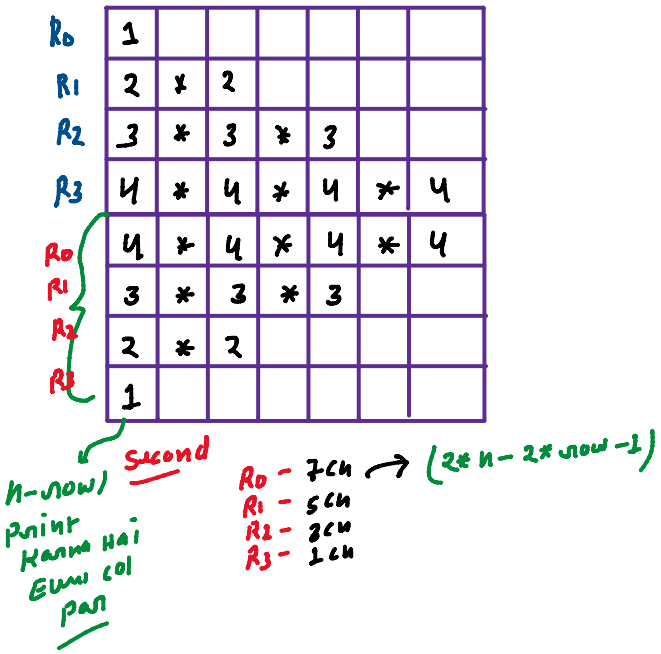
HOMEWORK DAY:04

02 September 2023 10:47

HW CLASS:04 31/08/23

https://www.linkedin.com/in/manojoffcialmj/

PATTERN:16 Fancy pattern 2



Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 4 (row < n)

Step 02: See what is happening in each row (inner loop)

First → col < (2*row + 1)

R_0 = 1ch

R_1 = 3ch

R_2 = 5ch

R_3 = 7ch

CODE OF PATTERN 16

```cpp
// Pattern 16: Fancy pattern 2
#include<iostream>
using namespace std;

int main(){
    int num;
    cin >> num;
    int n=num/2;

    // Todo 1: First
    for(int row=0; row<n; row = row+1){
        for(int col=0; col<(2*row+1); col = col+1){
            if(col%2==0){
                cout << row+1 << " ";
            }
            else{
                cout << "* ";
            }
        }
        cout << endl;
    }
    // Todo 2: Second
    for(int row=0; row<n; row = row+1){
        for(int col=0; col<(2*n-2*row-1); col = col+1){
            if(col%2==0){
                cout << n-row << " ";
            }
            else{
                cout << "* ";
            }
        }
        cout << endl;
    }
    return 0;
}

/*
OUTPUT: when n is 8
1
2 * 2
3 * 3 * 3
4 * 4 * 4 * 4
4 * 4 * 4 * 4
3 * 3 * 3
2 * 2
1
*/
```

PATTERN:18 Fancy Pattern 3

R_0	1					
R_1	1	–	2			
R_2	1	–	–	3		
R_3	1	–	–	–	4	
R_4	1	2	3	4	5	
	C_0	C_1	C_2	C_3	C_4	

When (R == 0) → print 1

When (Row == n-1) → print (1 to row+1)

When (Row != 0) || (Row != n-1) → print

1 – 2
1 – – 3
1 – – – 4

Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 5 (row < n)

Step 02: See what is happening in each row (inner loop)

$R_0 \Rightarrow 1$ } (col < row+1)

$R_1 \Rightarrow 3$ch
$R_2 \Rightarrow 4$ch
$R_3 \Rightarrow 5$ch
} (col < row+2)

$R_5 \Rightarrow 5$ch } (col < n)

CODE OF PATTERN 18

```cpp
// Pattern 18: Fancy pattern 3
#include<iostream>
using namespace std;

int main(){
    int n;
    cin >> n;

    // Outer loop
    for(int row=0; row<n; row = row+1){
        // Inner loop
        if(row==0){
            cout << row+1 << " ";
        }
        else if(row==n-1){
            for(int col=0; col<n; col = col + 1){
                cout << col + 1;
            }
        }
        else{
            for(int col=0; col<row+2; col = col + 1){
                if(col==0){
                    cout << col + 1;
                }
                else if(col==row+2-1){
                    cout << col;
                }
                else{
                    cout << " ";
                }
            }
        }
        cout << endl;
    }
}

/*
OUTPUT: when n is 5
1
1 2
1  3
1   4
12345
*/
```

