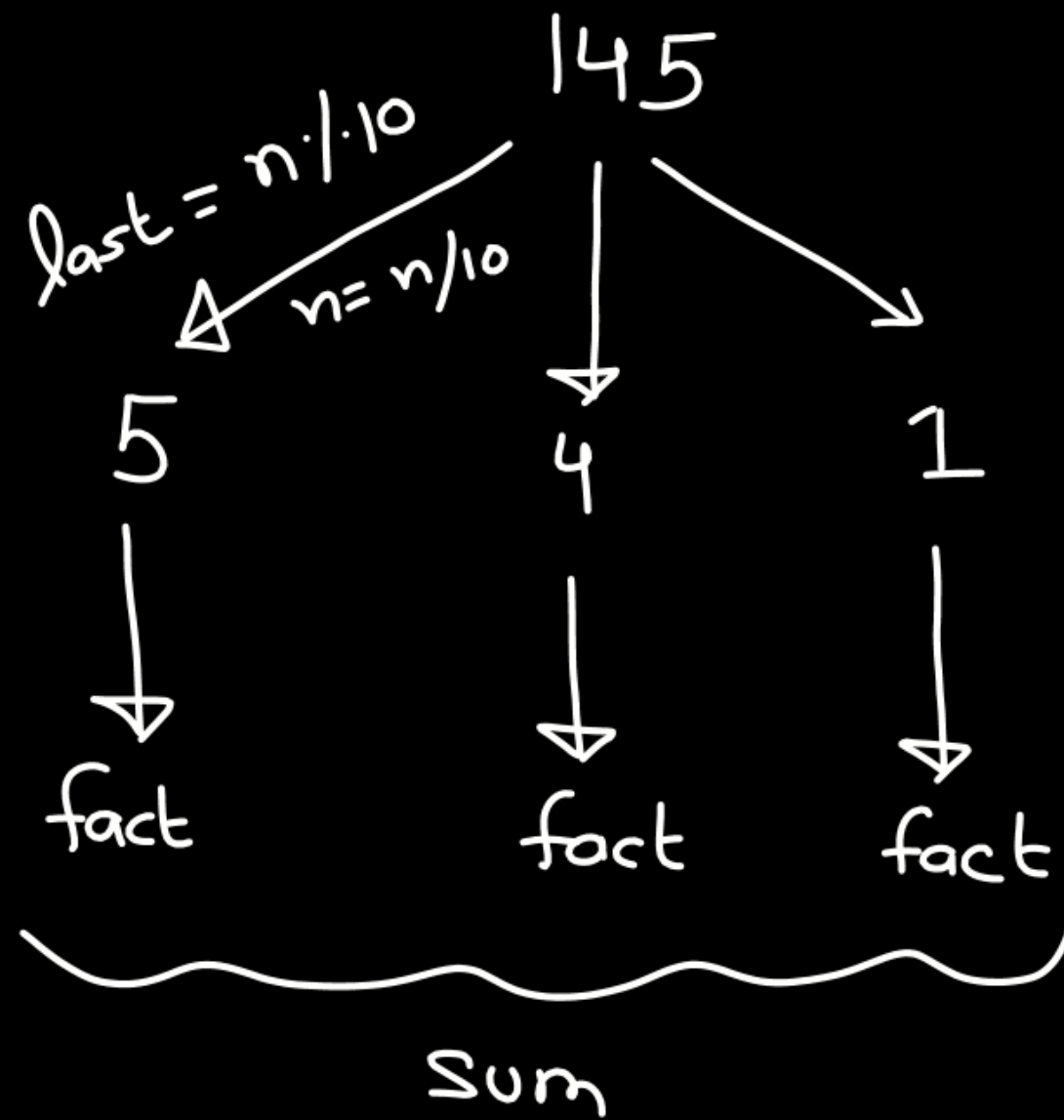
$$1! = 1$$

$$4! = 24$$

$$5! = 120$$

$$145$$

Sum of factorial of digit = n



$n = 145$

```
while (n > 0)
{
    last = n / 10;
    n = n / 10;
```

factorial of last

?

