Pattern Printing Questions

```
* * * *
* * * *
* * * *
```

Ques : Print the given pattern

5
✱ ✱ ✱ ✱ ✱
3 ✱ ✱ ✱ ✱ ✱
✱ ✱ ✱ ✱ ✱

No. of lines → $\boxed{n}$

Stars in each line → m

$\boxed{\text{Nested}}$ Loops

Solid Rectangle

```
for(int i=1;i<=3;i++){
    for(int i=1;i<=n;i++)
        printf("*");
    printf("\n");
}
```

n = 5

3

Output

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

5

* * * * *
3 * * * * *
* * * * *

no. of lines = no. of rows

no. of stars in each line = no. of columns

HW : Print the given pattern & dry run
↓
n = 2

```
****
****
****
****
```

int n;

print

Solid Square

Ques : Print the given pattern

n = 4
↓

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

n = 3
↓

```
1 2 3
1 2 3
1 2 3
1 2 3
```

n = 5

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

Number Square

Ques : Print the given pattern

n = 4

```
*
**
***
****
```

n = 2

```
*
* *
```

n = 3

```
*
* *
* * *
```

```
1    *
2    * *
3    * * *
4    * * * *
```

4

Star Triangle

```c
for(int i=1;i<=3;i++){ // no of lines / rows -> i
    for(int j=1;j<=i;j++){ // no of columns -> j
        printf("* ");
    }
    printf("\n");
}
```

i = 1̶ 2̶ 3̶ 4

j = 1̶ 2̶ 1̶ 2̶ 3̶ 1 2 3 4

n = 3

Output

- *
- * *
- * * *
- .

Ques : Print the given pattern

$n = 4$

```
****
***
**
*
```

$n = 4$

i

1	*	*	*	*	4
2	*	*	*		3
3	*	*			2
4	*				1

for(int i=1 ; i ≤ n; i++)
{
   for(int j =

→ \n

3

$i + j = n + 1$

$j = n + 1 - i$

Star Triangle Ulta

FOR ANY PATTERNS:

1) No. of lines

2) In each line what is happening

Ques : Print the given pattern

n = 4

1
1 2
1 2 3
1 2 3 4

n = 2

1
1 2

X
X X
X X X
X X X X

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

Number Triangle

HW : Print the given pattern

Hint

1 2 3 4   k k k k
1 2 3   k k k
1 2   k k
1   k

Number Triangle Ulta

Ques : Print the given pattern

n = 4

1
1 3
1 3 5
1 3 5 7

1 3 5 7   →   1 3 5 7   →   1 3 5 7
1 3 5 7
1 3 5 7
1 3 5 7

1 3 5 7 9 . . . . 'n'

$$a_n = 1 + (n-1) \cdot 2 \Rightarrow \boxed{2n-1}$$

Odd Number Triangle

```c
for(int i=1;i<=n;i++){
    for(int j=1;j<=i;j=j+2){
        printf("%d ",j);
    }
    printf("\n");
}
```

4

i = 1 2 3

j = 1 3 1 3 5

1
1
1 3
1 3

1
1
1 3
1

```
for(int i=1;i<=n;i++){
    int a = 1;
    for(int j=1;j<=i;j++){
        printf("%d ",a);
        a = a + 2;
    }
    printf("\n");
}
```

$a = \cancel{1}\ 3$

$i = \cancel{1}\ 2$

$j = \cancel{1}\ \cancel{2}\ 1$

n = 3

1
1 3
1 3 5

Output

. 1

. 3 5

. 7 9 11

* Ques : Print the given pattern

n = 4

A B C D
A B C D
A B C D
A B C D

n = 3

A B C
A B C
A B C

n = 5

A B C D E
A B C D E
A B C D E
A B C D E
A B C D E

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

Alphabet Square

PW SKILLS

1   2   3   A   B   C

1   2   3   →   A   B   C

1   2   3   A   B   C

    ↓

  6 5   6 6   6 7

  6 5   6 6   6 7

  6 5   6 6   6 7

$A \rightarrow 65$ $a = 97$

$B \rightarrow 66$

$C \rightarrow 67$

HW : Print the given pattern

```
A
A B
A B C
A B C D
```

→

```
A  B  C  D
A  B  C  D
A  B  C  D
A  B  C  D
```

& Dry Run

Alphabet Triangle

```
1              1              A
2  A B         1 2            A B
3  1 2 3       1 2 3          A B C
4  A B C D     1 2 3 4        A B C D
5  1 2 3 4 5   1 2 3 4 5      A B C D E
```

