|  |  |
| --- | --- |
| **Max Sum Increasing subseq In C++** | |
| #include <iostream>  #include <climits>  using namespace std;  int MaxSumIncreasingSubseq(int arr[], int size) {      int omax = INT\_MIN;      int\* dp = new int[size];      //int dp[size];      for (int i = 0; i < size; i++) {          int maxSum = arr[i];          for (int j = 0; j < i; j++) {              if (arr[j] <= arr[i]) {                  maxSum = max(maxSum, dp[j] + arr[i]);              }          }          dp[i] = maxSum;          omax = max(omax, dp[i]);      }      delete[] dp; // Don't forget to free the allocated memory      return omax;  }  int main() {      int arr[] = {10, 22, 9, 33, 21, 50, 41, 60, 80, 3};      int size = sizeof(arr) / sizeof(arr[0]);      int maxSum = MaxSumIncreasingSubseq(arr, size);      cout << maxSum << endl;      return 0;  } | arr = {10, 22, 9, 33, 21, 50, 41, 60, 80, 3}  **Step-by-Step Dry Run (Table Format)**   | **Index (i)** | **arr[i]** | **Initial dp[i]** | **Comparisons (j < i, arr[j] ≤ arr[i])** | **Updated dp[i]** | | --- | --- | --- | --- | --- | | **0** | 10 | 10 | - | **10** | | **1** | 22 | 22 | j=0 (10 ≤ 22) → dp[1] = max(22, 10+22) | **32** | | **2** | 9 | 9 | - | **9** | | **3** | 33 | 33 | j=0 (10 ≤ 33) → dp[3] = max(33, 10+33) j=1 (22 ≤ 33) → dp[3] = max(43, 32+33) | **65** | | **4** | 21 | 21 | j=0 (10 ≤ 21) → dp[4] = max(21, 10+21) | **31** | | **5** | 50 | 50 | j=0 (10 ≤ 50) → dp[5] = max(50, 10+50) j=1 (22 ≤ 50) → dp[5] = max(60, 32+50) j=3 (33 ≤ 50) → dp[5] = max(100, 65+50) | **100** | | **6** | 41 | 41 | j=0 (10 ≤ 41) → dp[6] = max(41, 10+41) j=1 (22 ≤ 41) → dp[6] = max(51, 32+41) j=3 (33 ≤ 41) → dp[6] = max(91, 65+41) | **91** | | **7** | 60 | 60 | j=0 (10 ≤ 60) → dp[7] = max(60, 10+60) j=1 (22 ≤ 60) → dp[7] = max(70, 32+60) j=3 (33 ≤ 60) → dp[7] = max(110, 65+60) j=5 (50 ≤ 60) → dp[7] = max(150, 100+60) | **150** | | **8** | 80 | 80 | j=0,1,3,5,6,7 (comparing all increasing values) → dp[8] = max(10+80, 32+80, 65+80, 100+80, 91+80, 150+80) | **255** | | **9** | 3 | 3 | - | **3** |   **Final DP Table**   | **Index (i)** | **arr[i]** | **dp[i] (Max Sum IS Ending at i)** | | --- | --- | --- | | **0** | 10 | 10 | | **1** | 22 | 32 | | **2** | 9 | 9 | | **3** | 33 | 65 | | **4** | 21 | 31 | | **5** | 50 | 100 | | **6** | 41 | 91 | | **7** | 60 | 150 | | **8** | 80 | 255 | | **9** | 3 | 3 |   **Final Answer**  Output: 255  **Summary:**   * The **largest increasing subsequence** contributing to **255** is:   10 → 22 → 33 → 50 → 60 → 80   * + Sum = 10 + 22 + 33 + 50 + 60 + 80 = 255 |
| Output:- 255  {10, 22, 33, 50, 60, 80} → sum = 10 + 22 + 33 + 50 + 60 + 80 = 255 | |