**A1) Basic Maths Concepts & Questions**

**1) Addition of Two Numbers -**

class Solution {

public:

int addition(int a, int b) {

int c = a + b;

return c;

}

};

// -------------------

2) **Prime Number –**

class Solution {

public:

bool isPrime(int n) {

for(int i=2; i\*i<=n; i++)

{

if (n%i == 0)

{

return false;

}

}

return true;

}

};

// -------------------

3) **LCM And GCD**

class Solution {

public:

int gcd(int a, int b) {

if (a == 0)

return b;

return gcd(b % a, a);

}

int lcm(int a, int b) {

return (a \* b) / gcd(a, b);

}

vector<int> lcmAndGcd(int a, int b) {

int g = gcd(a, b);

int l = lcm(a, b);

return {l, g};

}

};

// -------------------

4) **Count Squares**

class Solution {

public:

int countSquares(int n) {

return sqrt(n-1);

}

};

// -------------------

**5) Count Digits**

class Solution {

public:

// Function to count the number of digits in n that evenly divide n

int evenlyDivides(int n) {

int count = 0;

int originalNumber = n;

while(n!=0)

{

int lastdigit = n%10;

if(lastdigit!=0 && originalNumber%lastdigit == 0)

{

count++;

}

n/=10;

}

return count;

}

};

// -------------------

**6) Armstrong Numbers**

class Solution {

public:

bool armstrongNumber(int n) {

int originalNumber = n;

int sum = 0;

while (n != 0) {

int lastDigit = n % 10;

sum += pow(lastDigit, 3); // sum of cubes of digits

n /= 10;

}

return sum == originalNumber; // check after the loop

}

};

// -------------------

**7) Multiplication Table**

#include <vector>

using namespace std;

class Solution {

public:

vector<int> getTable(int n) {

vector<int> table(10);

for (int i = 0; i < 10; i++) {

table[i] = n \* (i + 1);

}

return table;

}

};

// -------------------

**8) Reversing the equation**

class Solution {

public:

string reverseEqn(string s) {

int n = s.length();

reverse(s.begin(), s.end());

int prev = 0;

for(int i = 0; i<n; i++)

{

if(s[i]=='+' || s[i]=='-' || s[i]=='\*'|| s[i] == '/')

{

int j = i-1;

while(prev<j)

{

swap(s[prev],s[j]);

j--;

prev++;

}

prev = i+1;

}

}

int j = n-1;

while(prev<j)

{

swap(s[prev],s[j]);

j--;

prev++;

}

return s;

}

};

// -------------------

**9) All divisors of a Number**

class Solution {

public:

void print\_divisors(int n) {

vector<int> ans;

for(int i=1; i<=sqrt(n); i++)

{

if(n%i==0)

{

ans.push\_back(i);

if(n/i!=i)

{

ans.push\_back(n/i);

}

}

}

sort(ans.begin(), ans.end());

for(int &i : ans)

{

cout<<i<<" ";

}

}

};

// -------------------

**10) Sum of Digit is Palindrome or not**

class Solution {

public:

bool isDigitSumPalindrome(int n) {

int sum = 0, originalSum;

int temp = n;

while(temp != 0) {

sum += temp % 10;

temp /= 10;

}

originalSum = sum;

int reversed = 0;

while(sum != 0) {

reversed = reversed \* 10 + (sum % 10);

sum /= 10;

}

return originalSum == reversed;

}

};

// -------------------

**11)** [**258. Add Digits**](https://leetcode.com/problems/add-digits/)

class Solution {

public:

    int addDigits(int num) {

        while (num >= 10) {

            num = num / 10 + num % 10;

        }

        return num;

    }

};

// -------------------

**12) Sum of all prime numbers between 1 and n**

class Solution{

public:

bool isPrime(int num) {

if(num <= 1) return false;

for(int i = 2; i\*i <= num; i++) {

if(num % i == 0) return false;

}

return true;

}

long long prime\_Sum(int n){

long long sum = 0;

for(int i = 2; i <= n; i++){

if(isPrime(i))

sum += i;

}

return sum;

}

};

// -------------------

**13)** [**2413. Smallest Even Multiple**](https://leetcode.com/problems/smallest-even-multiple/)

class Solution {

public:

    int smallestEvenMultiple(int n) {

      if(n%2 == 0)

      {

        return n;

      }

      return 2\*n;

    }

};

// -------------------

**14)** [**2427. Number of Common Factors**](https://leetcode.com/problems/number-of-common-factors/)

class Solution {

public:

    int commonFactors(int a, int b) {

        int count = 0;

        int m = min(a, b); // no factor can be greater than min(a,b)

        for (int i = 1; i <= m; i++) {

            if (a % i == 0 && b % i == 0) {

                count++;

            }

        }

        return count;

    }

};

// -------------------

**15) Multiply two strings**

// -------------------

**16) Largest prime factor**

class Solution {

public:

long long int largestPrimeFactor(int N)

{

long long int ans = 0;

for (int i = 2; i \* i <= N; i++)

{

while (N % i == 0)

{

if (i > ans)

{

ans = i;

}

N = N / i;

}

}

if (N > 1)

{

if (N > ans)

{

ans = N;

}

}

return ans;

}

};

// -------------------

**17)** [**204. Count Primes**](https://leetcode.com/problems/count-primes/)

#include <bits/stdc++.h>

using namespace std;

class Solution {

public:

    int countPrimes(int n) {

        vector<bool> isPrime(n, true);

        int count = 0;

        for (int i = 2; i < n; i++) {

            if (isPrime[i]) {

                count++;

                for (int j = 2 \* i; j < n; j += i) {

                    isPrime[j] = false;

                }

            }

        }

        return count;

    }

};

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_