```
#include <iostream> // --> Includes input-output stream for using cin and cout
#include <cstdlib> // --> Includes standard library for dynamic memory allocation
#include <omp.h> // --> Includes OpenMP library for parallel processing
using namespace std; // --> Uses standard namespace to avoid std:: prefix

void bubble(int *, int); // --> Declares the bubble sort function
void swap(int &, int &); // --> Declares the swap function

void bubble(int *a, int n) // --> Defines the bubble sort function with array and size as parameters
{
  for (int i = 0; i < n; i++) // --> Outer loop for each pass of bubble sort
  {
    int first = i % 2; // --> Determines if pass is even or odd for odd-even sort

#pragma omp parallel for shared(a, first) // --> OpenMP directive to parallelize the inner loop
    for (int j = first; j < n - 1; j += 2) // --> Inner loop for comparing and swapping adjacent elements
    {
      if (a[j] > a[j + 1]) // --> If current element is greater than next element
      {
        swap(a[j], a[j + 1]); // --> Swap the two elements
      }
    }
  }
}

void swap(int &a, int &b) // --> Defines swap function with reference parameters
{
  int temp; // --> Temporary variable to hold value during swap
  temp = a; // --> Store value of a in temp
  a = b; // --> Assign value of b to a
```

```
    b = temp; // --> Assign value of temp to b
}

int main() // --> Main function starts program execution
{
    cout << "\n\nName: Shriharsh Deshmukh\nRoll No.62 \t Div.A\n\n"; // --> Prints student info
    int *a, n; // --> Declares pointer for array and variable for size
    cout << "\nEnter total number of elements: "; // --> Prompts user to enter number of elements
    cin >> n; // --> Reads number of elements
    a = new int[n]; // --> Dynamically allocates memory for array
    cout << "\nEnter elements: "; // --> Prompts user to enter array elements
    for (int i = 0; i < n; i++) // --> Loop to read array elements
    {
        cin >> a[i]; // --> Reads individual element into array
    }

    bubble(a, n); // --> Calls bubble sort function on array

    cout << "\nSorted array is:\n"; // --> Prints sorted array label
    for (int i = 0; i < n; i++) // --> Loop to print sorted array
    {
        cout << a[i] << " "; // --> Prints individual array element
    }
    cout << endl; // --> Prints newline after array

    delete[] a; // --> Deallocates dynamically allocated memory
    return 0; // --> Returns 0 to indicate successful execution
}

// Run Commands:
// g++ -fopenmp -o bubble_sort 3_bubble_sort.cpp // --> Compile with OpenMP support
```

```
// .\bubble_sort // --> Run the compiled program
```