A % B → Remainder when A / B.

10 % 3 = 1 $\qquad$ 10 = 3 * 3 + 1 (3 → q, 1 → r)

D = q * d + r

dividend = q * divisor + r $\qquad$ int → $2 * 10^9$ $\qquad$ long → $9 * 10^{18}$

6 = 1 * 3 + 3 x

= 2 * 3 + 0

0 <= A % B <= B - 1

Properties

1) (a + b) % m = (a % m + b % m) % m

(① a + b, ② % m)

a = 9 $\qquad$ (we can store <= 10)

b = 8 $\qquad$ (9 + 8) % 5 = 17 % 5 → 2

m = 5 $\qquad$ 9 % 5 = 4

8 % 5 = 3 $\qquad$ (4 + 3) % 5 = 7 % 5 = 2

2) (a * b) % m = (a % m) * (b % m) % m

3) (a - b) % m = (a % m − b % m + m) % m

(0 $\qquad$ m − 1 $\qquad$ + m) > 0

a = 7 $\qquad$ b = 10 $\qquad$ m = 5

(a - b) % m = (7 - 10) % 5 = -3 % 5 → Python → 2 ✓

-3 + 5 = 2

Java/C++ → -3

a % m = 7 % 5 = 2

b % m = 10 % 5 = 0

(2 - 0) % 5 = 2 ✓

4) $(a^b) \% m = (a \% m)^b \% m$

$6^5 \% 8 \rightarrow 6 * (36)^2 \% 8$

$x^n \rightarrow x * x^{n/2} * x^{n/2}$ (n → odd)

$= x * (x^2)^{n/2}$

$= 6 * (36 \% 8)^2 \% 8 = 6 * (4^2) \% 8 = 6 * 16 \% 8$

$= (6 * 0) \% 8 = \underline{0}$

Q → $(37^{103} - 1) \% 12 = ((37^{103}) \% 12 - (1 \% 12) + 12) \% 12$

$= ((37 \% 12)^{103} - 1 + 12) \% 12$

$= (1 - 1 + 12) \% 12 = \underline{0}$ (Ans)

> if (A < B)
>
> A % B = A

Q → $(25 + 13) \% 7 = ((25 \% 7) + (13 \% 7)) \% 7$

$= (4 + 6) \% 7 = 10 \% 7 = \underline{3}$

Q → Given an integer array, find count of pairs (i, j) s.t $(A[i] + A[j]) \% M = 0$ & $i < j$

	0	1	2	3	4	5
A =	4	3	6	3	8	12

M = 6

i	j	
0	4	$(4+8) \% 6 = 0$
1	3	$(3+3) \% 6 = 0$
2	5	$(6+12) \% 6 = 0$

Ans = 3

Bruteforce → TC = $O(N^2)$ SC = $O(1)$

Solution → $(A[i] + A[j]) \% M = 0$ $[0 \quad M-1]$

$$= ((A[i] \,\%\, M) + (A[j] \,\%\, M)) \,\%\, M = 0$$

($M-1$ over $A[i] \,\%\, M$; $M-1$ over $A[j] \,\%\, M$)

$\{0, M\}$

$M = 6$

	0	1	2	3	4	5	6	7	8	9	10
A = [	2	3	4	8	6	15	5	12	17	7	18]
A[i] % M →	2	3	4	2	0	3	5	0	5	1	0

x freq (x)

0 → 3 → sum = 0 ⇒ ${}^3C_2 = \frac{3*(3-1)}{2} = 3$

1 → 1

2 → 2

3 → 2 → ${}^2C_2 = 1$

4 → 1

5 → 2

$f(x) * f(M-x)$

$2*1 = 2$

$1*2 = 2$

Ans = 3 + 2 + 2 + 1 = 8

$M = 6$

	0	1	2	3	4	5	6	7	8	9	10
A = [	2	3	4	8	6	15	5	12	17	7	18]
A[i] % M →	2	3	4	2	0	3	5	0	5	1	0

$6-1 = 5$

0 → 2

1 → 1

2 → 2

3 → 2

4 → 1

5 → 2

∀x, find freq (M-x)

