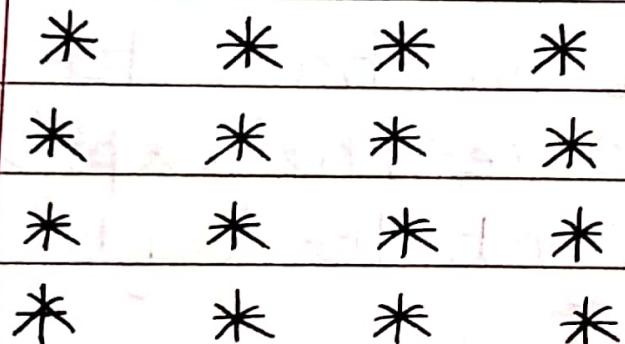


Homework of 3/02/2023

i) Pattern - 1 Solid square pattern



We can observe from the above pattern that there will be same no. of rows &

columns :

$i=0 \rightarrow 4$ stars

$i=1 \rightarrow 4$ stars

$i=2 \rightarrow 4$ stars

$i=3 \rightarrow 4$ stars

No. of rows \rightarrow 0 to 3 } 0 and 3 inclusive

No. of columns \rightarrow 0 to 3 } 0 and 3 inclusive

Code

```
for (int i=0; i<n; i++) {
    for (int j=0; j<n; j++) {
        cout << "*";
    }
    cout << endl;
}
```

2) Pattern-2 Hollow square pattern

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

We can observe that from the above pattern that there are same no. of rows & columns. Also if the row is 1st or last or 1st column or last column, then only we have to print the star, else print space. Outer for loop is same as that of previous pattern.

Code for inner loop

```

for (int j=0; j<n; j++) {
    1st column   last column   1st row
    ↑           ↑           ↑           ↑
    if (j==0 || j==n-1 || i==0 || i==n)
        cout << "* " j;
    else
        cout << " " j;
}

```

3) Pattern - 3 Hollow inverted half pyramid

```

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

```

- 1) If $i == 0$ i.e 1st row , then all the stars are printed
- 2) If $j == 0$ i.e 1st column , then all the stars are printed
- 3) For the respective row , star will be printed for the last column only . In between nothing is printed.

Formulae for last column for the respective row will be one less than the loop is running . If loop run for k times , then last column will be $k-1$.

Code

```

for (int i=0 ; i<n ; i++) {

```

```

for (int j=0; j<n-i-1; j++) {
    1st row      1st column      last column
    ↑           ↑                ↑
    if (i==0 || j==0 || j==n-i-2) {
        cout << "* ";
    }
    else {
        cout << " ";
    }
}
cout << endl;
}

```

Note → 0 to $n-i-1$, last column will be
 $n-i-1 - 1$ i.e. $n-i-2$

4) Pattern - 4 Hollow full pyramid

*	1 star	$2*i+1 \leftarrow$
* * *	2 star + 1 space	① } odd
* * *	3 star + 2 space	③ } ⑤]

1) The stars will be printed for 1st & last column always. In last column, the stars will only be printed for even numbered columns.

2) There are some spaces on to the left which also needs to be printed.

$i=0 \rightarrow$ 1st row $i=1 \rightarrow$ 2nd row $i=2 \rightarrow$ 3rd row	space = 2 space = 1 space = 0	$\} n=3$
---	-------------------------------------	----------

Formulae for Space = $n-i-1$

Code

```

for (int i=0 ; i<n ; i++) {
    int space = n - i - 1;
    for (; space >= 0 ; space--) {
        cout << " ";
    }
    for (int j=0 ; j < 2*i+1 ; j++) {
        if (j==0 || j==2*i || (i==n-1 & j==0)) {
            cout << "*";
        } else {
            cout << " ";
        }
    }
    cout << endl;
}

```

5) Numeric hollow half pyramid Pattern - 5

```

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

```

- 1) For 1st row we just print 1.
- 2) For the last row all the numbers from 1 to n will be printed.

- 3) If not 1st and not last row then first check if we encounter 1st column, then print 1. If we encounter last column, then print one more than the row number & if the column is not 1st & not the last column then just print the space.

