Name - Srushti Bhivaji Salgar
PRN - B24CE1079
Class - SE 2   Batch - A
Subject - Data Structures

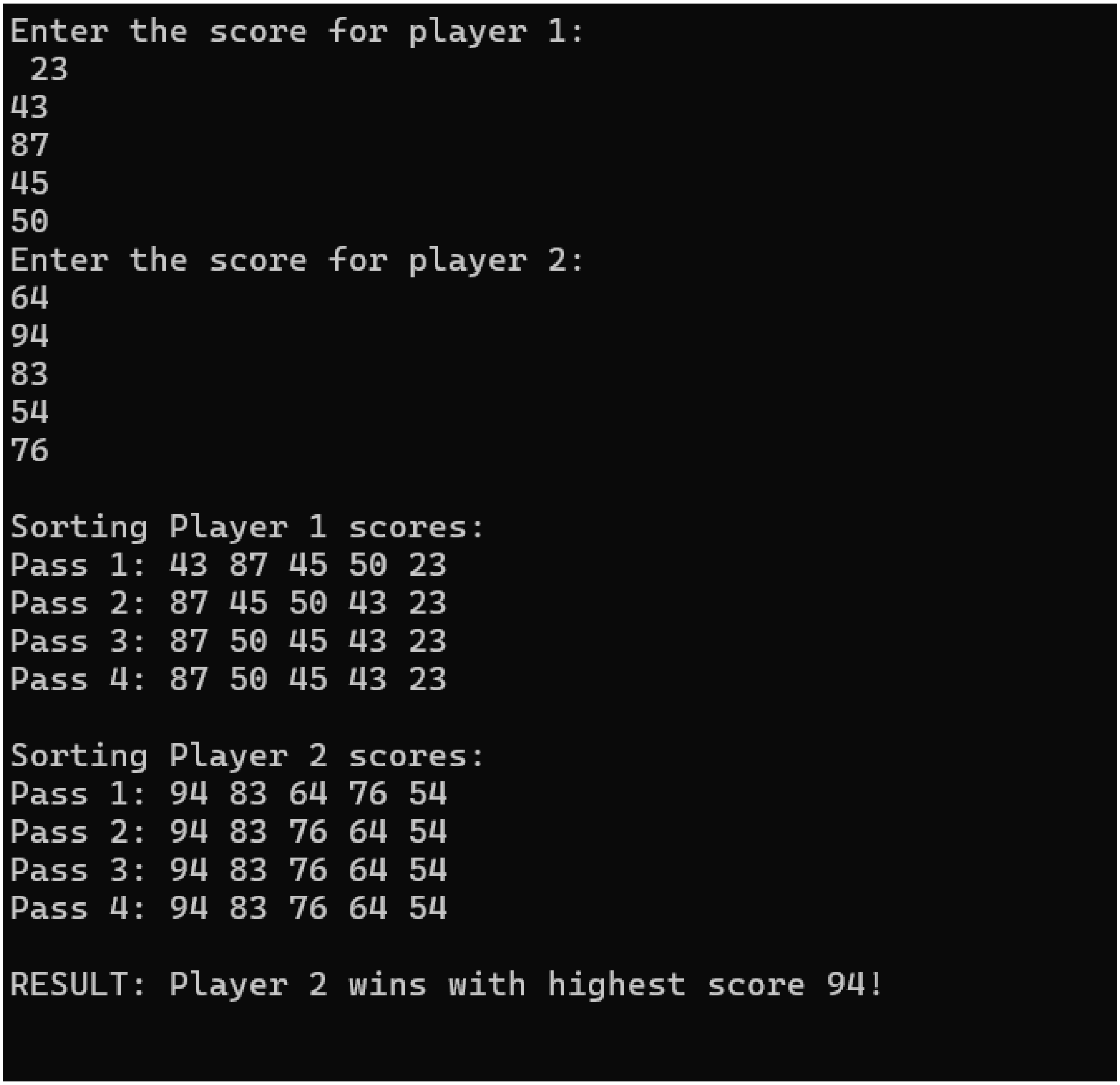
Assignment 2

/*Game Development:
write a game development program that implements the Bubble Sort algorithm.
The program will simulate a simple game where the player can input a set of numbers, and the numbers will be sorted using Bubble Sort
to simulate a "level-up" scenario where the player's scores are sorted in ascending order.*/

