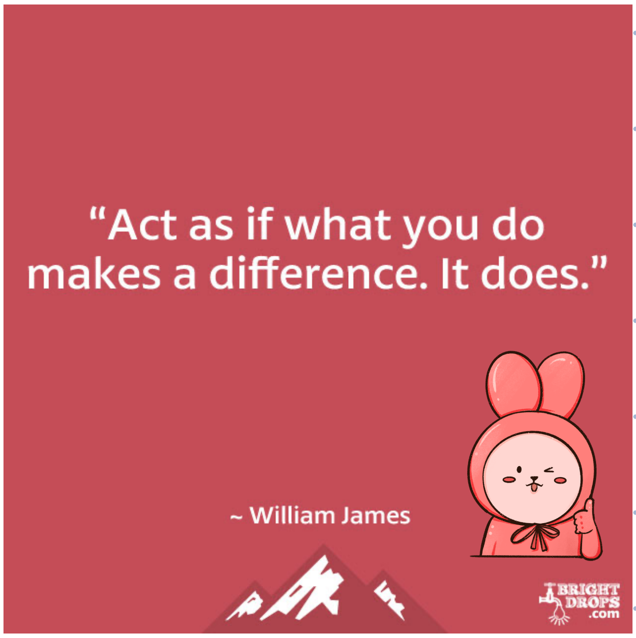
NOTES: Combinatorics & prime



Todays content

- → Addition & Multiplication rule
- → Permutation basics
- → Combination basics
- → Prime no.
- → Sieve of Eratosthenes

DSA 3 & DSA 4

- DSA: Maths: Combinatorics Basics & Prime Numbers
- DSA: Lab Session on Prime Numbers & 2 Pointers
- DSA: Lab Session on Maths & 2 Pointers
- DSA: Backtracking
- DSA: Lab Session on Backtracking
- DSA: Linked List: Sorting and Problems
- DSA: Linked List: Doubly Linked List & Detecting Loop
- DSA: Trees 4: Morris Inorder Traversal + LCA
- DSA: Lab Session on Binary Trees 2
- DSA: Hashing 3: Internal Implementation & Problems
- Contest 3: Math, Two Pointers, Backtracking, Linked List & Trees
- DSA: Heaps: Introduction
- DSA: Heap Sort & Greedy
- DSA: Lab Session on Heaps & Greedy
- DSA: Lab Session on Interview Problems 1
- DSA: DP 1: One Dimensional
- DSA: DP 2: Two Dimensional
- DSA: DP 3: Knapsack
- DSA: Lab Session on Applications of Knapsack
- DSA: Graphs 1: Introduction, DFS & Cycle Detection
- DSA: Graphs 2: BFS & MST
- DSA: Graphs 3: Dijkstra Algo & Topological Sort
- DSA: Lab Session on Interview Problems 2
- DSA: Revision of DSA 3 & 4
- DSA: Contest 4: Heaps, Greedy, DP & Graphs
- Contest 5: Full Syllabus

Addition & Multiplication

01. Given 3 True/False questions, every question has to be answered. In how many ways can you answer all the questions.

$$\underline{2} * \underline{2} * \underline{2} = 2^3 = 8 \text{ ways}$$

T F T F T F

T F F

T F T

T T F

F F F

F F T

F T F

F T T

T T T

8 ways

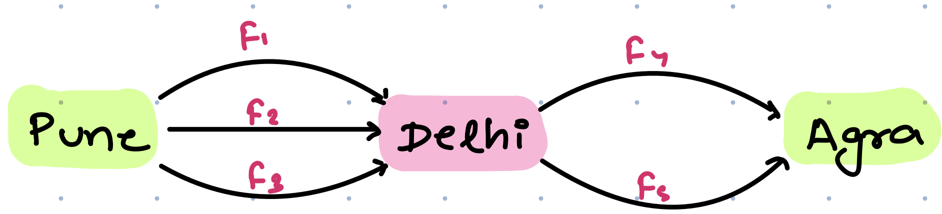
Q2 Given 10 girls & 7 boys. How many different pairs can we form?

