|  |  |
| --- | --- |
| **Binary Recursive in C++** | |
| #include <iostream>  #include <vector>  using namespace std;  int binsearch(const vector<int>& arr, int low, int high, int x) {  if (low > high) {  return -1;  }  int mid = (low + high) / 2;  if (arr[mid] == x) {  return mid;  } else if (arr[mid] > x) {  return binsearch(arr, low, mid - 1, x);  } else {  return binsearch(arr, mid + 1, high, x);  }  }  int main() {  vector<int> arr = {3, 5, 7, 8, 9, 11, 45, 76};  int result = binsearch(arr, 0, arr.size() - 1, 11);  cout << result << endl;  return 0;  } | Here's a **tabular dry run** of the **recursive binary search** code for:  arr = {3, 5, 7, 8, 9, 11, 45, 76}  x = 11  **🧮 Dry Run Table**   | **Call #** | **low** | **high** | **mid = (low+high)/2** | **arr[mid]** | **Comparison** | **Action** | | --- | --- | --- | --- | --- | --- | --- | | 1 | 0 | 7 | (0+7)/2 = 3 | 8 | 8 < 11 | Search right → low = mid+1 = 4 | | 2 | 4 | 7 | (4+7)/2 = 5 | 11 | 11 == 11 | **Found** → return 5 |   **✅ Output**  5 |
| 5 | |