→

if - else

& Number
Alphabet Triangle

Ques : Print the given pattern

n 'odd'

```
  1 2 3 4 5
1 # # * # #
2 # # * # #
3 * * * * *
4 # # * # #
5 # # * # #
```

```
    1 2 3  n = 3
  1 # * #
  2 * * *
  3 # * #
```

↓

2

n = 1

*

$$\frac{n}{2} + 1 \rightarrow \frac{5}{2} + 1 = 2 + 1 = 3$$

Star Plus

```
j
     1 2 3 4
1    x x k x
2    k a k k
3    k k a k
4    k k k a
i
```

```
for (int i=1; i≤4 ; i++)          rows → i
{
    for (int j=1; j≤4; j++)       columns → j
    {
        printf("*");
    }
    printf("\n");
}
```

HW : Print the given pattern

$n \rightarrow$ rows $m \rightarrow$ colums

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

Ques : Print the given pattern

n is odd

n = 5

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

if ($i == j$ || $i + j == n+1$)   print ("*");

else   print (" ");

Star Cross

*Ques : Print the given pattern

n = 4

1
2 3
4 5 6
7 8 9 10

n = 3

1
2 3
4 5 6

```
for (i=1; i≤n; i++)
{
    for (j=1; j≤i; j++)
    {
        printf ( )
    }
}
```

 # Homework :

1
3 5
7 9 11
13 15 17 19

Floyd's Triangle

*Ques : Print the given pattern

n=4

1 2 3 4

1 1
2 0 1
3 1 0 1
4 0 1 0 1

if (i % 2 != 0) → start with 1

else → start with 0

0 & 1 Triangle

$n = 4$

```c
int a;
for(int i=1;i<=n;i++){
    if(i%2!=0) a = 1;
    else a = 0;
    for(int j=1;j<=i;j++){
        printf("%d ",a);
        if(a==0) a = 1;
        else a = 0;
    }
    printf("\n");
}
```

→ start

→ switch

$a = 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0$

Output
- 1
- 0 1
- 1 0 1
- 0 1 0 1

$n = 5$

	1	2	3	4	5
1	1 ✓				
2	0	1 ✓			
3	1	0	1 ✓		
4	0	1	0	1 ✓	
5	1	0	1	0	1 ✓

```
if ( i == j               ) printf ("1");

if ( i+j % 2 == 0)  printf ("1");
else   printf ("0");
```

*Ques : Print the given pattern

```
    1  2  3  4
1   #  #  # *
2   #  # * *
3   # * * *
4  * * * *
```

(=)

```
# # #
# #
#
```

\+

```
  *
 * *
* * *
* * * *
```

```
for ( int i =        ){
    for (        ){
    }
    for (        ){
    }
}
```

Star Triangle Mast

```
1    # # # *
i  2    # # * *
   3    # * * *
   4    * * * *
```

```
           j
i = 1  | # → 3    → 4  | K
i = 2  | # → 2    → 4  |
i = 3  | # → 1    → 4  |
i = 4  | # → 4    → 4  |
```

$$i + j = n$$
$$\boxed{j = n - i}$$

```
for(int i=1 ; i<=4 ; i++){

    for(int j=1 ; j<=n-i ; j++){
        printf("#");
    }
    for(int k=1 ; k<=i ; k++){
        printf("*");
    }
    printf("\n");
}
```