Code for inner for loop

```
for (int j=0 ; j < i+1 ; j++) {
    if (j==0)
        cout << 1;
    else if (j==n-1)
        cout << j+1 << " ";
    else {
        if (j==i)
            cout << 1 << " ";
        else if (j==i)
            cout << i+1 << " ";
        else
            cout << " ";
    }
}
```

- 6) Pattern - 6 Numeric follow inverted half pyramid

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

- 1) For 1st row, all numbers will be printed.
- 2) For the last row, n will be printed.
- 3) For other rows, start number will be $i+1$ only. In last column of each respective row, n will be printed.

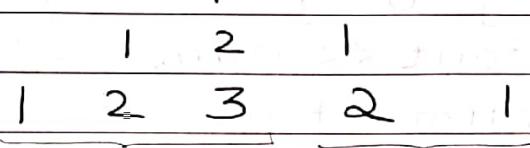
Code for inner for loop

```

for (int i=0 ; i< n-i ; i++) {
    if (i==0) {
        cout << j+1 << " ";
    }
    else if (i==n-1) {
        cout << n;
    }
    else {
        if (j==0) {
            cout << i+1 << " ";
        }
        else if (j==n-i-1) {
            cout << n;
        }
        else {
            cout << " ";
        }
    }
    cout << endl;
}

```

7) Pattern - 7 Numeric Palindrome equilateral triangle



- 1) First we need to print the spaces & we have already seen that how to print these spaces. Formulae for space will be $n-i-1$.
- 2) First we need to print pattern-1. It always start from 1. So we can take a variable num = 1 and run a loop from 0 to $i+1$ & print num.
- 3) Now we need to print pattern-2 which is basically the reverse counting. First initialize num = num - 2 as we have to start from one less than the previous, that's why we have subtracted 2.

Code for inner for loop

Step-1 Print spaces

```
int space = n - i - 1;
for ( ; space >= 0 ; space--) {
    cout << " ";
```

3

Step-2 Print pattern -1

```

int num = 1;
for (int i=0; i<i+1; i++) {
    cout << num << " ";
    num++;
}

```

Step-3 Print pattern -2

```

num = num-2;
for (; num>=1; num--) {
    cout << num << " ";
}

```

8) Fancy pattern #1

```

* * * * * * * 1 * * * * * *
* * * * * * 2 * 2 * * * * * *
* * * * * 3 * 3 * 3 * * * * ;
* * * * 4 * 4 * 4 * 4 * * * ;
* * * 5 * 5 * 5 * 5 * 5 * * * ;

```

 $n = 5$

No. of stars in 1st row in left side = 8 ($i=0$)

2nd row in left side = 7 ($i=1$)

3rd row in left side = 6 ($i=2$)

No. of stars = $n + 3 - i$ where i is
row number

Similarly we can print this on the right
side with same formulae.

Now we have to print numbers & then star pattern which is in between these stars.

$i = 0 \rightarrow 1$ number

(1)

$i = 1 \rightarrow 2$ numbers + 1 star

(3)

$i = 2 \rightarrow 3$ numbers + 2 stars

(5)

} odd times

Loop will run till $j < 2 * i + 1$

- If column is even, then we have to print $i + 1$.
- else we have to print *.

Code for inner for loop

1) Stars on left & right

int stars = $n + 3 - i$;

```
for ( ; stars > 0 ; stars -- ) {
    cout << "*";
```

3

2) Stars and number pattern

```
for (int j = 0 ; j < 2 * i + 1 ; j++) {
```

if ($j \% 2 == 0$)

cout << $i + 1$;

else

cout << "*";

3

S 9) Solid half diamond Pattern - 9

```

    *                                     } Pattern-1
    *   *
    *   *   *
    *   *   *   *
    *   *   *   *   *
    *   *   *   *
    *   *   *   *   } Pattern-2
    *   *
    *
  
```

The diagram shows a solid half diamond pattern. It consists of 6 rows of asterisks (*). The first five rows form a triangle pointing upwards, while the last row is a single horizontal line of asterisks. Braces on the right side group the rows into two parts: 'Pattern-1' covers the top four rows, and 'Pattern-2' covers the bottom three rows.

