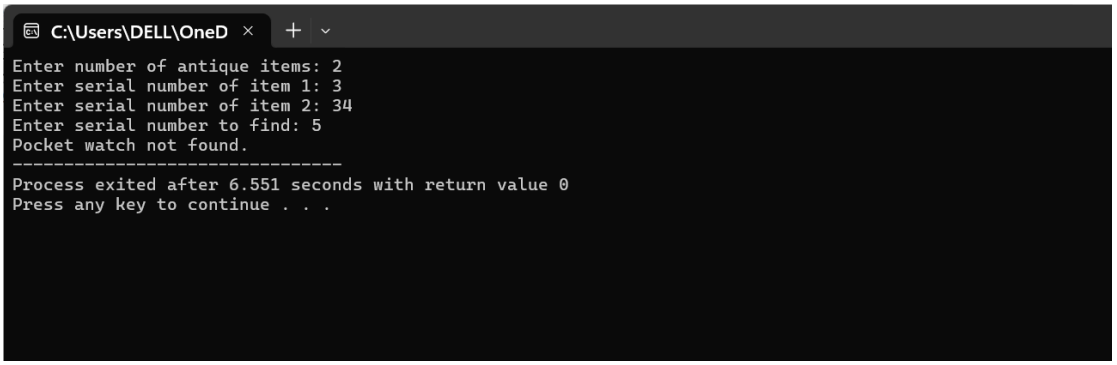
NAME:MOHSIN MUNIR **INSTRUCTOR:SIR SHARJEEL**
SEC: B **ROLL NO: FA24 BSET 077**

ASSIGNMENT LAB 04

Q) In a large inventory of antique items, you are tasked with finding a specific vintage pocket watch based on its serial number. You must implement a sequential search algorithm to locate the watch within the collection of diverse antique items.

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

int main() {
   int n;
   cout << "Enter number of antique items: ";
   cin >> n;

   int serial[100];
   for (int i = 0; i < n; i++) {
      cout << "Enter serial number of item " << i + 1 << ": ";
      cin >> serial[i];
   }

   int search;
   cout << "Enter serial number to find: ";
   cin >> search;

   int found = -1;
   for (int i = 0; i < n; i++) {
      if (serial[i] == search) {
         found = i;
         break;
      }
   }

   if (found != -1)
      cout << "Pocket watch found at position " << found + 1;
   else
      cout << "Pocket watch not found.";
}
```

OUTPUT:



NAME:MOHSIN MUNIR **INSTRUCTOR:SIR SHARJEEL**
SEC: B **ROLL NO: FA24 BSET 077**

ASSIGNMENT LAB 04

NAME:MOHSIN MUNIR **INSTRUCTOR:SIR SHARJEEL**

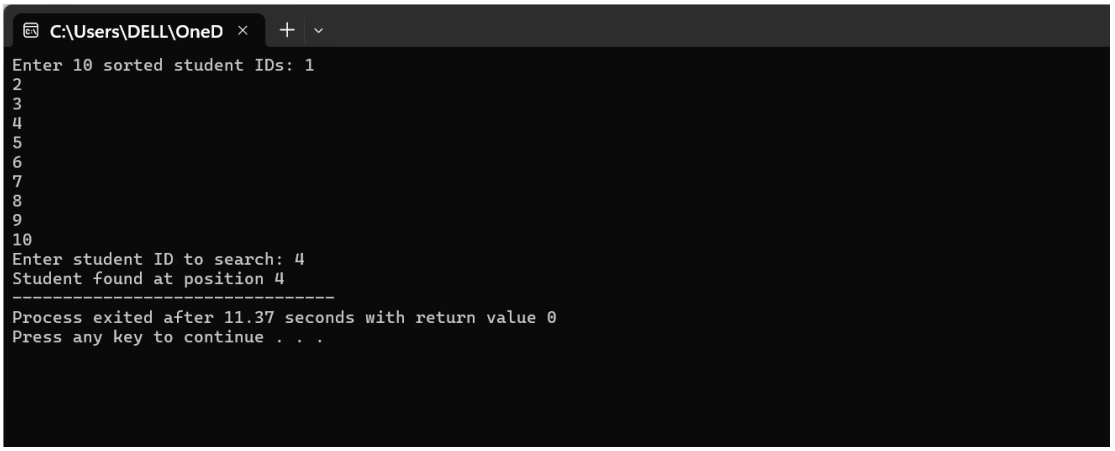
SEC: B **ROLL NO: FA24 BSET 077**

ASSIGNMENT LAB 04

Q) In a sorted list of 10 student records by student ID, you need to find the details of a particular student efficiently. Implement a suitable search code to locate the student's ID using their unique identification number.

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

int main() {
   int id[10];
   cout << "Enter 10 sorted student IDs: ";
   for (int i = 0; i < 10; i++)
      cin >> id[i];

   int search;
   cout << "Enter student ID to search: ";
   cin >> search;

   int low = 0, high = 9, mid;
   bool found = false;

   while (low <= high) {
      mid = (low + high) / 2;
      if (id[mid] == search) {
         found = true;
         break;
      } else if (id[mid] < search)
         low = mid + 1;
      else
         high = mid - 1;
   }

   if (found)
      cout << "Student found at position " << mid + 1;
   else
      cout << "Student not found.";
}
```

Output:



NAME:MOHSIN MUNIR **INSTRUCTOR:SIR SHARJEEL**
SEC: B **ROLL NO: FA24 BSET 077**

ASSIGNMENT LAB 04

NAME:MOHSIN MUNIR **INSTRUCTOR:SIR SHARJEEL**
SEC: B **ROLL NO: FA24 BSET 077**

ASSIGNMENT LAB 04

Q 3. ***In a classroom of 15 students, the teacher wants to organize the student's scores from the lowest to the highest. Implement a bubble sort algorithm to arrange the students' scores in ascending order, facilitating the ranking of their performance***

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

int main() {
   int score[15];
   cout << "Enter scores of 15 students: ";
   for (int i = 0; i < 15; i++)
      cin >> score[i];

   for (int i = 0; i < 14; i++) {
      for (int j = 0; j < 14 - i; j++) {
         if (score[j] > score[j + 1]) {
            int temp = score[j];
            score[j] = score[j + 1];
            score[j + 1] = temp;
         }
      }
   }

   cout << "Scores in ascending order: ";
   for (int i = 0; i < 15; i++)
      cout << score[i] << " ";
}
```