(A[i] % M)

ans = 1 + 1 + 1 + 1 + 2 + 2

= 8

```
∀i, freq[i] = 0
ans = 0
for i → 0 to (N-1) {
    x = A[i] % M
    y = (M-x) % M   // (M-0) % M = 0
    ans += freq(y)
    freq[x]++
}                  ↳ [0  M-1]
return ans
```

TC = O(N) SC = O(M)

GCD (Greatest Common Divisor)

gcd(x, y) = d ⇒ x % d = 0 & y % d = 0

gcd(15, 25) = 5

gcd(12, 30) = 6

{1, 2, 3, 6} → common factors

gcd(0, 4) = 4

1 1
2 2
3 4
4
⋮

gcd(0, x) = x

gcd(0, 0) = ∞

gcd(4, 7) = 1

Properties

1) gcd(0, x) = x ✓
2) gcd(x, y) = gcd(y, x) ✓
3) gcd(x, y, z) = gcd(x, gcd(y, z)) ✓
   = gcd(y, gcd(x, z)) = gcd(z, gcd(x, y))

4) $gcd(x, y) = gcd(x-y, y)$ ✓

$(x > y)$

5) $gcd(x, y) = gcd(x-y, y)$

$= gcd(x-y-y, y)$

$= gcd(x-y\ldots, y)$

$= gcd(x\%y, y)$ ✓

$gcd(100, 15) = gcd(100\%15, 15) = gcd(10, 15)$

gcd (15 , 21 , 33 , 45)

15	21	33	45
1	1	1	1
3	3	3	3 ← Ans = 3
5	7	11	5
15	21	33	9
			15
			45

Find gcd (x, y)

```
g = 1
for i → 2 to min(x, y) {
    if (x%i == 0 && y%i == 0)
        g = i
}
return g
```

TC = O(min(x, y))

$gcd(100, 15) = gcd(100\%15, 15) = gcd(10, 15)$

$= gcd(15, 10) = gcd(15\%10, 10) = gcd(5, 10)$

$= gcd(10, 5) = gcd(10\%5, 5) = gcd(0, 5) = 5$

$gcd(x, y) = gcd(x \% y, y)$

$= gcd(y, x \% y)$

$gcd(153, 971) = gcd(971, 153 \% 971 = 153)$ // $x < y \Rightarrow$ swap

$= gcd(153, 971 \% 153 = 53)$

$= gcd(53, 153 \% 53 = 47)$

$= gcd(47, 53 \% 47 = 6)$

$= gcd(6, 47 \% 6 = 5)$

$= gcd(5, 6 \% 5 = 1) = gcd(1, 5 \% 1 = 0) = \underline{1}$

```
int gcd(x, y) {        // 5, 12
    if (y == 0) return x
    return gcd(y, x % y)   // 12, 5 % 12 = 5
}
```

$TC = \underline{O(\log(\min(x, y)))}$ ✓

$gcd(1538276, 1538275)$

$= gcd(1538275, 1538276 \% 1538275 = 1)$

$= gcd(1, 1538275 \% 1 = 0) = \underline{1}$

Q → Given an integer array, find max gcd of the array after deleting an element.

$A = [\underset{0}{24}\ \underset{1}{16}\ \underset{2}{18}\ \underset{3}{30}\ \underset{4}{15}]$

					GCD
X	16	18	30	15	1
24	X	18	30	15	3 (Ans)
24	16	X	30	15	1
24	16	18	X	15	1
24	16	18	30	X	2

```
          0   1   2   3
A = [ 21   7   2   14 ]      gcd = 7
```

Bruteforce → ∀ i, iterate & find gcd excluding A[i].