Note :- pair → $\boxed{\text{1 Girl + 1 Boy}}$

Girls	Boys
G_1	B_1
G_2	B_2
G_3	B_3
G_4	⋮
⋮	B_7
G_{10}	

pair → $\underbrace{\underbrace{\text{1 Boy}}_{7} \;\&\; \underbrace{\text{1 Girl}}_{10}}_{\text{70 ways}}$

Eg 3



No. of ways to reach Agra from Pune via Delhi

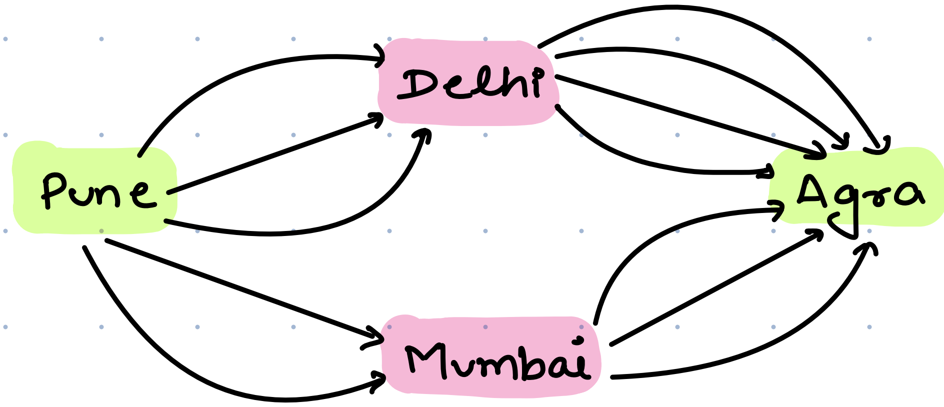
$\underbrace{\text{Pune} \longrightarrow \text{Delhi}}_{3}$ AND $\underbrace{\text{Delhi} \longrightarrow \text{Agra}}_{2}$

3 * 2 = 6 ways

$F_1 F_4$ $F_2 F_4$ $F_3 F_4$

$F_1 F_5$ $F_2 F_5$ $F_3 F_5$

Eg 4



No. of ways to reach Agra from pune ?

via Delhi **OR** via Mumbai

$\underbrace{\text{Pune} \rightarrow \text{Delhi}}_{3}$ & $\underbrace{\text{Delhi} \rightarrow \text{Agra}}_{4}$ $\underbrace{\text{Pune} \rightarrow \text{Mumbai}}_{2}$ & $\underbrace{\text{Mum} \rightarrow \text{Agra}}_{3}$

$3 * 4$ $2 * 3$

$= 12 \quad + \quad 6$

$\Rightarrow$ 18 ways

AND (*) $\Rightarrow$ Used to calculate possibilities occuring together in sequence

OR (+) $\Rightarrow$ Used to count possibilites occuring separately

Most Varied Meal Combo

FIGQLUT

Scenerio

Zomato, features an exciting option for its users - meal combos. Each combo includes one main course, one *dessert*, and one *beverage*, offering a complete dining experience from various restaurants. Zomato believes that a greater variety of combos can significantly enhance customer satisfaction.

Problem

You're tasked with helping **Zomato** identify which restaurant offers the most variety in its meal combos. You're provided with a list, shaped like a grid or a 2D matrix **A**, where each row corresponds to a different restaurant's offerings.

Each row is divided into three parts:

1. A[i][0] tells you the number of main courses,
2. A[i][1] the number of desserts, and
3. A[i][2] the number of beverages a restaurant offers.

Your challenge is to analyze this data and pinpoint which restaurant gives its customers the most options to mix and match their meal combo.

```
A = [
    [3, 2, 2],   # Restaurant 1
    [4, 3, 3],   # Restaurant 2
    [1, 1, 1]    # Restaurant 3
]
```

→ 3 * 2 * 2

→ 4 * 3 * 3 ⇒ 36 ways

→ 1 * 1 * 1

Permutation → Arrangements of objects

→ Order matters (i, j)

(i, j) ≠ (j, i)

Given 3 distinct characters, in how many ways can we arrange them?

str = "abc"

$\underline{3} * \underline{2} * \underline{1} = 3! = 6$ ways

a → b → c = abc

a → c → b = acb

b → a → c = bac

b → c → a = bca

c → a → b = cab

c → b → a = cba

In how many ways n distinct characters can be arranged?

