Question No: 1

Single File Programming Question

**Lowest Common Multiple (LCM)**

You are given an array of positive integers, arr, of size array\_length. You are asked to build set S which consists of the LCM of every pair of adjacent elements in arr. Your task is to find the largest element in set S.

For example, for the array {1, 2, 3, 4}, set S = { lcm(1,2), lcm(2,3), lcm(3,4) } = {2, 6, 12}. The largest element is 12.

**Notes**

The array contains only positive integers.

Adjacent elements may NOT be circular, i.e. they may not wrap around the end of the array

**Definition of LCM**

A multiple of a number, num, is a number that can be divided by num. For example, the multiples of 5 are 5, 10, 15, 20, 25, and so on. The lowest common multiple (LCM) of two numbers, a and b, is the smallest positive number that is a multiple of both a and b.

**Example 1**

**Input**

4

1 3 2 4

**Output**

6

**Explanation**

set S = {lcm(1, 3), lcm(3, 2), lcm(2, 4) } = {3, 6, 4}

Largest element = 6

**Example 2**

**Input**

5

7 3 2 9 12

**Output**

36

**Explanation**

set S = { lcm(7, 3), lcm(3, 2), lcm(2, 9), lcm(9, 12) } = {21, 6, 18, 36}

Largest element = 36

Input format

The first line of input denotes the size of an array.

The next lines contain the space-separated integers denoting the value of the array.

Output format

The output contains an integer denoting the largest element in set S as specified in the problem statement.

Code constraints

4 <= array\_length <= 105

1 <= array[i] <= 1500

Sample testcases

Input 1

4

1 3 2 4

Output 1

6

Input 2

5

7 3 2 9 12

**C PROGRAMMING :**

**#include<stdio.h>**

**int gcd(int a, int b)**

**{**

**if (b == 0)**

**return a;**

**return gcd(b, a % b);**

**}**

**int lcm(int arr[], int n)**

**{**

**int lm, large = -9999;**

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

**{**

**lm = (arr[i] \* arr[i-1]) / gcd(arr[i], arr[i-1]);**

**if(lm > large)**

**large = lm;**

**}**

**return large;**

**}**

**int main()**

**{**

**int n, i;**

**scanf("%d", &n);**

**int arr[n];**

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

**scanf("%d", &arr[i]);**

**printf("%d", lcm(arr,n));**

**return 0;**

**}**

**C++ PROGRAMMING :**

**#include<iostream>**

**using namespace std;**

**int gcd(int a, int b)**

**{**

**if (b == 0)**

**return a;**

**return gcd(b, a % b);**

**}**

**int lcm(int arr[], int n)**

**{**

**int lm, large = -9999;**

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

**{**

**lm = (arr[i] \* arr[i-1]) / gcd(arr[i], arr[i-1]);**

**if(lm > large)**

**large = lm;**

**}**

**return large;**

**}**

**int main()**

**{**

**int n, i;**

**cin >> n;**

**int arr[n];**

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

**cin >> arr[i];**

**cout << lcm(arr, n);**

**return 0;**

**}**

**JAVA PROGRAMMING :**

**import java.util.\*;**

**class Main**

**{**

**static int gcd(int a, int b)**

**{**

**if (b == 0)**

**return a;**

**return gcd(b, a % b);**

**}**

**static int lcm(int arr[], int n)**

**{**

**int lm, large = -9999;**

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

**{**

**lm = (arr[i] \* arr[i-1]) / gcd(arr[i], arr[i-1]);**

**if(lm > large)**

**large = lm;**

**}**

**return large;**

**}**

**public static void main(String args[])**

**{**

**Scanner sc = new Scanner(System.in);**

**int n = sc.nextInt();**

**int arr[] = new int[n];**

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

**arr[i] = sc.nextInt();**

**System.out.print(lcm(arr, n));**

**}**

**}**

**PYTHON :**

**def gcd(a,b):**

**if (b == 0):**

**return a**

**return gcd(b, a%b)**

**def lcm(arr, n):**

**large = -9999**

**for i in range(1, n):**

**lm = (arr[i] \* arr[i-1]) // gcd(arr[i], arr[i-1])**

**if lm > large:**

**large = lm**

**return large**

**n = int(input())**

**arr = list(map(int, input().split()))**

**print(lcm(arr, n))**

Question No: 2

Single File Programming Question

Given a rod of length N inches and an array of prices that contains prices of all pieces of a size smaller than N. Determine the maximum value obtainable by cutting up the rod and selling the pieces.

**Example :**

length  | 1  2  3  4  5  6  7  8

--------------------------------------------

price  | 1  5  8  9 10 17 17 20

The maximum obtainable value is 24(by cutting in eight pieces of length 1).

Input format

The first element of integer N denoting the length of the rod.after that array of the price.

Output format

Print the maximum selling pieces.

Code constraints

Integers only.

Sample testcases

Input 1

8 3 5 8 9 10 17 17 20

Output 1

24

Input 2

9 2 4 6 7 8 12 14 15 17

Output 2

18

**C PROGRAMMING :**

**#include<bits/stdc++.h>**

**using namespace std;**

**int max(int a, int b) { return (a > b)? a : b;}**

**int cutRod(int price[], int n)**

**{**

**int val[n+1];**

**val[0] = 0;**

**int i, j;**

**for (i = 1; i<=n; i++)**

**{**

**int max\_val = INT\_MIN;**

**for (j = 0; j < i; j++)**

**max\_val = max(max\_val, price[j] + val[i-j-1]);**

**val[i] = max\_val;**

**}**

**return val[n];**

**}**

**int main()**

**{ int n,i,k=1;**

**scanf("%d",&n);**

**int a[n];**

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

**scanf("%d",&a[i]);**

**printf("%d", cutRod(a, n));**

**return 0;**

**}**