TC = O(N * N log (A[i]))

```
          0   1   2   3   4
i   A = [ 24  16  18  30  15 ]
0         x   16  18  30  15
1         24  x   18  30  15
2         24  16  y   30  15
3         24  16  18  x   15      x = gcd(y, 18)
4         24  16  18  30  x       gcd(x, 30)
```

gcd excluding A[i] = gcd (P[i-1], S[i+1])

```
P[0] = A[0]
for i → 1 to (N-1) {
    P[i] = gcd (P[i-1], A[i])
}
S[N-1] = A[N-1]
for i → N-2 to 0 {
    S[i] = gcd (S[i+1], A[i])
}
ans = S[1]   // excluding A[0]
for i → 1 to (N-2) {
    ans = max (ans, gcd (P[i-1], S[i+1]))
}
```

ans = max(ans, P[N-2]) // excluding A[N-1]

return ans

TC = O(N log(A[i])) SC = O(N)

To prove gcd(x, y) = gcd(x-y, y)

Let gcd(x, y) = d x % d = 0 y % d = 0

⇒ (x-y) % d = 0

⇒ d is factor of x, y, (x-y).

Let gcd(x-y, y) = t (x-y) % t = 0 y % t = 0

⇒ (x-y+y) % t = 0

= x % t = 0

⇒ t is a factor of x, y, (x-y).

d is common factor of (x-y), y

& t is greatest common factor ⇒ d <= t

t is common factor of x, y

& d is greatest common factor ⇒ t <= d

> d = t

Hence Proved!

Combinatorics Basics

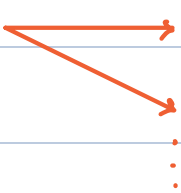
TABLE OF CONTENTS

.

< *Question* > : Given 10 Girls and 7 Boys. How many different pairs of 1:1 can be formed?

Pair → 1 Girl & 1 Boy

Boys	Girls
B_1	G_1
B_2	G_2
B_3	G_3
B_4	G_4
B_5	G_5
B_6	G_6
B_7	G_7
	G_8
	G_9
	G_{10}



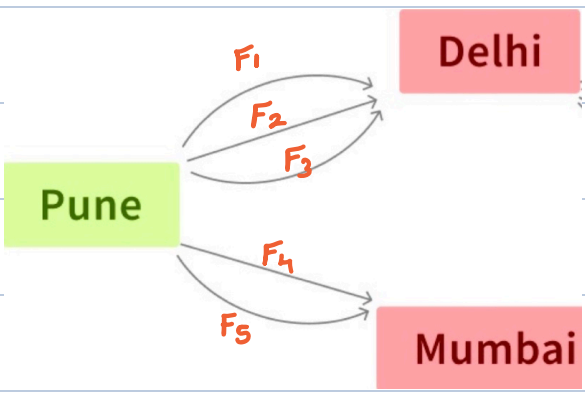
ways = 7 * 10 = 70

Task A → n ways

Task B → m ways

Task A & B → n * m ways

< *Question* > : Find number of ways to travel from Pune to Delhi or Mumbai?



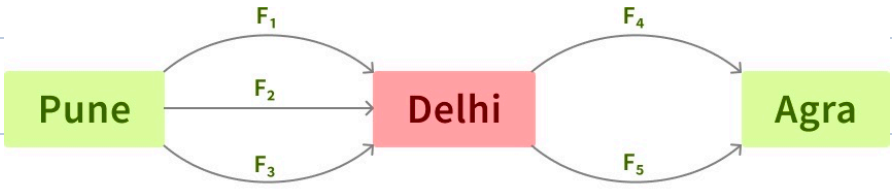
3 + 2 = 5

Task A → n ways

Task B → m ways

Task A or B → n + m ways

Example :



Number of ways to reach Agra from Pune via Delhi

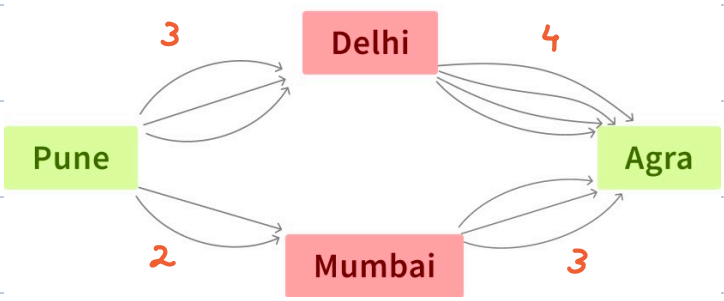
P ⟶ D ⇒ 3 ways

D ⟶ A ⇒ 2 ways

P ⟶ D & D ⟶ A ⇒ 3 * 2 = 6 ways

Number of ways of reaching Agra from Pune ?