$N * (N-1) * (N-2) * \dots 1 = N!$ ways

Q Given 5 distinct characters, in how many ways we can arrange 2 characters

str= " a b c d e "

$\underline{5} * \underline{4} = 20$ ways

Q Given n distinct characters, in how many ways we can arrange 3 characters

$N * (N-1) * (N-2)$ ways

Q Given n distinct characters, in how many ways we can arrange r characters

$$N * (N-1) * (N-2) * \ldots * N-(r-1)$$

$$\Rightarrow \frac{N * (N-1) * (N-2) * \ldots * N-r+1 * (N-r) * (N-r-1) * \ldots * 1}{N-r * (N-r-1) * \ldots * 1}$$

$$\frac{N!}{(N-r)!} = {}^{N}P_{r}$$ ⇒ No. of ways to arrange r characters from N distinct characters

Combination → No. of ways to select something
→ Order doesn't matter

Q Given 4 players, no. of ways to select 3 players

$\{P_1 \quad P_2 \quad P_3 \quad P_4\}$

P_1	P_2	P_3	
P_1	P_2		P_4
P_1		P_3	P_4
	P_2	P_3	P_4

Q Given 4 players, no. of ways to arrange 3 players

P_1	P_2	P_3	P_1	P_2	P_4	P_1	P_3	P_4	P_2	P_3	P_4
P_1	P_2	P_3	P_1	P_2	P_4	P_1	P_3	P_4	P_2	P_3	P_4
P_1	P_3	P_2	P_1	P_4	P_2	P_1	P_4	P_3	P_2	P_4	P_3
P_2	P_1	P_3	P_2	P_1	P_4	P_3	P_1	P_4	P_3	P_2	P_4
P_2	P_3	P_1	P_2	P_4	P_1	P_3	P_4	P_1	P_3	P_4	P_2
P_3	P_1	P_2	P_4	P_1	P_2	P_4	P_1	P_3	P_4	P_2	P_3
P_3	P_2	P_1	P_4	P_2	P_1	P_4	P_3	P_1	P_4	P_3	P_2

For every selection → 3! arrangements

No. of selection * No. of arrangements for each selection = Total no. of arrangements

$^4P_3$

$$\Rightarrow x * 3! = \frac{4!}{(4-3)!}$$

$$\Rightarrow x * 3! = \frac{4!}{(4-3)!}$$

$$\Rightarrow x = \frac{4!}{3! * (4-3)!} \Rightarrow \text{4 selection}$$

Q Given N players, count the no. of ways to select r players

Arrange r players = $r!$

Total no. of arrangement = $^{N}P_{r} = \dfrac{N!}{(N-r)!}$

No. of selection * No. of arrangements for each selection = Total no. of arrangements

$$x * r! = \frac{N!}{(N-r)!}$$

$$x = \frac{N!}{r!\,(N-r)!}$$

$$^{N}C_{r} = \frac{N!}{r! * (N-r)!}$$

(No. of ways to select r items out of N)

Properties of Combination

01. No. of ways to select 0 items out of N

$$^{N}C_{0} = \frac{N!}{0!\,(N-0)!} = \frac{N!}{N!} = 1 \text{ way}$$

02. No. of ways to select N items out of N

$$^{N}C_{N} = \frac{N!}{N!\,(N-N)!} = \frac{N!}{N!} = 1 \text{ way}$$

03. No. of ways to select r items out of N

$$^{N}C_{r} = \frac{N!}{r!\,(N-r)!}$$

04. No. of ways to select (N-r) items out of N

$$^{N}C_{N-r} = \frac{N!}{(N-r)!\;N-(N-r)!} = \frac{N!}{(N-r)!\;r!}$$

$$^{N}C_{r} = {}^{N}C_{N-r}$$

* **Special property of Combination**

Select r items out of N item

___ ___ ___ ___

select → ${}^{N-1}C_{r-1}$

reject → ${}^{N-1}C_{r}$

$$ {}^{N}C_{r} = {}^{N-1}C_{r-1} + {}^{N-1}C_{r} $$

Prime numbers → No. which only have $\underbrace{\text{2 factors}}_{\text{1 \& itself}}$

N = 2 → Yes

N = 5 → Yes

N = 1 → No

N = 10 → No

Q Given a no. N, check if it is prime or not.

Idea → Get factors count.

if (count == 2) → prime

```
boolean checkprime (int N)
{
    int cnt = 0
    for (i=1; i*i <= N; i++)
    {
        if (N % i == 0)
        {
            if (i == N/i) cnt ++;
            else   cnt += 2
        }
    }
    if (cnt == 2) return true;
    else return false;
}
```

TC : $O(\sqrt{N})$
SC : $O(1)$

Q Given a no. N, print all primes from 1 to N.

