|  |  |
| --- | --- |
| **Smaller no on left in C++** | |
| #include <iostream>  #include <vector>  #include <stack>  using namespace std;  vector<int> leftSmaller(int n, int a[]) {  vector<int> ans(n);  stack<int> st;  for (int i = n - 1; i >= 0; i--) {  while (!st.empty() && a[i] < a[st.top()]) {  int idx = st.top();  ans[idx] = a[i];  st.pop();  }  st.push(i);  }  while (!st.empty()) {  int idx = st.top();  ans[idx] = -1;  st.pop();  }  return ans;  }  int main() {  int arr[] = {4, 8, 5, 2, 25};  int n = sizeof(arr) / sizeof(arr[0]);  vector<int> result = leftSmaller(n, arr);  cout << "Resulting list:" << endl;  for (int i : result) {  cout << i << " ";  }  cout << endl;  return 0;  } | Input: arr = {4, 8, 5, 2, 25} 📋 Dry Run Table:  | **i** | **arr[i]** | **Stack (index)** | **Action** | **ans (after step)** | | --- | --- | --- | --- | --- | | 4 | 25 | [] | Stack empty, push 4 | [?, ?, ?, ?, ?] | | 3 | 2 | [4] | 2 < 25 → ans[4] = 2, pop 4; push 3 | [?, ?, ?, ?, 2] | | 2 | 5 | [3] | 5 > 2 → push 2 | [?, ?, ?, ?, 2] | | 1 | 8 | [3, 2] | 8 > 5 → push 1 | [?, ?, ?, ?, 2] | | 0 | 4 | [3, 2, 1] | 4 < 8 → ans[1] = 4, pop 1; 4 < 5 → ans[2] = 4, pop 2; push 0 | [?, 4, 4, ?, 2] | |  |  | [3, 0] | Final elements → set ans[3] = -1, ans[0] = -1 | [-1, 4, 4, -1, 2] |  ✅ Final Output: -1 4 4 -1 2 ✅ Explanation (Index-wise):  | **Index** | **arr[i]** | **Left Smaller Element** | | --- | --- | --- | | 0 | 4 | -1 (nothing to the left) | | 1 | 8 | 4 | | 2 | 5 | 4 | | 3 | 2 | -1 | | 4 | 25 | 2 | |
| Resulting list:  -1 4 4 -1 2 | |