

18/1/20

wanted for loop

expected o/p = * * * * *

```

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

```

```

2 for (int i=1; i<=5; i++){
    for (int j=1; j<=5; j++){
        printf("* ");
    }
    printf("\n");
}

```

exp o/p →

```

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

```

2

DRY RUN

i	j	i<=4	j<=4	o/p	j++	i++
1	1	1<=4	1<=4	*	2	
1	2	1<=4	2<=4	*	3	
1	3	1<=4	3<=4	*	4	
1	4	1<=4	4<=4	*	5	2
2	1	2<=4	1<=4	*	2	
2	2	2<=4	2<=4	*	3	
2	3	2<=4	3<=4	*	4	
2	4	2<=4	4<=4	*	5	3
3	1	3<=4	1<=4	*	2	
3	2	3<=4	2<=4	*	3	
3	3	3<=4	3<=4	*	4	
3	4	3<=4	4<=4	*	5	4
4	1	4<=4	1<=4	*	2	
4	2	4<=4	2<=4	*	3	
4	3	4<=4	3<=4	*	4	
4	4	4<=4	4<=4	*	5	5x

③ Expected output =

```

1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
    
```

```

code:-
void main() {
    for(int i=1; i<=5; i++) {
        for(int j=1; j<=5; j++) {
            printf("1 ");
        }
        printf("\n");
    }
}
    
```

DRY RUN

i	j	i<=5	j<=5	O/P	j++	i++
1	1	1<=5	1<=5	1	2	
	2	1<=5	2<=5	1	3	
	3	1<=5	3<=5	1	4	
	4	1<=5	4<=5	1	5	
	5	1<=5	5<=5	1	6	
	6	1<=5	6<=5 X			2
2	1	2<=5	1<=5	1	2	
	2		2<=5	1	3	
	3		3<=5	1	4	
	4		4<=5	1	5	
	5		5<=5	1	6	
	6		6<=5 X			3
3	1	3<=5	1<=5	1	2	
	2		2<=5	1	3	
	3		3<=5	1	4	
	4		4<=5	1	5	
	5		5<=5	1	6	
	6		6<=5 X			4

i	j	$i < j$	$i <= j$	o/p	j++	i++
4	1	$4 < 5$	$4 <= 5$	1-	2	
	2		$2 <= 6$	1-	3	
	3		$3 <= 5$	1-	4	
	4		$4 <= 5$	1-	5	
	5		$5 <= 5$	1-	6	
	6		$6 <= 5 \times$			5
5	1	$5 <= 5$	$5 <= 5$	1-	2	
	2		$2 <= 5$	1-	3	
	3		$3 <= 5$	1-	4	
	4		$4 <= 5$	1-	5	
	5		$5 <= 5$	1-	6	
	6		$6 <= 5 \times$			6
6		$6 <= 5 \times$				

④ Expected output =

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

end of

```
void main() {
```

```
for (int i=1; i<=6; i++)
```

```
for (int j=1; j<=5; j++)
```

```
print("A")
```

$$3 \quad p \text{ statt } f(n)$$

DRY RUN

i	j	i <= 5	j <= 5	o/p	j++	i++
1	1	1 <= 5	1 <= 5	A -	2	
	2		2 <= 5	A -	3	
	3		3 <= 5	A -	4	
	4		4 <= 5	A -	5	
	5		5 <= 5	A -	6	
	6		6 <= 5 X			2
2	1	2 <= 5	1 <= 5	A -	2	
	2		2 <= 5	A -	3	
	3		3 <= 5	A -	4	
	4		4 <= 5	A -	5	
	5		5 <= 5	A -	6	
	6		6 <= 5 X			3
3	1	3 <= 5	1 <= 5	A -	2	
	2		2 <= 5	A -	3	
	3		3 <= 5	A -	4	
	4		4 <= 5	A -	5	
	5		5 <= 5	A -	6	
	6		6 <= 5 X			4
4	1	4 <= 5	1 <= 5	A -	2	
	2		2 <= 5	A -	3	
	3		3 <= 5	A -	4	
	4		4 <= 5	A -	5	
	5		5 <= 5	A -	6	
	6		6 <= 5 X			5
5	1	5 <= 5	1 <= 5	A -	2	
	2		2 <= 5	A -	3	
	3		3 <= 5	A -	4	
	4		4 <= 5	A -	5	
	5		5 <= 5	A -	6	
	6		6 <= 5 X			6
6	6	6 <= 5 X				

18/7/20

Q) code:

```
void main() {
    for(int i=1; i<6; i++) {
        for(int j=i; j<6; j++) {
            printf("%d", j);
        }
        printf("\n");
    }
}
```

DRY RUN

i	j	i<6	j<6	o/p	j++	i++
1	1	1<6	1<6	1	2	
	2		2<6	2	3	
	3		3<6	3	4	
	4		4<6	4	5	
	5		5<6	5	6	
	6		6<6 X			
2	1	2<6	1<6	1	2	2
	2		2<6	2	3	
	3		3<6	3	4	
	4		4<6	4	5	
	5		5<6	5	6	
	6		6<6 X			
3	1	3<6	1<6	1	2	3
	2		2<6	2	3	
	3		3<6	3	4	
	4		4<6	4	5	
	5		5<6	5	6	
	6		6<6 X			
4	1	4<6	1<6	1	2	4
	2		2<6	2	3	
	3		3<6	3	4	
	4		4<6	4	5	
	5		5<6	5	6	
	6		6<6 X			

18/11/20

i	j	j < 6	j < 6	o/p	j++	i++
4	1	4 < 6	1 < 6	1	2	
	2		2 < 6	2	3	
	3		3 < 6	3	4	
	4		4 < 6	4	5	
	5		5 < 6	5	6	
	6		6 < 6 X			
5	1	5 < 6	1 < 6	1	2	
	2		2 < 6	2	3	
	3		3 < 6	3	4	
	4		4 < 6	4	5	
	5		5 < 6	5	6	
	6		6 < 6 X			
6	1	6 < 6 X				
	2					
	3					
	4					
	5					
	6					