Quiz :



P ⟶ D ⟶ A ⇒ 3 * 4 = 12

P ⟶ M ⟶ A ⇒ 2 * 3 = 6

P ⟶ D ⟶ A or P ⟶ M ⟶ A ⇒ 12 + 6 = 18

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.

Examples

Example 1 :

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

Ans = $\forall i$, max (A[i][0] * A[i][1] * A[i][2])

Output : 2

Explanation for input 1 :

- Restaurant 1 offers 12 combos (3 mains x 2 desserts x 2 beverages) → 12
- Restaurant 2 offers 36 combos (4 mains x 3 desserts x 3 beverages) → 36
- Restaurant 3 offers 1 combo (1 main x 1 dessert x 1 beverage) → 1

So, Restaurant 2 provides the most variety with 36 possible combinations.

Permutations → Arrangement of objects

< *Question-1* > : Given 3 distinct characters. In how many ways, we can arrange them?

S = " a b c "

a b c b a c c a b

a c b b c a c b a

Ans = 6

< Question-2 > : In how many ways, you can arrange 4 distinct characters?

s = "d a t e"

↓ ↓ ↓ ↓

4 & 3 & 2 & 1

⇒ 4 * 3 * 2 * 1 = 24

< Question-3 > : In how many ways n distinct characters can be arranged?

↓ ↓ ↓ ↓

N N-1 N-2 . . . 1

ways = N * (N-1) * (N-2) * . . . * 1

= N!

< Question-4 > : Given 4 distinct characters, in how many ways can we arrange 2 ?

s = "d a t e"

↓ ↓

4 & 3

ways = 4 * 3 = 12

< Question-5 > : In how many ways can we arrange R out of N distinct characters?

1 2 R

N N-1 . . . N-(R-1)

R = 4

N N-1 N-2 N-3

ways = N * (N-1) * (N-2) * . . . (N-(R-1))

N * (N-1) * (N-2) * . . . (N-(R-1)) * (N-R) * (N-R-1) * . . . 2 * 1

(N-R) * (N-R-1) * . . . 2 * 1

$= \frac{N!}{(N-R)!} \rightarrow {}^{N}P_{R}$

Combinations → selection of objects

< Question > : In how many ways can we select 3 players from a pool of 4 players?

[P1 P2 P3 P4]

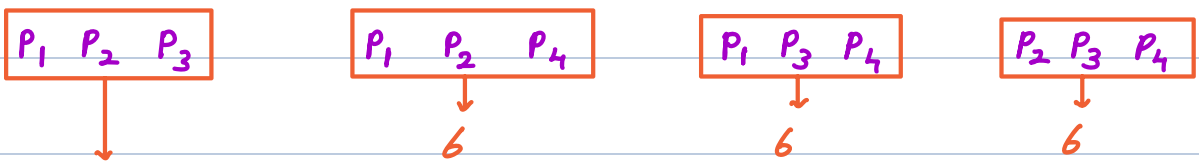
P_1 P_2 P_3 P_1 P_3 P_4 # ways = 4

P_1 P_2 P_4 P_2 P_3 P_4

< Question > : What is the number of ways to arrange the players in 3 slots ?