}

factorial of x

```

Prod = 1
for (i = 1; i <= x; i++)
    Prod = Prod * i;
  
```

last (with arrow pointing to ~~x~~)

```

n = 145
sum = 0
while (n > 0)
{
    last = n / 10;
    n = n / 10;
  }
  
```

```

Prod = 1;
for (i = 1; i <= last; i++)
    Prod = Prod * i;
  
```

←

```

sum = sum + Prod;
  
```

←

① $145 > 0 \Rightarrow \text{True}$
 $\text{last} = 5$
 $n = 14$
 $\text{Prod} = 120$

② $14 > 0$
 $\text{last} = 4$
 $n = 1$
 $\text{Prod} = 24$

③ $1 > 0$
 $\text{last} = 1$
 $n = 0$
 $\text{Prod} = 1$

factorial of x

```
Prod = 1
for (i = 1; i <=  $x$ ; i++)
    Prod = Prod * i;
```

last (with arrow pointing to ~~x~~)

```
n = 145
sum = 0 a = n;
while (n > 0)
{
    last = n / 10;
    n = n / 10;
```

```
if (sum == a)
    printf("%d is strong", a);
else
    printf("%d is not strong", a);
```

```
Prod = 1;
for (i = 1; i <= last; i++)
    Prod = Prod * i;
sum = sum + Prod;
}
```

```
Prod = 1;
for (i = 1; i <= 5; i++)
    Prod = Prod * i;
printf("%d", Prod);
```

```
Prod = 1;
for (i = 1; i <= 6; i++)
    Prod = Prod * i;
printf("%d", Prod);
```

```
Prod = 1;
for (i = 1; i <= 7; i++)
    Prod = Prod * i;
printf("%d", Prod);
```

5! 6! a b 7!

```
for (n = 5; n <= 7; n++)
{
    Prod = 1;
    for (i = 1; i <= n; i++)
        Prod = Prod * i;
    printf("%d", Prod);
}
```

fact of n

n = 5
n = 6
n = 7

Reverse

$$n = 145;$$

$$\text{Abtakpareverse} = 0;$$

$$\textcircled{1} \quad \begin{array}{l} \text{last} = n \% 10; \\ n = n / 10; \end{array} \quad \textcircled{5}$$

5

$$\text{Abtakpareverse} = \text{Abtakpareverse} \times 10 + \text{last};$$

$$\textcircled{2} \quad \begin{array}{l} \text{last} = 4 \\ n = 1 \end{array} \quad \text{Abtakpareverse} = 54$$

decimal

~
x

$$\Rightarrow 10 \times \text{old} + \text{sym}$$

$$e^x = \underbrace{1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots}_{\text{no. of terms} \Rightarrow 4}$$

$x = 2$

float \Rightarrow sum = 0.0;

Sin x

cos x

Try

for (i=0; i<=3; i++)

