|  |  |
| --- | --- |
| **Chocolate Distribution in C++** | |
| #include <iostream>  #include <algorithm>  #include <vector>  #include <climits>  using namespace std;  class ChocolateDistribution {  public:  static int find(vector<int>& arr, int n, int m) {  // Sort the array of weights  sort(arr.begin(), arr.end());  int minDifference = INT\_MAX;  // Find the minimum difference between maximum and minimum weights in subarrays of size m  for (int i = 0; i <= n - m; ++i) {  int minWeight = arr[i];  int maxWeight = arr[i + m - 1];  int difference = maxWeight - minWeight;    if (difference < minDifference) {  minDifference = difference;  }  }  return minDifference;  }  };  int main() {  // Hardcoded input  int n = 8;  vector<int> arr = {3, 4, 1, 9, 56, 7, 9, 12};  int m = 5;  // Call the find method to get the minimum difference  int ans = ChocolateDistribution::find(arr, n, m);  // Print the result  cout << ans << endl;  return 0;  } | ****Inputs:**** arr = {3, 4, 1, 9, 56, 7, 9, 12}  n = 8  m = 5 ****Step 1: Sort the array**** Sorted arr = {1, 3, 4, 7, 9, 9, 12, 56} ****Step 2: Sliding window of size**** m = 5 We'll check all subarrays of length m = 5 and calculate max - min.   | **i** | **Subarray** | **Min (arr[i])** | **Max (arr[i + m - 1])** | **Difference** | | --- | --- | --- | --- | --- | | 0 | {1, 3, 4, 7, 9} | 1 | 9 | 8 | | 1 | {3, 4, 7, 9, 9} | 3 | 9 | 6 | | 2 | {4, 7, 9, 9, 12} | 4 | 12 | 8 | | 3 | {7, 9, 9, 12, 56} | 7 | 56 | 49 |  ✅ Minimum Difference: From the table above, the **minimum difference** is 6 (from subarray {3, 4, 7, 9, 9}). ****🖨️ Final Output:**** 6 |
| 6 | |