Lec-6
2020-05 - 11:30

Take N (number of rows), print the following pattern (for N = 6)

```
0
1 1
2 2 2
3 3 3 3
4 4 4 4 4
5 5 5 5 5 5
```

$n_{C_R} = \frac{n!}{(n-R)!\,R!}$

$S_{C_3} = \frac{5!}{(5-3)!\,3!} = \frac{5\times4\times3}{2\times1}$

$3! = 3\times2\times1$

$10! = 10\times9\times8\times7\times6\times5\times4\times3\times2\times1$

$2! = 2\times1$

Take N (number of rows), print the following pattern (for N = 6)

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

$\frac{1}{2} = \frac{2}{4} = \frac{3}{6} = \frac{4}{8}$

$n_{C_R} = \frac{n!}{(n-R)!\,R!}$ → ① $n_{C_{R+1}} = \frac{n!}{(n-(R+1))!\,(R+1)!}$ (Rule:)

① ÷ ②

$\frac{n_{C_{R+1}}}{n_{C_R}} = \frac{\cancel{n!}}{(n-R)!\,(R+1)!} \times \frac{(n-R)!\,R!}{\cancel{n!}}$

Rule:

$n_{C_{R+1}} = \frac{(n-R)!\,R!}{(n-R)!\,(R+1)!}$

$n! = n\times(n-1)!$

$(n-R)! = (n-R)\times(n-R-1)!$

$n_{C_{R+1}} = \frac{(n-R)!\,R!}{(n-R)!\,(R+1)!}$

$n_{C_{R+1}} = \frac{(n-R)!\,R!}{(n-R)!\,(R+1)\,R!}$

$n_{C_{R+1}} = \frac{R!}{R+1}$

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int n = Integer.parseInt(System.in);
    int c = 1;
    int row = 0;
    int tmp = 0;
    int[][] a = new int[n][];
    while (n > 0) {
        int[] r = a[row] = new int[row + 1];
        r[0] = 1;
        int cr = 1;
        for (int i = 1; i < r.length; i++) {
            r[i] = cr = cr * (row - i + 1) / i;
        }
        n--;
        row++;
    }
    System.out.println();
    //print
    rows();
}
}
```

0 1
1 1 1
2 1 2 1
3 1 3 3 1
4 1 4 6 4 1
5 1 5 10 10 5 1

14 C 4 1

$n_{C_R} = \frac{n!}{r!\,(n-r)!}$

$n_{C_R} = \frac{3!}{2!\,(3-2)!}$

$(n-r)! = \frac{n!}{r!}$

$\frac{(n-r)!}{(n-1)!} = \frac{n!}{r!}$

$n_{C_R} = \frac{n!}{(n-r)!\,r!} = 0$

Number Systems

① decimal — 10 → (0-9)

② Binary — 2 → (0-1)

⑧ octal — (8) (0-8)

⑯ HEX — (16) (0-F)

base-6 0-5

① → convert decimal to binary

(1)(1)(1)

2 8 4 6 →

3 5 6 7

6 4 1 2

$131/10 = 3$

$13/10 = (1)$

$1/10 = 1$

$1/10 = 1$

1×10^3

1×10^2

1×10^1

7×10^0

3×10^3

2×10^2

1×10^1

6×10^0

0111

10111.

10111.

0111.00

+ 1111.1

10111.1

01101.1

100110001

1001001 0

10110 11

0.001 001

$3\times4 = 3\times8^0 + 0\times8^1 + 4\times8^2 + 3\times8^3$

2 | 149	
2 | 74	1 x0
2 | 37	0 x1
2 | 18	1 x2
2 | 9	0 x3
2 | 4	1 x4
2 | 2	0 x5
2 | 1	1 x6
0	1 x7

(203)

2 | 37	1
2 | 18	0
2 | 9	1
2 | 4	0
2 | 2	0
2 | 1	1
0	

100 101

$1\times2^0 + 1\times2^2 + 1\times2^3 + 1\times2^5$

11000

$\text{sum} = 1\times2^0 + 0\times2^1 + 1\times2^2 + 1\times2^3 + 0\times2^4 + 1\times2^5$

$\text{sum} = 1 + 0 + 4 + 8 + 0 + 32$

$\text{sum} = 45$

while (n>0)

int temp = n % 2;

sum = sum + temp * pow(2, i);

n = n/2;

i = i + 1;

2 | 37	1
2 | 18	0
2 | 9	1
2 | 4	0
2 | 2	0
2 | 1	1
0	

100000

100000

11 0 0 1 1

Octal

16 | 6111	
16 | 1011	x0 → 1
16 | 101	x1 → 5
16 | 10	x2 → 2
16 | 1	x3 → 1
0	

10 | 10111	x0 → 7
10 | 1011	x1 → 3
10 | 101	x2 → 5
10 | 10	x3 → 2
10 | 1	x4 → 1
0	

10110110

1 0 11 0 11 0

$128 + 32 + 16 + 1 + 2$

182

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int num = 45;
    int sum = 0;
    int mul = 1;
    int i = 0;
    while (num > 0) {
        int temp = num % 2;
        sum = sum + temp * mul;
        mul = mul * 10;
        num = num / 2;
    }
    System.out.println(sum);
}
```

2 | 45	5×2⁰
2 | 22	4×2¹
2 | 11	1×2²
2 | 5	1×2³
2 | 2	0×2⁴
0	

101101

101110

x0

(5×2⁰)

(5×2⁰)

5×2⁰ = 5 (decimal)

(101101)

GWHILE (n>0)

temp = n%2

sum = sum + temp*mul

mul = mul*10

n = n/2

Data Type

base b=5

101

$2^2 = 3$

3^4

$(0,255)$

$2^8 = 256$

256

$-128 \to 127$

$-128 \to 127$

$[-128, 127]$

base b=5

type	size bit	sign byte	Range
byte	8	1	-2^8 to 2^8-1
short	16	2	-2^{16} to $2^{16}-1$
int	32	4	-2^{32} to $2^{32}-1$
long	64	8	-2^{63} to $2^{63}-1$

-128 to 127

-2^7 to 2^7-1

2^8

1010 0100

0101 1011

1111

0101 1011

int a = 0 33

$2\times4^1 + 0\times4^0 = 2\times4 + 0 = 8$

0000 1000