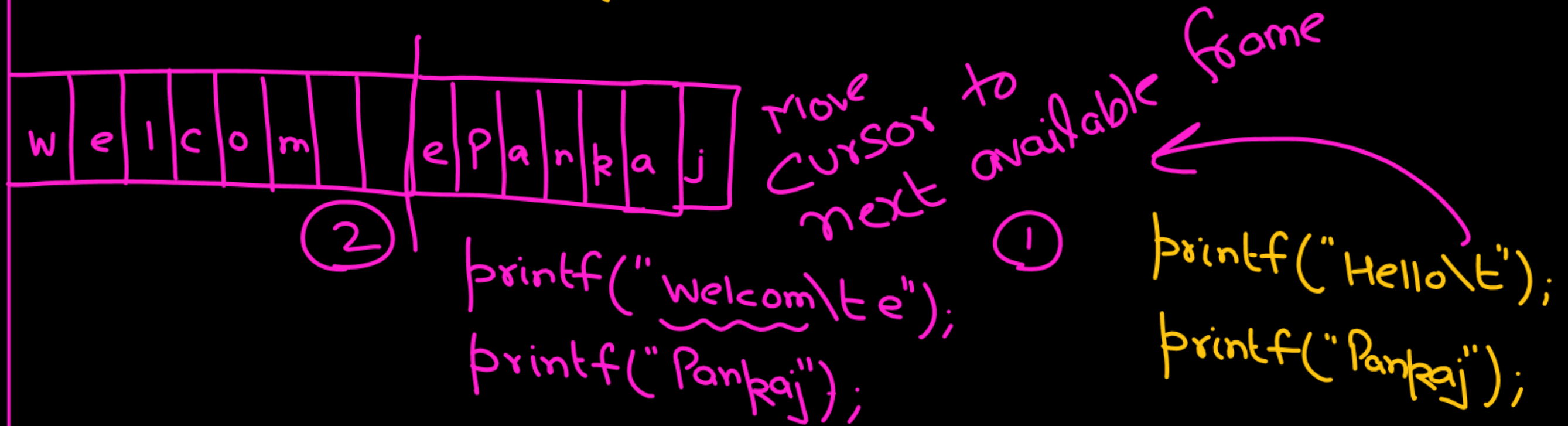
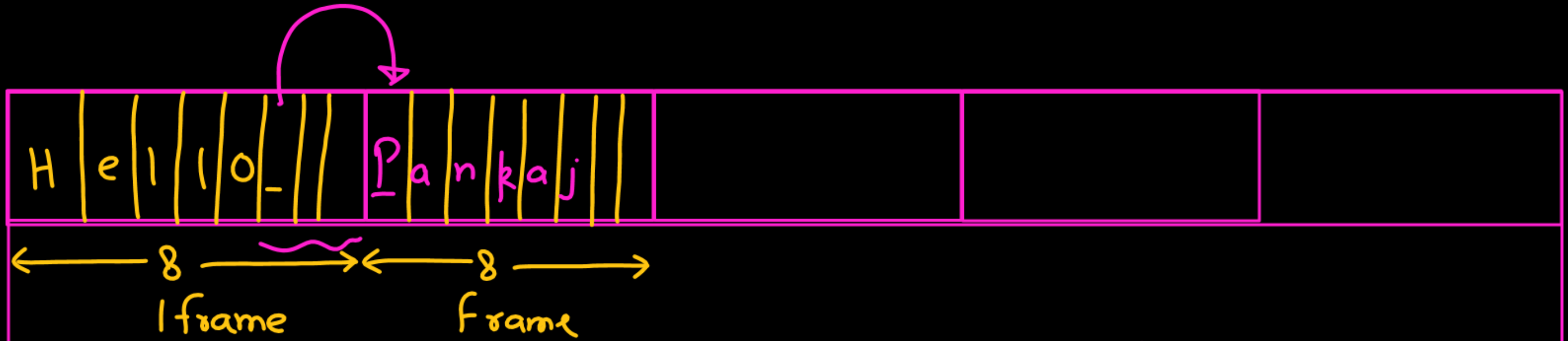
{