Given 4 players → [P1 P2 P3 P4]

$$Ans = {}^4P_3 = \frac{4!}{(4-3)!} = \frac{24}{1} = 24$$



$P_1\ P_2\ P_3$

$P_1\ P_3\ P_2$

$P_2\ P_3\ P_1$

$P_2\ P_1\ P_3$

$P_3\ P_1\ P_2$

$P_3\ P_2\ P_1$

6

ways = 4 * 6 = 24

Arrange R out of N items = Select R out of N items & Arrange selected R items

$$\boxed{{}^NP_R = {}^NC_R * R!}$$

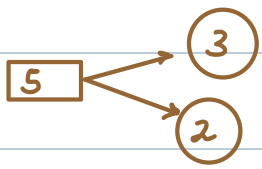
$$\Rightarrow \boxed{{}^NC_R = \frac{{}^NP_R}{R!} = \frac{N!}{(N-R)! * R!}}$$

Properties

1. ${}^{N}C_{0}$ → Select nothing from N items → 1 $\quad \frac{N!}{(N-0)! * 0!} \to 1$

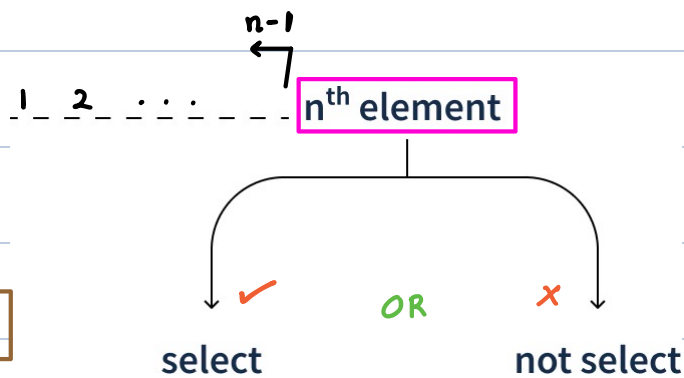
2. ${}^{N}C_{N}$ → Select everything → 1 $\quad \frac{N!}{(N-N)! * N!}$

3. ${}^{N}C_{N-r}$ → Select (N-R) items



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

< *Question* > : Given N distinct elements, select r distinct elements. → ${}^{N}C_{R}$



$x! = x * (x-1)!$

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

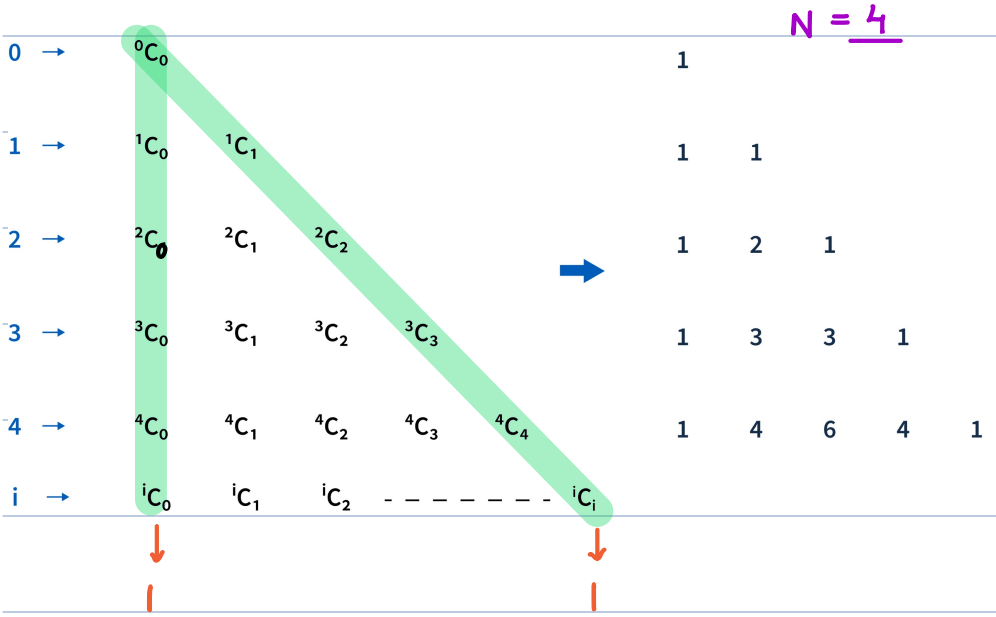
$$\frac{(N-1)!}{(N-1-(R-1))! * (R-1)!} + \frac{(N-1)!}{(N-1-R)! * R!} = \frac{(N-1)!}{(N-R-1)! * (R-1)!} \left(\frac{1}{N-R} + \frac{1}{R}\right)$$