Pattern-1

$i = 0 \rightarrow 1$ star

$i = 1 \rightarrow 2$ star

$i = 2 \rightarrow 3$ star

So in the i^{th} row, there will be $i+1$ stars.

Code for pattern-1

```

for (int i=0 ; i<n ; i++) {
    for (int j=0 ; j<i+1 ; j++) {
        cout << "* ";
    }
    cout << endl;
}
  
```

Pattern-2

There are now $n-1$ rows in this pattern

$i^{\circ} = 0 \rightarrow 4 \text{ stars}$

$i^{\circ} = 1 \rightarrow 3 \text{ stars}$

$i^{\circ} = 2 \rightarrow 2 \text{ stars}$

$i^{\circ} = 3 \rightarrow 1 \text{ star}$

Formulae = $n - i^{\circ} - 1$

Code for pattern-2

```
for (int i=0; i<n-1; i++) {  
    for (int j=0; j<n-i-1; j++) {  
        cout << "* " ;  
    }  
    cout << endl;  
}
```

12) Floyd's triangle pattern Pattern-12

```
1  
2 3  
4 5 6  
7 8 9 10
```

From observation, we can see that we just have to do counting. So initialize the variable num = 1. Now from observation

$i=0 \rightarrow 1$ number	}	$i+1$ numbers
$i=1 \rightarrow 2$ numbers		
$i=2 \rightarrow 3$ numbers		

Hence loop will run from $j=0$ to $j < i+1$ & simply we have to print num & increment it.

Code of inner for loop int num=1;

```
for (int j=0 ; j < i+1 ; j++) {
```

```
    cout << num << " ";
```

num + + j

3

classmate

91

Date _____
Page _____

14) Pattern-14 Butterfly pattern

Space - 1

P-1	*	-	*	P-2
	*	*	*	
	*	*	*	
	*	*	*	
P-3	*	*	-	P-4
	*	-	-	*

Space - 2

Pattern 1 & Pattern 2

 $i^{\circ} = 0 \rightarrow 1 \text{ star}$ $i^{\circ} = 1 \rightarrow 2 \text{ star}$ $i^{\circ} = 2 \rightarrow 3 \text{ star}$

Pattern 1 & Pattern 2

 $i^{\circ} + 1 \text{ stars}$ Loop will run from $j = 0$ to $j < i^{\circ} + 1$.Code for pattern-1 & pattern-2

```
for (int i=0; i<n; i++) {
    // Pattern - 1
    for (int j=0; j< i+1; j++) {
        cout << "* ";
    }
    // Space - 1
}
```

// Pattern - 2

```
for (int j=0; j< i+1; j++) {
    cout << "* ";
}
cout << endl;
```

Space - 1 $n = 3$

$i = 0 \rightarrow 4$ spaces	$\left\{ \begin{array}{l} \text{ } \\ \text{ } \\ \text{ } \end{array} \right.$				
$i = 1 \rightarrow 2$ spaces					$2 * (n - i - 1)$
$i = 2 \rightarrow 0$ spaces					

Code for space - 1

```
int space = 2 * (n - i - 1);
for (; space > 0; space--) {
    cout << " ";
}
```

Pattern - 3 and 4 $n = 3$

$i = 0 \rightarrow 3$ stars	$\left\{ \begin{array}{l} \text{ } \\ \text{ } \\ \text{ } \end{array} \right.$			
$i = 1 \rightarrow 2$ stars				$n - i$
$i = 2 \rightarrow 1$ star				

The loop will run from $j = 0$ to $j < n - i$.

Code for pattern - 3 & pattern - 4

```
for (int i = 0; i < n; i++) {
    // Pattern 3
    for (int j = 0; j < n - i; j++) {
        cout << "* ";
    }
    // Space - 2
}
```

// Pattern - 4

```
for (int j = 0; j < i + 1; j++) {
    cout << "* ";
}
```

cout << endl;

3

Space - 2

$i = 0 \rightarrow 0 \text{ space}$ }
 $i = 1 \rightarrow 2 \text{ space}$ }
 $i = 2 \rightarrow 4 \text{ space}$ } $2 * i^{\circ} \text{ spaces}$

Code for space - 2

```
int space = 2 * i;  
for ( ; space > 0; space--) {  
    cout << " ";
```

3