PATTERN:20 Numerical Hollow Inverted Half Pyramid

R0	1	2	3	4	5	
R1	2	-	-	5		
R2	3	-	5			
R3	4	5				
R4	5					
	C0	C1	C2	C3	C4	

Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 5 (row < n)

Step 02: See what is happening in each row (inner loop)

R0 = 5ch
R1 = 4ch
R2 = 3ch
R3 = 2ch
R4 = 1ch

Total col = (n - row)

When (Row == 0)
↳ Print col+1 to n-1

When (Row == n-1)
↳ Print (row+1) OR Print n

When (Row != 0 || row != n-1)
↳ (col == 0)
↳ Print (row+1)
(col == n - row - 1)
↳ Print n

CODE OF PATTERN 20

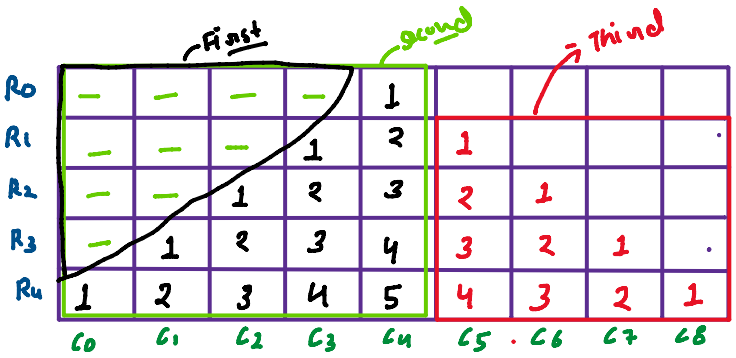
```cpp
// Pattern 20: Numeric Hollow Inverted Half Pyramid
#include<iostream>
using namespace std;

int main(){
    int n;
    cin >> n;

    // Outer Loop
    for(int row=0; row<n; row = row + 1){
        // Inner Loop
        if(row==0){
            for(int col=0; col<n; col = col + 1){
                cout << col + 1 << " ";
            }
        }
        else if(row==n-1){
            cout << n << " ";
        }
        else{
            for(int col=0; col<n-row; col = col + 1){
                if(col==0){
                    cout << row + 1 << " ";
                }
                else if(col==n-row-1){
                    cout << n << " ";
                }
                else{
                    cout << "  ";
                }
            }
        }
        // new line
        cout << endl;
    }
    return 0;
}

/*
OUTPUT: when n is 5
1 2 3 4 5
2     5
3   5
4 5
5
*/
```

PATTERN:21 Numeric Palindrome Equilateral Palindrome



Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 5 (row < n)

Step 02: See what is happening in each row (inner loop)

First

R0 = 4s (n-row-1)
R1 = 3s
R2 = 2s
R3 = 1s
R4 = 0s

Print Space

Second

R0 = 1 ch (Row+1)
R1 = 2 ch
R2 = 3 ch
R3 = 4 ch
R4 = 5 ch

Print (col+1)

Third

R0 = 0 ch (Row)
R1 = 1 ch
R2 = 2 ch
R3 = 3 ch
R4 = 4 ch

Print

Remember remember jab tak answer than zero Rahega.

CODE OF PATTERN 21

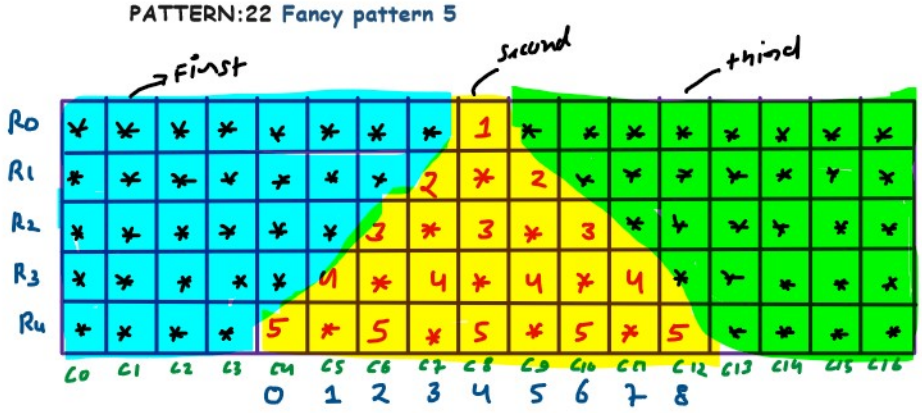
```cpp
// Pattern 21: Numeric Palindrome Equilateral Palindrome
#include<iostream>
using namespace std;

int main(){
    int n;
    cin>>n;

    // Outer loop
    for(int row=0; row<n; row++){
        // Todo 01: First
        for(int col=0;col<(n-row-1);col++){
            cout<<"  ";
        }
        // Todo 02: Second
        int number = 0;
        for(int col=0;col<(row+1);col++){
            number = col+1;
            cout<<number<<" ";
        }
        // Todo 03: Third
        for(number;number>0;){
            number = number-1;
            if(number>0)
            cout<<number<<" ";
        }
        // New line
        cout<<endl;
    }
    return 0;
}

/*
OUTPUT: when n is 5
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
*/
```