$\frac{1}{N-R} + \frac{1}{R} = \frac{R + N - R}{(N-R) * R} = \frac{N}{(N-R) * R}$

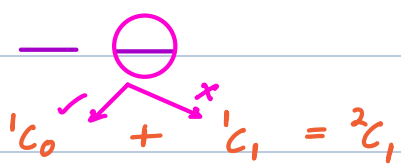
$\frac{(N-1)!}{(N-R-1)! * (R-1)!} * \frac{N}{(N-R) * R} = \frac{N!}{(N-R)! * R!} = {}^{N}C_{R}$

Pascal Triangle

- Generate the Pascal's triangle for given N



${}^{2}C_{1} = {}^{1}C_{0} + {}^{1}C_{1}$



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

```
for i → 0 to N {
    c[i][0] = 1        c[i][i] = 1
    for j → 1 to (i-1) {
        c[i][j] = (c[i-1][j-1] + c[i-1][j]) % M
    }
} return c
```

TC = $O(N^2)$ SC = $O(1)$

Nth Column Title

- Find the Nth column title

N =	1	2	3		26	27	28		50	51	52	53	54		
	A	B	C	- - -	Z	AA	AB	- - - - - - - - - -	AX	AY	AZ	BA	BB	- - -	BZ

N = 30 → AD ✓

N = 50 → AX ✓

N = 100 → CV ✓

int → binary

10 → 2

10 → 26

N = 50 = 110010

A → 1 0

B → 2

C → 3

Z → 26 25

N = 500 → SF

26	500	6	F
26	19	19	S
	0		

2	50	0
2	25	1
2	12	0
2	6	0
2	3	1
2	1	1
	0	

N = 100 → CV

26	100	22	V
26	3	3	C
	0		

0 <= A % 26 <= 25

N = 26

26	26 - 1	25 → Z
	0	

N = 27

26	27 - 1 = 26	0
26	1 - 1 = 0	0
	0	

↑ AA

ans = " "

while (N > 0) {

    ch = (char) ((N-1) % 26 + 'A')

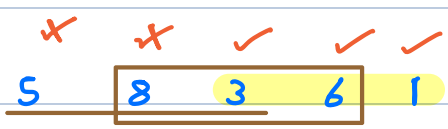
    ans = ch + ans

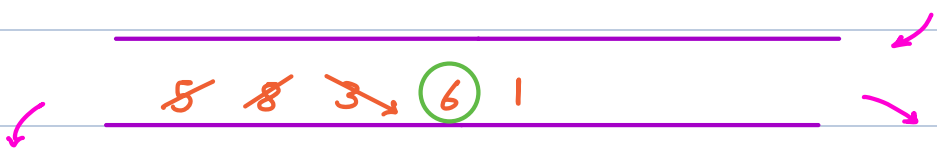
    N = (N-1) / 26

}

return ans

TC = O($\log_{26}$ (N)) SC = O(1)





Ans = q.front()

SCALER

Prime Numbers Advanced

TABLE OF CONTENTS

Prime Numbers

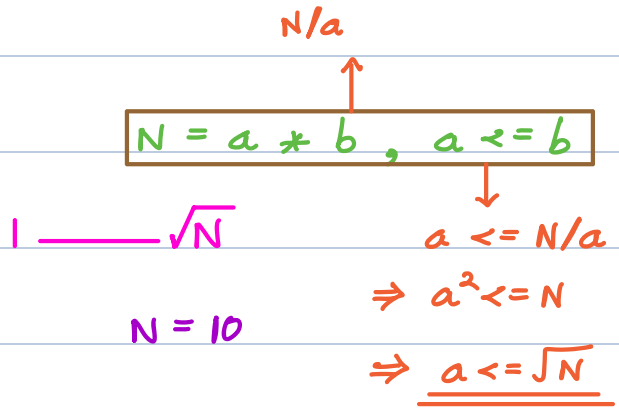
A natural number having 2 factors 1 & itself.

