<aside> 💡 **Question 1**

Given an integer n, return *true* if it is a power of two. Otherwise, return *false*.

An integer n is a power of two, if there exists an integer x such that n == 2x.

**Example 1:** Input: n = 1

Output: true

**Example 2:** Input: n = 16

Output: true

**Example 3:** Input: n = 3

Output: false

</aside>

Sol: #include <bits/stdc++.h>

using namespace std;

bool powerOfTwo(int n)

{

    while(n % 2 == 0)

        n /= 2;

    return (n == 1);

}

int main()

{

    int n = 16;

    if(powerOfTwo(n))

        cout << "true" << '\n';

    else

        cout << "false" << '\n';

    return 0;

}

Output:True

<aside> 💡 **Question 2**

Given a number n, find the sum of the first natural numbers.

**Example 1:**

Input: n = 3

Output: 6

**Example 2:**

Input : 5

Output : 15

</aside>

Sol: #include <iostream>

using namespace std;

int sumOfFirstN(int n) {

    int sum = (n \* (n + 1)) / 2;

    return sum;

}

int main() {

    int n;

    n=3;

    int sum = sumOfFirstN(n);

    cout << "Sum of the first " << n << " natural numbers is: " << sum << endl;

    return 0;

}

Output:6

<aside> 💡 **Question 3**

\*\*\*\*Given a positive integer, N. Find the factorial of N.

**Example 1:**

Input: N = 5

Output: 120

**Example 2:**

Input: N = 4

Output: 24

</aside>

Sol: #include <iostream>

using namespace std;

int factorial(int N) {

    int fact = 1;

    for (int i = 1; i <= N; i++) {

        fact \*= i;

    }

    return fact;

}

int main() {

    int N;

    N=5;

    int fact = factorial(N);

    cout << "Factorial of " << N << " is: " << fact << endl;

    return 0;

}

Output: Factorial of 5 is: 120

<aside> 💡 **Question 4**

Given a number N and a power P, the task is to find the exponent of this number raised to the given power, i.e. N^P.

**Example 1 :**

Input: N = 5, P = 2

Output: 25

**Example 2 :** Input: N = 2, P = 5

Output: 32

</aside>

Sol: #include <iostream>

#include <cmath>

using namespace std;

int main() {

    int N, P;

   N=5;

   P=2;

    int result = pow(N, P);

    cout << N << " raised to the power " << P << " is: " << result << endl;

    return 0;

}

Output: 5 raised to the power 2 is: 24

<aside> 💡 **Question 5**

Given an array of integers **arr**, the task is to find maximum element of that array using recursion.

**Example 1:**

Input: arr = {1, 4, 3, -5, -4, 8, 6}; Output: 8

**Example 2:**

Input: arr = {1, 4, 45, 6, 10, -8}; Output: 45

</aside>

Sol: #include <iostream>

#include <algorithm>

using namespace std;

int findMax(int arr[], int start, int end) {

    if (start == end) {

        return arr[start];

    }

    int mid = (start + end) / 2;

    int leftMax = findMax(arr, start, mid);

    int rightMax = findMax(arr, mid + 1, end);

    return max(leftMax, rightMax);

}

int main() {

    int arr[] = {1, 4, 3, -5, -4, 8, 6};

    int size = sizeof(arr) / sizeof(arr[0]);

    int maxElement = findMax(arr, 0, size - 1);

    cout << "Maximum element: " << maxElement << endl;

    return 0;

}

Output: Maximum element: 8

<aside> 💡 **Question 6**

Given first term (a), common difference (d) and a integer N of the Arithmetic Progression series, the task is to find Nth term of the series.

**Example 1:**

Input : a = 2 d = 1 N = 5 Output : 6 The 5th term of the series is : 6

**Example 2:**

Input : a = 5 d = 2 N = 10 Output : 23 The 10th term of the series is : 23

</aside>

Sol: #include <iostream>

using namespace std;

int findNthTerm(int a, int d, int N) {

    int Nth\_term = a + (N - 1) \* d;

    return Nth\_term;

}

int main() {

    int a, d, N;

    a=2;

    d=1;

    N=5;

    int Nth\_term = findNthTerm(a, d, N);

    cout << "The " << N << "th term of the series is: " << Nth\_term << endl;

    return 0;

}

Output: The 5th term of the series is: 6

<aside> 💡 **Question 7**

Given a string S, the task is to write a program to print all permutations of a given string.

**Example 1:**

***Input:***

S = “ABC”

***Output:***

“ABC”, “ACB”, “BAC”, “BCA”, “CBA”, “CAB”

**Example 2:**

***Input:***

S = “XY”

***Output:***

“XY”, “YX”

</aside>

Sol:

#include <bits/stdc++.h>

using namespace std;

void permute(string& a, int l, int r)

{

    if (l == r)

        cout << a << endl;

    else {

        for (int i = l; i <= r; i++) {

            swap(a[l], a[i]);

            permute(a, l + 1, r);

            swap(a[l], a[i]);

        }

    }

}

int main()

{

    string str = "ABC";

    int n = str.size();

    permute(str, 0, n - 1);

    return 0;

}

Output: ABC

ACB

BAC

BCA

CBA

CAB

<aside> 💡 **Question 8**

Given an array, find a product of all array elements.

**Example 1:**

Input : arr[] = {1, 2, 3, 4, 5} Output : 120 **Example 2:**

Input : arr[] = {1, 6, 3} Output : 18

</aside>

Sol: #include <iostream>

using namespace std;

long long findProduct(int arr[], int size) {

    long long product = 1;

    for (int i = 0; i < size; i++) {

        product \*= arr[i];

    }

    return product;

}

int main() {

    int arr[] = {1, 2, 3, 4, 5};

    int size = sizeof(arr) / sizeof(arr[0]);

    long long product = findProduct(arr, size);

    cout << "Product of array elements: " << product << endl;

    return 0;

}

Product of array elements: 120