PATTERN:22 Fancy pattern 5



Step 01: Count the numbers of row (outer loop)

Numbers of row = (row < n)

n = 5

Step 02: See what is happening in each row (inner loop)

First

R0 = 8 * (n-n+3)

R1 = 7 *

R2 = 6 *

R3 = 5 *

R4 = 4 *

Print (*)

Second

R0 = 1 ch

R1 = 3 ch

R2 = 5 ch

R3 = 7 ch

R4 = 9 ch

(2Row+1)

if(col%2==0){ Print(row+1) } else { Print(*) }

third

R0 = 8ch

R1 = 7ch

R2 = 6ch

R3 = 5ch

R4 = 4ch

→ Repeat First

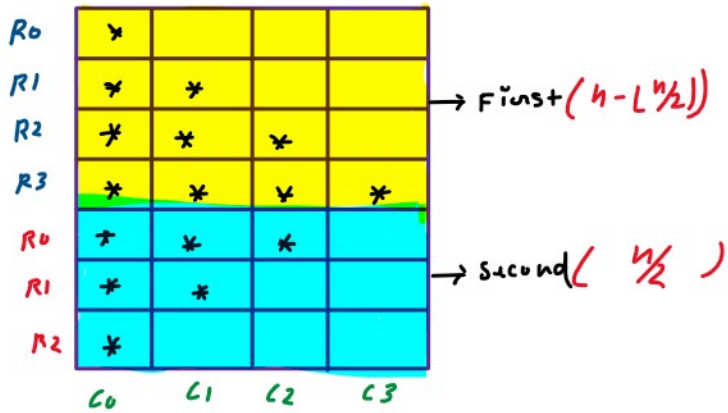
CODE OF PATTERN 22

```cpp
// Pattern 22: Fancy pattern 5
#include<iostream>
using namespace std;

int main(){
    int n;
    cin>>n;
    // Outer loop
    for(int row=0;row<n;row++){
        // Todo 01: First
        for(int col=0;col<(n-row+3);col++){
            cout<< "* ";
        }
        // Todo 02: Second
        for(int col=0;col<(2*row+1);col++){
            // If col is even
            if(col%2==0){
                cout<<(row+1)<<" ";
            }
            else{
                cout<<"* ";
            }
        }
        // Todo 03: Third
        for(int col=0;col<(n-row+3);col++){
            cout<< "* ";
        }
        // New line
        cout<<endl;
    }
    return 0;
}

/*
OUTPUT: when n is 5
* * * * * * * * 1 * * * * * * * *
* * * * * * * 2 * 2 * * * * * * *
* * * * * * 3 * 3 * 3 * * * * * *
* * * * * 4 * 4 * 4 * 4 * * * * *
* * * * 5 * 5 * 5 * 5 * 5 * * * *
*/
```

PATTERN:23 Solid Half Diamond



Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 7

Step 02: See what is happening in each row (inner loop)

First

R0 = 1 * → (row+1)
R1 = 2 * print(*)
R2 = 3 *
R3 = 4 *

Second

R0 = 3 * → (n/2 − row)
R1 = 2 * print(*)
R2 = 1 *

CODE OF PATTERN 23

```cpp
// Pattern 23: Solid Half Diamond
#include<iostream>
using namespace std;

int main(){
    int n;
    cin>>n;

    // Todo 01: First
    for(int row=0;row<(n-(n/2));row++){
        for(int col=0;col<(row+1);col++){
            cout<<"* ";
        }
        // New line
        cout<<endl;
    }
    // Todo 02: Second
    for(int row=0;row<(n/2);row++){
        for(int col=0;col<((n/2)-row);col++){
            cout<<"* ";
        }
        // New line
        cout<<endl;
    }
    return 0;
}

/*
OUTPUT:when n is 7
*
* *
* * *
* * * *
* * *
* *
*
*/
```