2 3 5 7 11 13 17 . . .

< Question > : Given a number N. Check if it is prime or not.

N = 23 → Ans = true

N = 46 → Ans = false




```
for i → 2 to √N {
    if (N % i == 0)
        return false
}
return true
```

TC = O(√N) SC = O(1)

< Question > : Given an integer N. Check every number from 1 to N if it is a prime number or not.

$1 \le N \le 10^6$

N = 10 [1 2 3 4 5 6 7 8 9 10]

F T T F T F T F F F

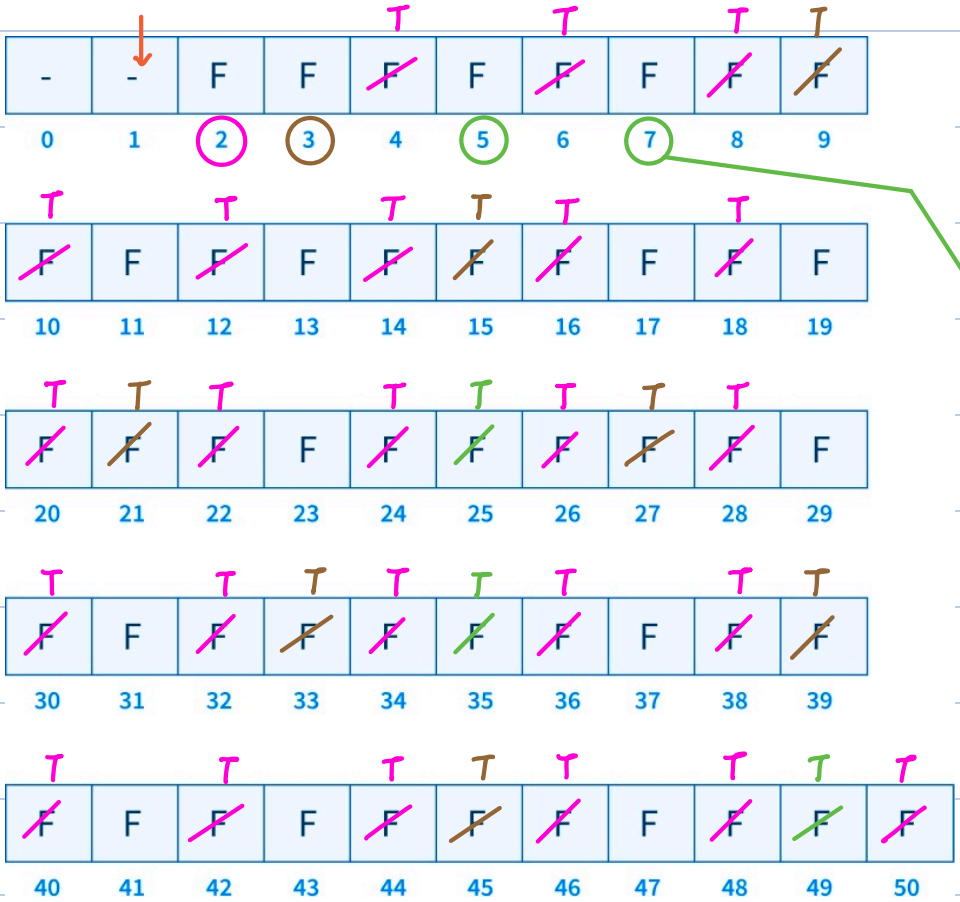
Bruteforce →

```
for i → 1 to N {
    isP[i] = checkPrime(i)
}
```

TC = O(N * √N) SC = O(1)

Sieve of Eratosthenes → chocolates for all except prime no.

class monitor



5 * 2 → 10
5 * 3 → 15
5 * 4 → 20
5 * 5 → 25

7 * 7 = 49

∀ i, isP[i] = true

isP[0] = isP[1] = false

for i → 2 to √N { // for (i = 2; i * i <= N; i++)

    if (isP[i]) {

        for (j = i * i; j <= N; j += i) {

            isP[j] = false

        }

    }

} return isP

SC = O(1) / O(N)