CODE

```
#include<iostream>
using namespace std;

int main(){
    int n = 5;
    int player1[n];
    int player2[n];
    int temp = 0;

    // Input for player 1
    cout << "Enter the score for player 1:\n ";
    for(int i = 0; i < n; i++){
        cin >> player1[i];
    }

    // Input for player 2
     cout << "Enter the score for player 2:\n";
    for(int i = 0; i < n; i++){
        cin >> player2[i];
    }

    // Bubble Sort (Descending) for player 1 with passes
    cout << "\nSorting Player 1 scores:\n";
    for(int i = 0; i < n-1; i++){
        for(int j = 0; j < n-i-1; j++){
            if(player1[j] < player1[j+1]){   // just reverse condition
                temp = player1[j];
                player1[j] = player1[j+1];
                player1[j+1] = temp;
            }
        }
        cout << "Pass " << i+1 << ": ";
        for(int k = 0; k < n; k++){
            cout << player1[k] << " ";
        }
        cout << endl;
    }

    // Bubble Sort (Descending) for player 2 with passes
    cout << "\nSorting Player 2 scores:\n";
    for(int i = 0; i < n-1; i++){
        for(int j = 0; j < n-i-1; j++){
            if(player2[j] < player2[j+1]){   // reverse condition
                temp = player2[j];
                player2[j] = player2[j+1];
                player2[j+1] = temp;
            }
        }
        cout << "Pass " << i+1 << ": ";
        for(int k = 0; k < n; k++){
            cout << player2[k] << " ";
        }
        cout << endl;
    }

    // Winner Announcement
    cout << "\nRESULT: ";
    if(player1[0] > player2[0]){
        cout << "Player 1 wins with highest score " << player1[0] << "!\n";
    } else if(player2[0] > player1[0]){
        cout << "Player 2 wins with highest score " << player2[0] << "!\n";
    } else {
        cout << "It's a tie! Both have highest score " << player1[0] << "!\n";
    }

    return 0;
}
```

OUTPUT



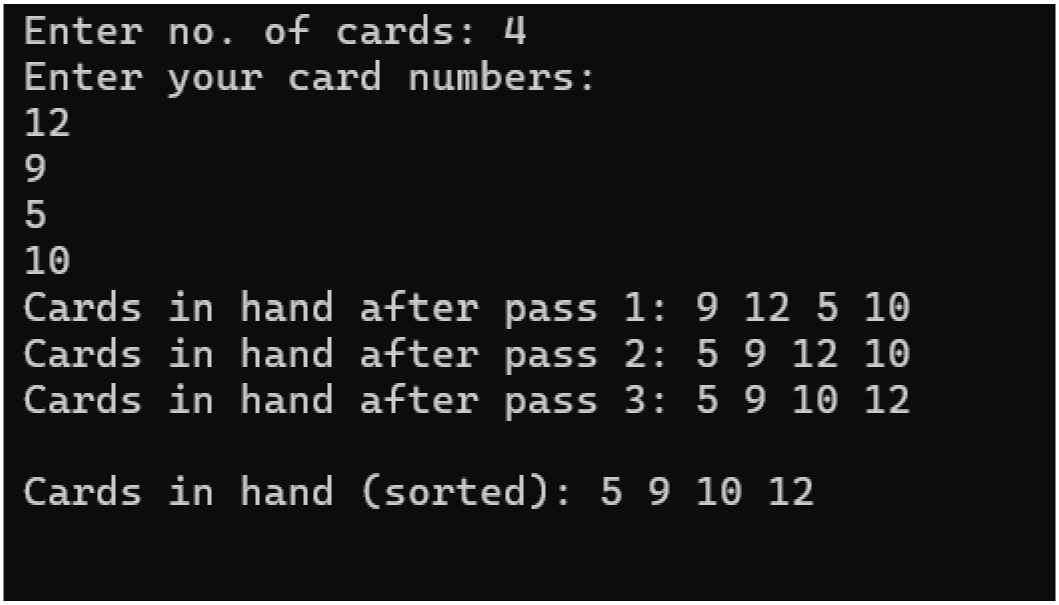
```
Enter the score for player 1:
 23
43
87
45
50
Enter the score for player 2:
64
94
83
54
76

Sorting Player 1 scores:
Pass 1: 43 87 45 50 23
Pass 2: 87 45 50 43 23
Pass 3: 87 50 45 43 23
Pass 4: 87 50 45 43 23

Sorting Player 2 scores:
Pass 1: 94 83 64 76 54
Pass 2: 94 83 76 64 54
Pass 3: 94 83 76 64 54
Pass 4: 94 83 76 64 54

RESULT: Player 2 wins with highest score 94!
```

/*Organizing Cards in a Hand:
Application: When playing card games, players often use an approach similar to insertion sort to organize their cards.
They pick one card at a time and insert it into the correct position in their hand, maintaining a sorted sequence.
Write a program that demonstrates how to organize (sort) cards in a hand using insertion sort*/

CODE

```
#include <iostream>
using namespace std;

int main() {
    int n, temp;
    cout << "Enter no. of cards: ";
    cin >> n;
    int arr[n];

    // Accept card numbers
    cout << "Enter your card numbers:\n";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    // Insertion Sort
    for (int p = 0; p < n - 1; p++) {
        int i = p + 1;   // index of unsorted card
        int j = i - 1;   // index of sorted part
        temp = arr[i];   // card to be placed

        while (j >= 0 && temp < arr[j]) {
            arr[j + 1] = arr[j]; // shift larger cards to right
            j--;
        }
        arr[j + 1] = temp; // place card in correct position

        // Print cards in hand after each pass
        cout << "Cards in hand after pass " << p + 1 << ": ";
        for (int k = 0; k < n; k++) {
            cout << arr[k] << " ";
        }
        cout << endl;
    }
    // Final sorted hand
    cout << "\nCards in hand (sorted): ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;

    return 0;
}
```

OUTPUT



```
Enter no. of cards: 4
Enter your card numbers:
12
9
5
10
Cards in hand after pass 1: 9 12 5 10
Cards in hand after pass 2: 5 9 12 10
Cards in hand after pass 3: 5 9 10 12

Cards in hand (sorted): 5 9 10 12
```