HW : Print the given pattern

```
_ _ _ ****
_ _ ****
_ ****
****
```

→

+

```
_ _ _ &
_ _ & &
_ & & &
& & & &
```

→

$n = 4$

```
& & & &
_ & & &
_ _ & &
_ _ _ &
```

$nsp = 0;$
$nst = n;$

$nsp++;$
$nst--;$

Rhombus

HW : Print the given pattern

```
        A
      A B
    A B C
  A B C D
```

→

```
A
A B
A B C
A B C D
```

Alphabet Triangle

*Ques : Print the given pattern

i   K $n = 4$

```
i
1    _ _ _*            1    k
2    _ _ ***           2    k k k
3    _ *****           3    k k k k k
4    *******           4    k k k k k k k
                       ·
                       ·
                       ·
                       n
```

k
k k
k k k
k k k k

1 , 3, 5, 7

int nst = 1;

↪ nst = nst + 2;

→ $a_n = 1 + (n-1) \cdot 2 \Rightarrow 2^n - 1$ K = 1; K <= n-i

→ $\boxed{a_i = 2*i - 1}$

Star Pyramid

```
int nst = 1;
for(int i=1;i<=3;i++){
    for(int j=1;j<=nst;j++){
        printf("*");
    }
    nst = nst + 2;
    printf("\n");
}
```

Output

*

* * *

* * * * *

nst = 1 3 5

i = 1 2

j = 1 1

$n=4$

1   _ _ _ *

2   _ _ * * *

3   _ * * * * *

4   * * * * * * *

```
int nsp = 3;
int nst = 1;
```

HW : Print the given pattern

```
    1
  1 2 3
 1 2 3 4 5
1 2 3 4 5 6 7
```

Clarity

```
1
1 2 3
1 2 3 4 5
1 2 3 4 5 6 7
```

Number Pyramid

*HW : Print the given pattern

```
    A
  A B C
 A B C D E
A B C D E F G
```
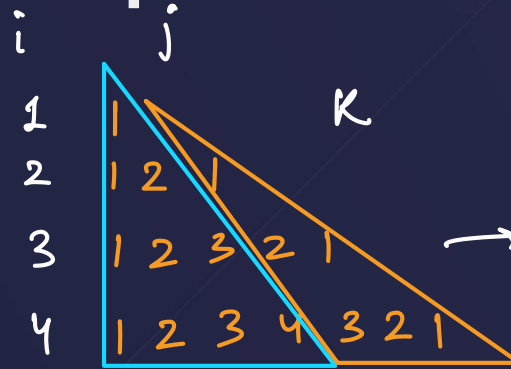
Alphabet Pyramid

Ques : Print the given pattern

```
i
1    _ _ _ 1
2    _ _ 1 2 1
3    _ 1 2 3 2 1
4    1 2 3 4 3 2 1
```

→

```
   i    j              K
   1    1
   2    1 2
   3    1 2 3 2 1
   4    1 2 3 4 3 2 1
```

→

```
      1
      2 1
      3 2 1
```

x **Homework :**

```
_ _ _ A
_ _ A B A
_ A B C B A
A B C D C B A
```

Number Pyramid Mast

SKILLS

Ques : Print the given pattern   n is odd

i   n=7

Power of [nst] & [nsp]

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

nsp --
nst += 2

_ x
x x x
  x
n = 3

_ _ x
_ x x x
x x x x x
  x x x
    x
n = 5

$\left(\dfrac{n}{2}+1\right)^{th}$ line → middle line

middle line → nsp = 0
                nst = n

Star Diamond

```
for(int i=1;i<=n;i++){
    for(int j=1;j<=nsp;j++){ // spaces
        printf(" ");
    }
    for(int k=1;k<=nst;k++){ // stars
        printf("*");
    }
    if(i<ml){
        nsp--;
        nst+=2;
    }
    else{
        nsp++;
        nst-=2;
    }
    printf("\n");
}
```