$y = x^i$

$z = \text{fact}(i)$

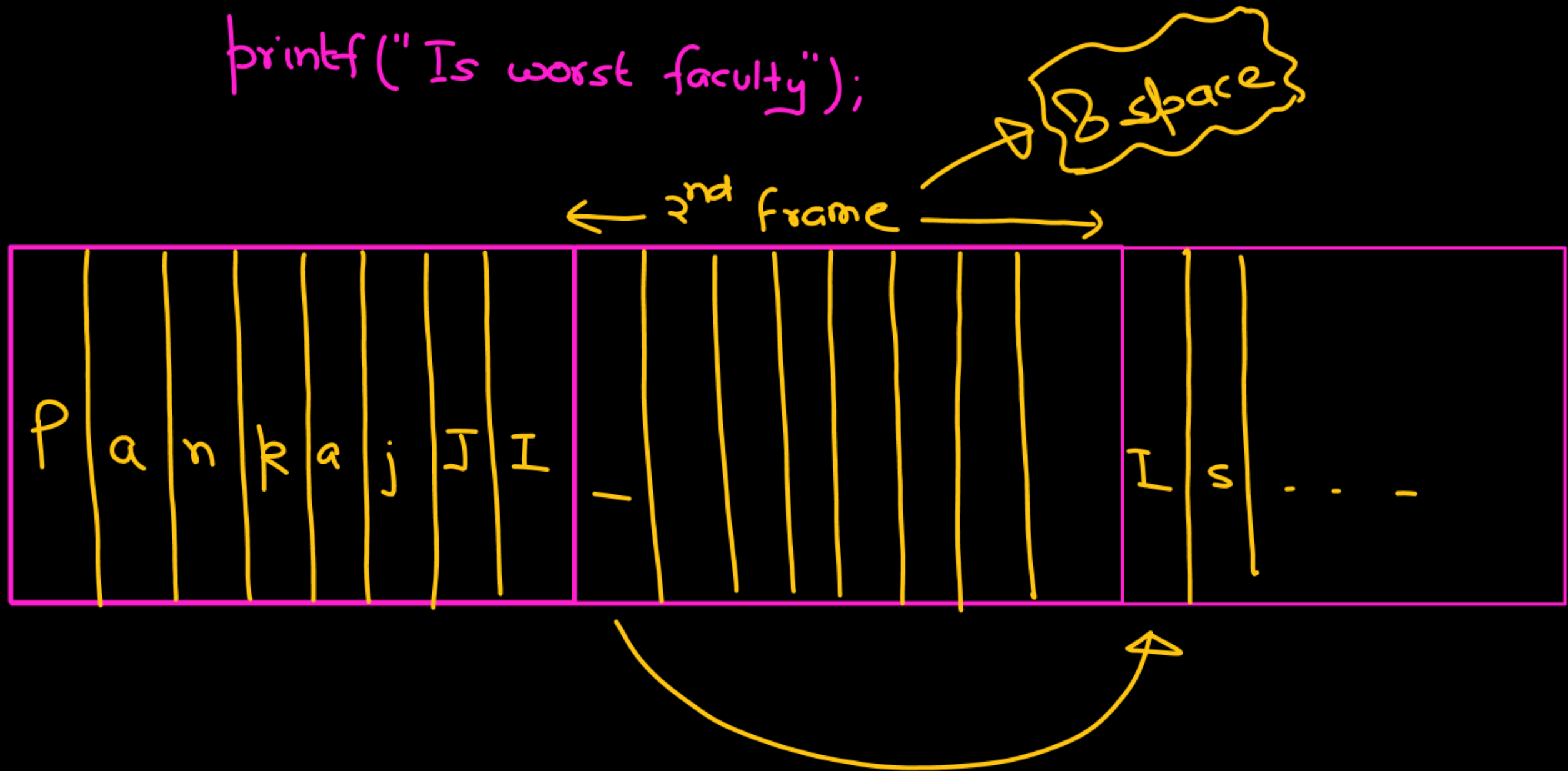
sum = sum + $\frac{y}{z}$;

}



```
printf("Pankaj Ji \t");
```

```
printf("Is worst faculty");
```



Panpakj JI \t

P	a	n	k	a	j		J	I	-										
---	---	---	---	---	---	--	---	---	---	--	--	--	--	--	--	--	--	--	--



1.
using loop

12345

```
for(i=1; i<=5; i++)  
{  
    printf("%d", i);  
}
```

```
for(x=1; x<=4; x++)  
{  
  
}
```

code ✓

12345 ✓
12345 ✓
12345 ✓
12345 ✓

```
for (i = 1; i <= 5; i++)  
{  
    printing —  
}
```

Row
⇒

1	1				
2	1	2			
3	1	2	3		
4	1	2	3	4	
5	1	2	3	4	5

```
for (Row = 1; Row <= 5; Row++)  
{
```

```
    for (col = 1; col <= 5; col++)  
    {
```

```
    }
```

Row \Rightarrow

	col
	1 2 3 4 5
1	1
2	1 2
3	1 2 3
4	1 2 3 4
5	1 2 3 4 5

Row col

1 1

2 1, 2

3 1, 2, 3

4 1, 2, 3, 4

5 1, 2, 3, 4, 5

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

{

for (col = 1; col <= row; col++)

