Lab 01 – Arrays

Task 01:

Let’s analyze the annual match attendance of a sports stadium from 2016 to 2025. You have an array called ATTENDANCE that records the number of spectators for each year. Write a program that computes the following:

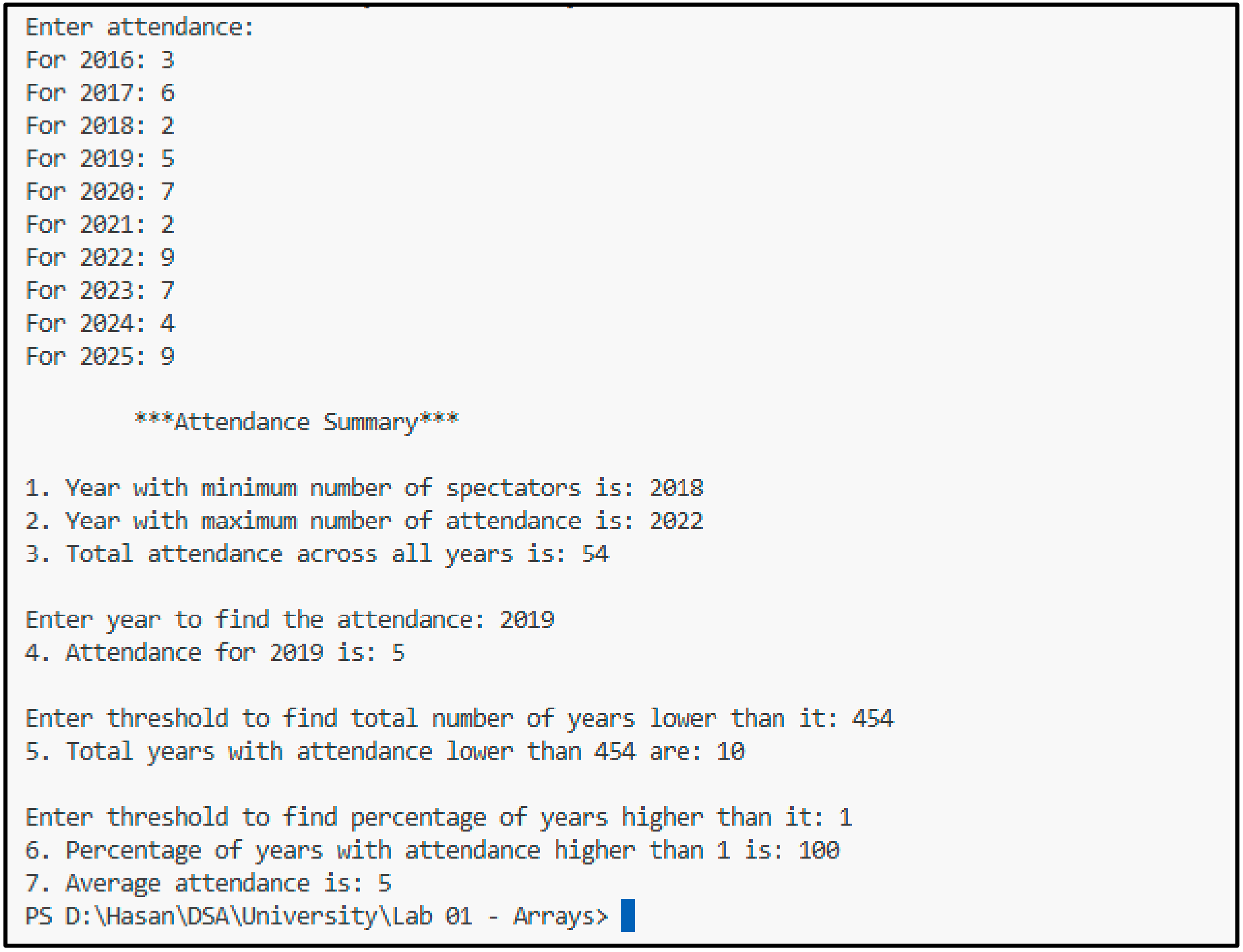
*a) **Lowest Attendance Year:** Identify the year with the minimum number of spectators.*
*b) **Highest Attendance Year:** Find the year with the maximum attendance.*
*c) **Total Attendance:** Calculate the total number of spectators overall*
*d) **Attendance for a Specific Year:** Retrieve attendance for a user-given year. Years.*
*e) **Years with Low Attendance:** Count how many years had attendance below a given threshold (e.g., 3,000).*
*f) **Percentage of High-Attendance Years:** Compute the percentage of years with attendance above 5,000.*
*g) **Prediction:** Compute the average attendance over the given years*