$n \to$ no. of lines $= 5$, $ml = 3$

$nsp = \dfrac{n}{2} = \cancel{2} \cancel{1} \cancel{0} \, 1$

$nst = \cancel{1} ; \cancel{3} \cancel{5} \, 3$

Output

1 _ _ k
2 _ k k
3 k k k k k
4 _ k k k
5 _ _ k

- _ _ k
- _ k k k
- k k k k k
- _ k k k
- _ _ k

i = 1

Ques : Print the given pattern

```
* * * * * * *
```

1 ` * * * _ * * * `

2 ` * * _ _ _ * * `

3 ` * _ _ _ _ _ * `

$nst = 3$

$nsp = 1$

$nsp += 2$

$nst --$

Number Pyramid Mast

*Ques : Print the given pattern

```
    1   2   3   4   5   6   7
1   1   2   3   4   5   6   7
2   1   2   3   _   5   6   7
3   1   2   _   _   _   6   7
4   1   _   _   _   _   _   7
```

n = 4

n = 2

```
1   2   3
1   _   3
```

n = 3

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

Number Pyramid Mast

```c
for(int i=1;i<=n;i++){    // 3 (crossed out n, replaced with 3)
    int a = 1;
    for(int j=1;j<=nst;j++){ // stars
        printf("%d",a);
        a++;
    }
    for(int k=1;k<=nsp;k++){ // spaces
        printf(" ");
        a++;
    }
    for(int j=1;j<=nst;j++){ // stars
        printf("%d",a);
        a++;
    }
    nst--;
    nsp+=2;
    printf("\n");
}
```

1   1 2 3 _ 5 6 7
2   1 2 _ _ _ 6 7
3   1 _ _ _ _ _ 7

nst = 3 2
nsp = 1 3

i = 1 2
a = 1 2 3 4 5 6 7 8

Output

• 1 2 3 _ 5 6 7
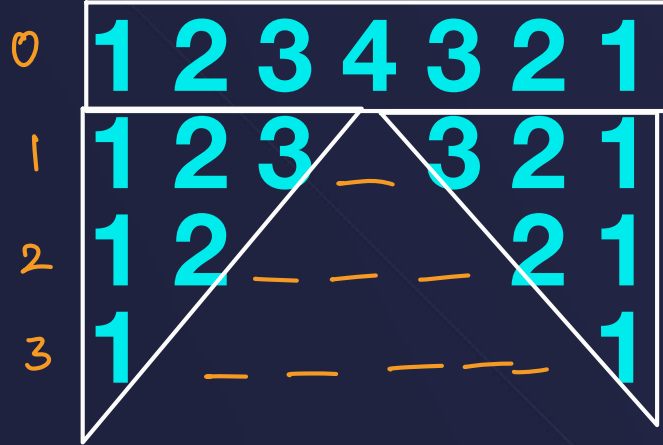• 1 2 _ _ _ 6 7

*HW : Print the given pattern

Same

```
A B C D E F G
A B C     E F G
A B         F G
A             G
```

*HW : Print the given pattern

```
0   1 2 3 4 3 2 1
1   1 2 3   3 2 1
2   1 2       2 1
3   1           1
```

```
1 2 3          —              3 2 1
1 2      +    — — —      +      2 1
1             — — — — —          1
```

Number Pyramid Mast

**Ques : Print the given pattern

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

$n = 4$

'$2n-1$' lines

```
3 3 3 3 3
3 2 2 2 3
3 2 1 2 3
3 2 2 2 3
3 3 3 3 3
```

$n = 3$

**Number Pyramid Mast

	3	2	1
	5	6	7

	1	2	3	4			
1	1	1	1	1	1	1	1
2	1	2	2	2	2	2	1
3	1	2	3	3	3	2	1
4	1	2	3	4	3	2	1
3 5	1	2	3	3	3	2	1
2 6	1	2	2	2	2	2	1
1 7	1	1	1	1	1	1	1
i							

j

n = 4

min

1 → 4
2 → 3
3 → 2
4 → 1

j = 1 2 3 4 5 6 7
b = 1 2 3 4 3 2 1

i = 1 2 3 4 5 6 7
a = 1 2 3 4 3 2 1

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

min(i,j)