

Week - 2 Assignments.

1. Numeric Hollow Half pyramid.

$$r_0 \rightarrow 1$$

$$r_1 \rightarrow 2$$

$$r_2 \rightarrow 3$$

$$r_3 \rightarrow 4$$

$$r_4 \rightarrow 5$$

$$r_0 \rightarrow 1$$

$$r_1 \rightarrow 1 \quad 2$$

$$r_2 \rightarrow 1 \quad 3$$

$$r_3 \rightarrow 1 \quad 4$$

$$r_4 \rightarrow 1 \quad 2 \quad 3 \quad 4 \quad 5$$

$$col = row + 1$$

and boundary conditions are.

if $i = 0$ or last print col
else

if $col = 0$ or last

print j

else

print space.

2. Numeric Hollow Inverted half pyramid.

Same concep of above

but here

$$col = row - i$$

$$1 \quad 2 \quad 3 \quad 4 \quad 5$$

$$1 \quad \quad \quad 4$$

$$1 \quad \quad 3$$

$$1 \quad 2$$

$$1$$

3. Numeric palindromic Equilateral pyramid.

① we know how
to print
space and
first pyramid

```

      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
  
```

② for second pyramid.

$r_0 \rightarrow 0$ char

$r_1 \rightarrow 1$

$r_2 \rightarrow 2$

$r_3 \rightarrow 3$

$r_4 \rightarrow 4$

formula,

col & row

5. Solid Half Diamond.

① Simply put
the logic of
diamond
patterns, two
times.

```

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

6. fancy pattern # 1

```

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



```

* x x x x x x x 1 x x x x x x x
x x x x x x x 2 x 2 x x x x x x
x x x x x x 3 x 3 x 3 x x x x x
x x x x x 4 x 4 x 4 x 4 x x x x
x x x x 5 x 5 x 5 x 5 x 5 x x x x

```

Here

$r_0 \rightarrow 8 * 1 \text{ char } 8 *$
 $r_1 \rightarrow 7 * 3 \text{ char } 7 *$
 $r_2 \rightarrow 6 * 5 \text{ char } 6 *$
 $r_3 \rightarrow 5 * 7 \text{ char } 5 *$
 $r_4 \rightarrow 4 * 9 \text{ char } 4 *$

for first * stars

$\rightarrow \text{num} = 5$

$\rightarrow n * 2 - \text{row} - 2$

for char and stars.

$\rightarrow 2 * i + 1$

where if col \rightarrow even

then print char

(i.e.) print star(*)

And again for star \rightarrow same formula of first one

$\rightarrow n * 2 - \text{row} - 2$

Fancy pattern #2

- Growing phase.
 (1) first pyramid
 is easy to draw
 as we know how
 we have to
 add only condition
 for giving star
 or characters.

```

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

- (2) Shrinking phase - Put $K = \text{count} - \text{num}$
 let declare a variable
 for first iteration to print
 7, 8, 9, 10

$\text{int } st = \text{num}$
 and then create another variable
 and put the value of
 previous and modify
 with this formula.

$$K = K - (\text{num} - i - 1)$$

Fancy pattern #3

→ Firstly we
 have to print
 stars at the
 boundary
 so that we
 know how to
 do.

```

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


→ & Here's another condition
if

$r = 0 \rightarrow 1$
 $r = 1 \rightarrow 3$
 $r = 2 \rightarrow 5$
 $r = 3 \rightarrow 7$

So we know
we have to
print numbers
straight
and then have to
do reverse.

So..

if (COL <= (2 * row) / 2)

Print ~~star~~ values

else

Print pattern

reverse values
in negative

Pascal's Triangle pattern → already done in class.

Pascal's Triangle pattern.

Pascal A → Binomial Coefficient.

$$C = C * (i - j) / j;$$

$$i = [1, N]$$

$$j = [1, i]$$

1	1				
1	2	1			
1	3	3	1		
1	4	6	4	1	
1	5	10	10	5	1

just put on this formula

Butterfly pattern

① firstly print
first pyramid
with stars

② then print
with space

③ step ①

④ Reverse

```

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

```

→ display area of circle ✓

→ Given no. is even or odd
(Normal and Bitwise Method)

↳ done.

Bitwise method.

① find last set bit

if it is zero then
it's odd else (odd)

→ find the factorial — ✓

→ check Given no. prime or Not.

① Take a variable count = 0

(ii) Loop \rightarrow from $\rightarrow 2$ to $\leq n$
 if it's divisible by any no.
 then its count increased.

(iii) Now if (count) then not prime
 else prime.

\rightarrow Reverse integer \rightarrow just
 \rightarrow find \downarrow no. by modulo
 last-

\rightarrow and then perform operation for
 joining.

\rightarrow Set the Kth bit.

let's take $N=10, K=2$.

① ~~for this we simple~~ 1010
~~right shift by K~~ $\begin{matrix} 2 & 1 & 0 \\ \uparrow & & \end{matrix}$
~~values and~~ \uparrow Set this bit.

② firstly left shift
 1 by K times
 $1 \ll 2$
 $= 100$

② No do bitwise OR b/w N & value

$N \mid 100$

\rightarrow got final value.

→ Convert the Temperature.

Just add the formula.

- Kelvin = Celsius + 273.15
- Fahrenheit = Celsius * 1.80 + 32

→ Count all set bits.

Bitwise

in loop

means -

→ in loop.

① $bit = N \times 2$

② if (bit) count++

① $bit = N + 1$

② if (bit) count++

③ $N = N \gg 1$

③ $N = N / 2$

→ Create No. using digits.

① int num = 0

$NUM = NUM * 10 + digit;$

→ print all digits of int ✓

→ KM to Miles.

$Miles = KM * 1.60934$

✓