N = 5 → 2, 3, 5

N = 14 → 2, 3, 5, 7, 11, 13

N = 10 → 2, 3, 5, 7

Brute force → Iterate from 1 to N & check if a no. is prime or not.

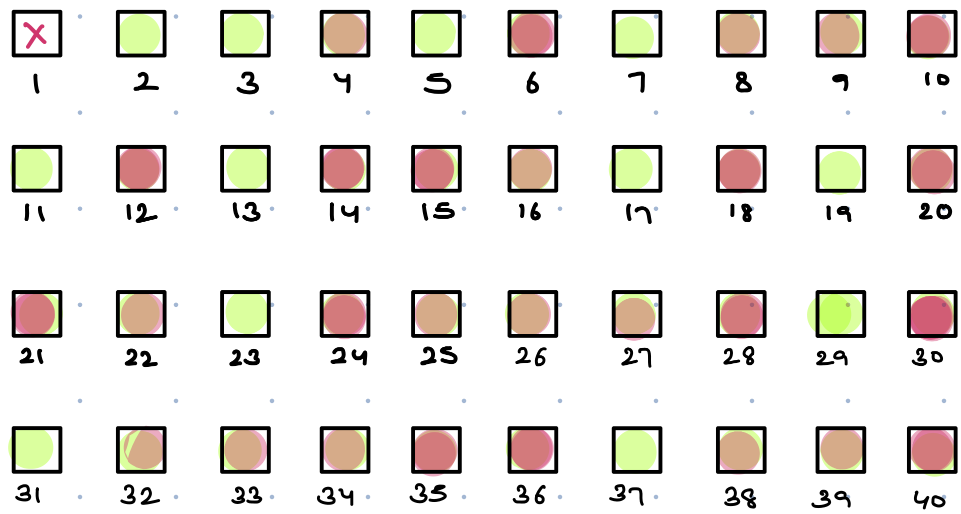
```
for (i=1; i<=N; i++)
{
    if (checkprime(i) == true) print(i);
}
```

TC : $O(N\sqrt{N})$
SC : $O(1)$

* Sieve of Eratosthenes

Find all the primes from 1 to 40

Assumption → Every no is a prime no



If 2 is a prime → Multiples of 2 are never going to be prime

```
void allprimes (int N)
|   boolean [] prime = new boolean [N+1];
|   Arrays.fill(prime, true)
|   prime[0] = false
|   prime[1] = false
|
|   for (i=2; i*i <= N; i++)
|   |   if (prime[i] == true)
```

TC: $O(N \log(\log N))$

SC: $O(N)$

```
        for ( j = i*i ; j ≤ N ; j += i )
        {
            prime [j] = false ;
        }
    }
}
for ( i = 2 ; i ≤ N ; i++ )
{
    if ( prime [i] == true ) print (i) ;
}
}
```

i	Multiples of i to mark them as false					
2	2 * 2	2 * 3	2 * 4	2 * 5	2 * 6	. .
3	3 * 2	3 * 3	3 * 4	3 * 5	3 * 6	. . .
4						
5	5 * 2	5 * 3	5 * 4	5 * 5		

Conclusion → To mark multiples of i as false, start from i * i .

$i\,[2 \rightarrow N]$ $\qquad$ $j = [i^2 \rightarrow N]$

2 $\qquad$ $\frac{N}{2}$ iterations

3 $\qquad$ $\approx \frac{N}{3}$ iterations

4

5 $\qquad$ $\approx \frac{N}{5}$ iterations

$\vdots$

$\sqrt{N}$ $\qquad$ $[(\sqrt{N}^2) \rightarrow N] = 1$ iteration

$\sqrt{N}+1$ $\qquad$ $[(\sqrt{N}+1)^2 \rightarrow N] = 0$ iteration

$\vdots$

N $\qquad$ 0 iterations

Total iterations $= \frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \cdots 1$

$$= N\underbrace{\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \cdots + \frac{1}{N}\right)}_{\text{Sum of reciprocals of prime No.}}$$

$$\Rightarrow N * \log(\log N)$$

$N = 10^8$

$N\sqrt{N} \longrightarrow N \log(\log N)$

$= 10^8 + 10^4$

$= 10^{12}$

$= 10^8 \log(\log 10^8)$

$= 10^8 + \log 8$

$= 10^8 * 0.9 \simeq 10^8$