**DS Assignment 2**

**NAME:**TruptiLohakare

**PRN:**B24CE1065

/\*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.

OR

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

\*/

**PROGRAM:**

**//BUBBLE SORT**

#include <iostream>

using namespace std;

int main() {

int m, i, j, temp;

cout << "Enter number of scores: ";

cin >> m;

int p1[m], p2[m];

cout << "Enter scores of Player 1:\n";

for (i = 0; i < m; i++) {

cout << "Score " << i + 1 << ": ";

cin >> p1[i];

}

cout << "Enter scores of Player 2:\n";

for (i = 0; i < m; i++) {

cout << "Score " << i + 1 << ": ";

cin >> p2[i];

}

// Bubble Sort for Player 1

for (i = 0; i < m; i++) {

for (j = 0; j < m - i - 1; j++) {

if (p1[j] > p1[j + 1]) {

temp = p1[j];

p1[j] = p1[j + 1];

p1[j + 1] = temp;

}

}

}

cout << "\nSorted scores of Player 1: ";

for (i = 0; i < m; i++) {

cout << p1[i] << " ";

}

cout << endl;

// Bubble Sort for Player 2

for (i = 0; i < m; i++) {

for (j = 0; j < m - i - 1; j++) {

if (p2[j] > p2[j + 1]) {

temp = p2[j];

p2[j] = p2[j + 1];

p2[j + 1] = temp;

}

}

}

cout << "Sorted scores of Player 2: ";

for (i = 0; i < m; i++) {

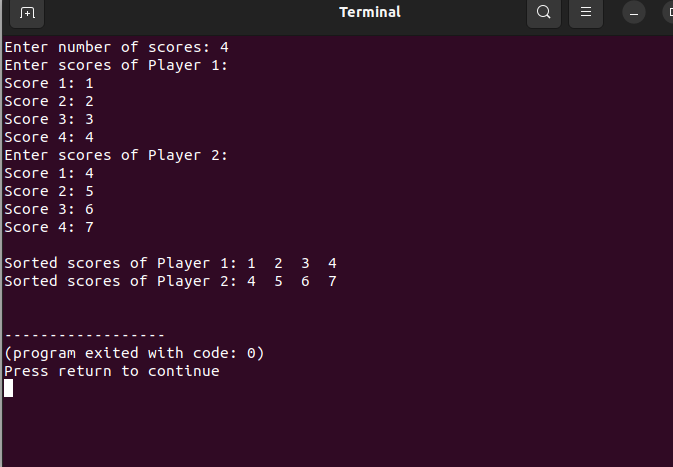
cout << p2[i] << " ";

}

cout << endl;

return 0;

}



**//INSERCTION SORT**

#include <iostream>

using namespace std;

int main() {

int n = 5, temp;

int arr[n];

cout << "Enter the card numbers:\n";

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

// Insertion sort algorithm

for (int i = 1; i < n; i++) {

temp = arr[i];

int j = i - 1;

// Shift elements greater than temp to the right

while (j >= 0 && arr[j] > temp) {

arr[j + 1] = arr[j];

j--;

}

arr[j + 1] = temp; // Insert temp at correct position

}

cout << "Sorted card numbers:\n";

for (int i = 0; i < n; i++) {

cout << arr[i] << " ";

}

cout << endl;

return 0;

}