PATTERN:24 Floyd Triangle

R0	1					
R1	2	3				
R2	4	5	6			
R3	7	8	9	10		
R4	11	12	13	14	15	
R5	16	17	18	19	20	21
	C0	C1	C2	C3	C4	C5

Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 6 (row < n)

Step 02: See what is happening in each row (inner loop)

R0 = 1 cn (Row+1)
R1 = 2 cn
R2 = 3 cn
R3 = 4 cn
R4 = 5 cn
R5 = 6 cn

Print number is incriased by 1 jab tak (Row<n) na ho jaye

CODE OF PATTERN 24

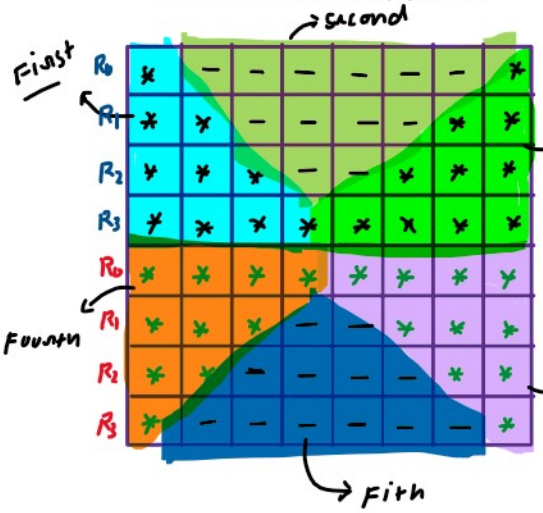
```cpp
// Pattern 24: Floyd Triangle
#include<iostream>
using namespace std;

int main(){
    int n;
    cin>>n;
    int number=0;

    // Outer loop
    for(int row=0;row<n;row++){
        // Inner loop
        for(int col=0;col<(row+1);col++){
            number = number + 1;
            cout<<(number)<<" ";
        }
        // New line
        cout<<endl;
    }
    return 0;
}

/*
OUTPUT: when n is 6
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
*/
```

PATTERN:25 Butterfly pattern



Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 8 (row < n)

Step 02: See what is happening in each row (inner loop)

(n/2)

First

R0 = 1 * (row+1)
R1 = 2 *
R2 = 3 *
R3 = 4 *

Print (*)

Second

R0 = 6 s (n - 2*row - 2)
R1 = 4 s
R2 = 2 s
R3 = 0 s

Print space

Third

↳ Repeat First

Fourth

R0 = 4 * ((n/2) - row)
R1 = 3 *
R2 = 2 *
R3 = 1 *

Print (*)

FIFTH

R0 = 0 s (2* row)
R1 = 2 s
R2 = 4 s
R3 = 6 s

Print space

sixth

↳ Repeat Fourth

CODE OF PATTERN 25

```cpp
// Pattern 25: Butterfly Pattern
#include<iostream>
using namespace std;

int main(){
    int num;
    cin>>num;
    int n=(num/2);

    // TODO PART 01
    for(int row=0;row<n;row++){
        // Todo 01: First
        for(int col=0;col<(row+1);col++){
            cout<<"* ";
        }
        // Todo 02: Second
        for(int col=0;col<(num-(2*row)-2);col++){
            cout<<"  ";
        }
        // Todo 03: Third
        for(int col=0;col<(row+1);col++){
            cout<<"* ";
        }
        // New line
        cout<<endl;
    }
    // TODO PART 02
    for(int row=0;row<n;row++){
        // Todo 04: Fourth
        for(int col=0;col<(n-row);col++){
            cout<<"* ";
        }
        // Todo 05: Fifth
        for(int col=0;col<(2*row);col++){
            cout<<"  ";
        }
        // Todo 06: Sixth
        for(int col=0;col<(n-row);col++){
            cout<<"* ";
        }
        // New line
        cout<<endl;
    }
    return 0;
}

/*
OUTPUT: when n is 8
*             *
* *         * *
* * *     * * *
* * * * * * * *
* * * * * * * *
* * *     * * *
* *         * *
*             *
*/
```