C++ Assignments | Selection and insertion sorting | Week 9

1. Which of the following is true about selection sort:
a) In each iteration we find the minimum element in the unsorted part of the array.
b) In each iteration we find the index of the minimum element in the unsorted part of the array.
c) We swap the index of the minimum element with the first element of the array.
d) It takes O(n^2) swaps.
—-the correct answer is a.
2.Which of the following examples represent the worst case input for an insertion sort?
a) array in sorted order
b) large array
c) normal unsorted array
d) array sorted in reverse order

Ans :D array sorted in reverse order.
3.How many passes would be required during insertion sort to sort an array of 5 elements?
a) 1
b) b) Depends on order of elements
c) c) 4
d) d) 5
Ans :-d-5
4.Given an array of digits (values are from 0 to 9), the task is to find the minimum possible sum of two numbers formed from digits of the array. Please note that all digits of the given array must be used to form the two numbers.

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int minimumSumOfTwoNumbers(vector<int>& digits) {
    // Step 1: Sort the array
    sort(digits.begin(), digits.end());

    // Step 2: Form two numbers
    int num1 = 0, num2 = 0;
    for (int i = 0; i < digits.size(); ++i) {
        if (i % 2 == 0) {
            num1 = num1 * 10 + digits[i];
        } else {
            num2 = num2 * 10 + digits[i];
        }
    }

    // Step 3: Calculate the sum
    int sum = num1 + num2;
    return sum;
}

int main() {
    vector<int> digits = {6, 8, 4, 5, 2, 3};
    int minSum = minimumSumOfTwoNumbers(digits);
```

```
    cout << "Minimum possible sum of two numbers: " << minSum << endl;
    return 0;
}
```

5.Given an array of strings arr[] with all strings in lowercase. Sort given strings using Bubble Sort and display the sorted array.

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

void bubbleSort(vector<string>& arr) {
    int n = arr.size();
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            // Compare adjacent strings and swap if out of order
            if (arr[j] > arr[j + 1]) {
                swap(arr[j], arr[j + 1]);
            }
        }
    }
}

int main() {
    vector<string> arr = {"apple", "orange", "banana", "grape", "kiwi"};

    cout << "Array before sorting:" << endl;
    for (const string& s : arr) {
        cout << s << " ";
    }
    cout << endl;

    // Sort using Bubble Sort
    bubbleSort(arr);

    cout << "Array after Bubble Sort:" << endl;
    for (const string& s : arr) {
        cout << s << " ";
    }
    cout << endl;

    return 0;
}
```