{

printf("%d", col);

}

}

col
1 2 3 4 5

1				
1	2			
1	2	3		
1	2	3	4	
1	2	3	4	5

Row col

1 1

2 1, 2

3 1, 2, 3

4 1, 2, 3, 4

5 1, 2, 3, 4, 5

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

{

for (col = 1; col <= row; col++)

{

printf("%d", col);

}

printf("\n");

}

col
1 2 3 4 5

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

Row	col
1	1
2	1,

1 1

1
2

	1	2	3	4	5
1	1				
2	2	2			
3	3	3	3		
4	4	4	4	4	
5	5	5	5	5	5

Row	col
1	1
2	1, 2
3	1, 2, 3
4	1, 2, 3, 4
5	1, 2, 3, 4, 5

```
for (Row = 1; Row <= 5; Row++)
```

```
{
```

```
for (col = 1; col <= Row; col++)
```

```
{
```

```
printf("%d", Row);
```

```
}
printf("\n");
```

```
}
```

Row	col.	last
1	5	6
2	4	6
3	3	6
4	2	6
5	1	6

Row	col
1	1,2,3,4,5
2	1,2,3,4
3	1,2,3
4	1,2
5	1

	1	2	3	4	5
1	1	2	3	4	5
2	1	2	3	4	
3	1	2	3		
4	1	2			
5	1				

for(Row=1; Row<=5; Row++)
 {
 for(col=1; col<=6-Row; col++)
 {
 }
 }
 }
 3 min % \Rightarrow code

```

for(Row=1; Row<=5; Row++)
{
    for(col=1; col<=6-Row; col++)

```

$5 \Rightarrow$

$6 \Rightarrow 7$

$n \Rightarrow (n+1) - \text{Row}$

```

        printf("%d", col);

```

```

    }

```

```

    printf("\n");

```

```

}

```

col

1 2 3 4 5

1 1 2 3 4 5

2 1 2 3 4

3 1 2 3

4 1 2

5 1

3 min % \Rightarrow code

Row 1 space
 ----- *
 Row 2 ----- **
 Row 3 ----- ***
 Row 4 ----- ****
 Row 5 ----- *****

Row col
 1 1 2 3 4 5
 2 * *
 3 ** *
 4 ***
 5 ****

Every Row → Some space
 to be
 printed
 → Some *
 to be printed

```
for (Row = 1; Row <= 5; Row++)  
{  
    1. Code to print spaces
```

```
    2. Code to print star
```

```
    3. Newline
```

```
}
```

Every Row → Some space
to be printed

→ Some *
to be printed

	col
	1 2 3 4 5
Row 1	*
Row 2	**
Row 3	***
Row 4	****
Row 5	*****

row + space = 5

$Row + space = 5$
 $Row \quad space \Rightarrow \underline{5 - Row} \quad star \Rightarrow Row$

Row

space

Star \Rightarrow Row

1

4

1

2

3

2

3

2

3

4

1

4

5

0

5

Row

1

2

3

U

—

col
1 2 3 4 5

* *

大大大

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

{

for (space = 1; space <= 5 - Row; space++)

{

printf(" ");

}

for (star = 1; star <= Row; star++)

{

printf("*");

printf("\n");

}

5

n

Maths

```
for(Row=1; Row<=n4; Row++)
{
```

```
    for(space=1; space<=n4-Row;
        space++)
        printf(" ");
```

```
    for(star=1; star<=2nRow-1;
        star++)
        printf("*");
```

```
    printf("\n");
```

```
}
```

Row+space=4

Row space

1 3

2 2

3 1

4 0

Row star

1 1

2 3

3 5

4 7

	1	2	3	4	5	6	7
1				*			
2			*	*	*		
3		*	*	*	*	*	
4	*	*	*	*	*	*	*

= $2 \times \text{Row} - 1$

$n=5$

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

$n=6$

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

```
for(Row=1; Row<=n; Row++)
{
    for(col=1; col<=i; col++)
    {
        //
    }
}
```