TC

i

2 → 4, 6, . . . N → ~N/2

3 → 9, 12, 15 . . . → ~N/3

4 → ———

5 → 25, 30, 35 . . . → ~N/5

$TC \rightarrow \frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \frac{N}{11} \cdots$

$= N\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} \cdots\right) <= N\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} \cdots\right)$

reciprocal of prime numbers

$\log(\log(N))$

$O(N \log(\log(N)))$

$\sum_{\forall i} \frac{1}{i} \rightarrow \int \frac{1}{x} dx = \log(x)$

$\log(N)$

$O(N \log(N))$

$\log_2(\log_2(2^{\overbrace{2^{32}}^{\sim 10^9}})) = \log_2(32) = 5$

< Question > : Find the count of divisors for every integer from 1 to N.

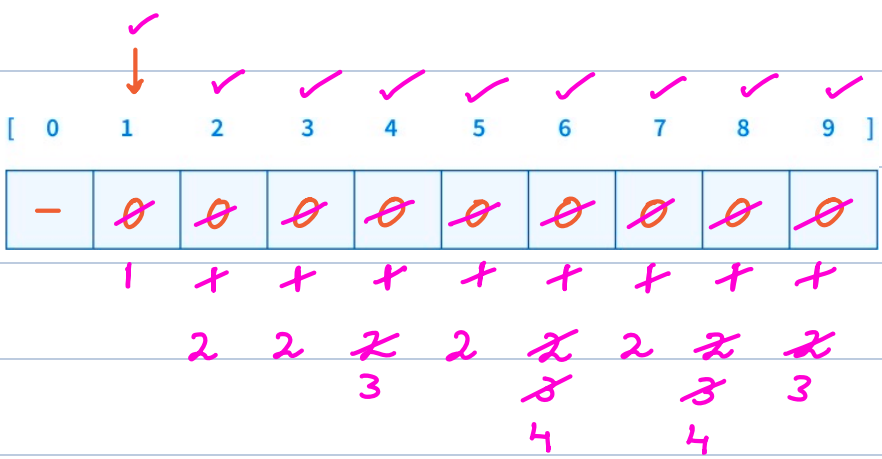
N = ~~10~~ 9

	0	1	2	3	4	5	6	7	8	9
ans[ ] →	–	1	2	2	3	2	4	2	4	3

Bruteforce →

```
for i → 1 to N {
    ans[i] = countDivisors(i)
}
```

TC = O(N √N) SC = O(1)



∀i, ans[i] = 0

```
for i → 1 to N {
    for (j = i; j <= N; j += i) {
        ans[j]++
    }
}
```

TC = O(N log(N)) SC = O(1)

Sorted Permutation Rank

< *Question* > : You are given a string A containing distinct characters (no characters are repeated). The task is to find the rank of this string among all its permutations when sorted in lexicographical (dictionary) order.

s = "cab"

a b c 1
a c b 2
b a c 3
b c a 4
c a b 5 ← Ans
c b a 6

s = "play"

a < l < p < y

a _ _ _ → 3! = 6

l _ _ _ → 3! = 6

p a _ _ → 2! = 2

p l a y → 1

15 (Ans)

s = "date"

a < d < e < t

a _ _ _ → 3! = 6

d a e _ → 1! = 1

d a t e → 1

8 (Ans)

```
fact[0] = fact[1] = 1
for i → 2 to N {
    fact[i] = (fact[i-1] * i) % M
}

rank = 1
for i → 0 to (N-1) {
    ch = s[i]
    crt = 0
    for j → (i+1) to (N-1) {
        if (s[j] < ch) crt++
    }
    rank += crt * fact[N-i-1]
    rank %= M
}
return rank
```

TC = $O(N^2)$ SC = $O(N)$ → fact[]

"utkarsh" (5)

	crt	rank = 1
< 'u' ⇒ 6	6	+ 6*6!
u t k	5	+ 5*5!
	2	+ 2*4!
		⋮