Code:

```
/*Tracking Sports Crowd*/

#include<iostream>
using namespace std;

int attendance[10];
int n = 10;

// a) Lowest attendance year
int lowestAttendanceYear(){
    int minId = 0;

    for(int i = 1; i < n; i++){
        if(attendance[i] < attendance[minId]) minId = i;
    }

    // finding year on the basis of index and returning it
    return 2016+minId;
}

// b) Highest attendance year
int highestAttendanceYear(){
    int maxId = 0;

    for(int i = 1; i < n; i++){
        if(attendance[i] > attendance[maxId]) maxId = i;
    }
    return 2016+maxId;
}

// c) Total attendance
int totalAttendance(){
    int sum = 0;
    for(int i = 0; i < n; i++) sum += attendance[i];
    return sum;
}

// d) Attendance for a specific year
int specificAttendance(int targetYear){
    int id = targetYear-2016;
    return attendance[id];
}

// e) Years with Low Attendance: Count how many years had attendance below a
given threshold (e.g 3,000).
int yearWithLowAttend(int threshold){
    int count = 0;

    for(int i = 0; i < n; i++){
        if(attendance[i] < threshold) count++;
    }
    return count;
}

// f) Percentage of High-Attendance Years: Compute the percentage of years with
attendance above 5,000.
float highAttendYearPer(int threshold){
    int count = 0;

    for(int i = 0; i < n; i++){
        if(attendance[i] > threshold) count++;
    }
    return (count/n)*100.0;
}

// g) Prediction: Compute the average attendance over the given years.
int predictAttendance(){
    int sum = totalAttendance();
    return sum/n;
}
```

```
int main(){
    cout << "Enter attendance: " << endl;
    for(int i = 0, year = 2016; i < n; i++){
        cout << "For " << year++ << ": ";
        cin >> attendance[i];
    }

    cout << "\n\t***Attendance Summary***" << endl << endl;
    cout << "1. Year with minimum number of spectators is: " <<
lowestAttendanceYear() << endl;
    cout << "2. Year with maximum number of attendance is: " <<
highestAttendanceYear() << endl;
    cout << "3. Total attendance across all years is: " << totalAttendance() <<
endl;

    int year = 0;
    cout << "\nEnter year to find the attendance: ";
    cin >> year;
    cout << "4. Attendance for " << year << " is: " << specificAttendance(year)
<< endl;

    int lowThreshold = 0;
    cout << "\nEnter threshold to find total number of years lower than it: ";
    cin >> lowThreshold;
    cout << "5. Total years with attendance lower than "<<lowThreshold<<" are: "
<< yearWithLowAttend(lowThreshold) << endl;

    int highThreshold = 0;
    cout << "\nEnter threshold to find percentage of years higher than it: ";
    cin >> highThreshold;
    cout << "6. Percentage of years with attendance higher than
"<<highThreshold<<" is: " << highAttendYearPer(highThreshold) << endl;

    cout << "7. Average attendance is: " << predictAttendance() << endl;
}
```

Output:



```
Enter attendance:
For 2016: 3
For 2017: 6
For 2018: 2
For 2019: 5
For 2020: 7
For 2021: 2
For 2022: 9
For 2023: 7
For 2024: 4
For 2025: 9

        ***Attendance Summary***

1. Year with minimum number of spectators is: 2018
2. Year with maximum number of attendance is: 2022
3. Total attendance across all years is: 54

Enter year to find the attendance: 2019
4. Attendance for 2019 is: 5

Enter threshold to find total number of years lower than it: 454
5. Total years with attendance lower than 454 are: 10

Enter threshold to find percentage of years higher than it: 1
6. Percentage of years with attendance higher than 1 is: 100
7. Average attendance is: 5
PS D:\Hasan\DSA\University\Lab 01 - Arrays>
```

Task 02:

Priority shift in amusement park

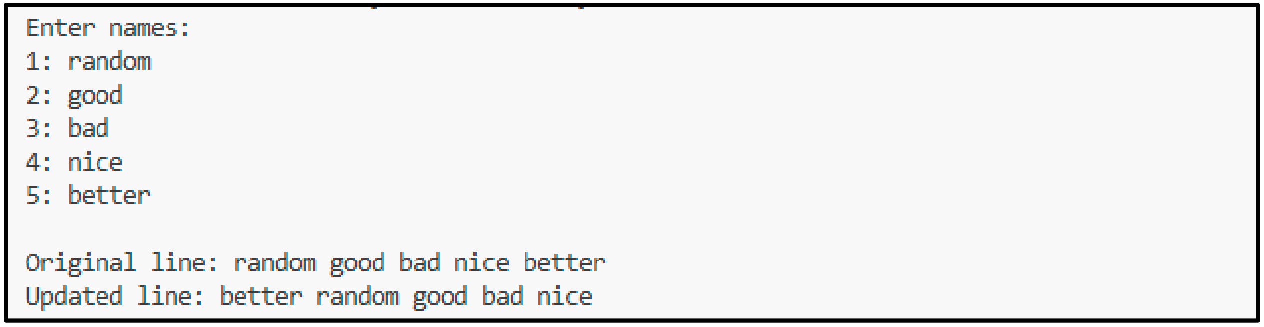
Sample input:
[Anum, Maryam, Daniyal, Bilal, Saad]

Sample Output:
[Saad, Anum, Maryam, Daniyal, Bilal]

Code:

```cpp
/*Priority Shift in Amusement Park*/

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

void rightRotate(string names[], int n){
    string last = names[n-1];

    // rotating
    for(int i = n-1; i > 0; i--) names[i] = names[i-1];

    names[0] = last;
}

int main(){
    string names[5];

    cout << "Enter names: " << endl;
    for(int i = 0; i < 5; i++){
        cout << i+1 << ": ";
        getline(cin, names[i]);
    }

    cout << "\nOriginal line: ";
    for(string s : names) cout << s << " ";
    cout << endl;

    rightRotate(names, 5);

    cout << "Updated line: ";
    for(string s : names) cout << s << " ";
    cout << endl;
}
```

Output:



```
Enter names:
1: random
2: good
3: bad
4: nice
5: better

Original line: random good bad nice better
Updated line: better random good bad nice
```

Task 03:

Seminar	Food Festival	Concert	Trade Show	Birthday	Workshop	Wedding Anniversary	Graduation Ceremony		
1-09-2024	3-09-2024	5-09-2024	11-09-2024	18-09-2024	19-09-2024	21-09-2024	23-09-2024		

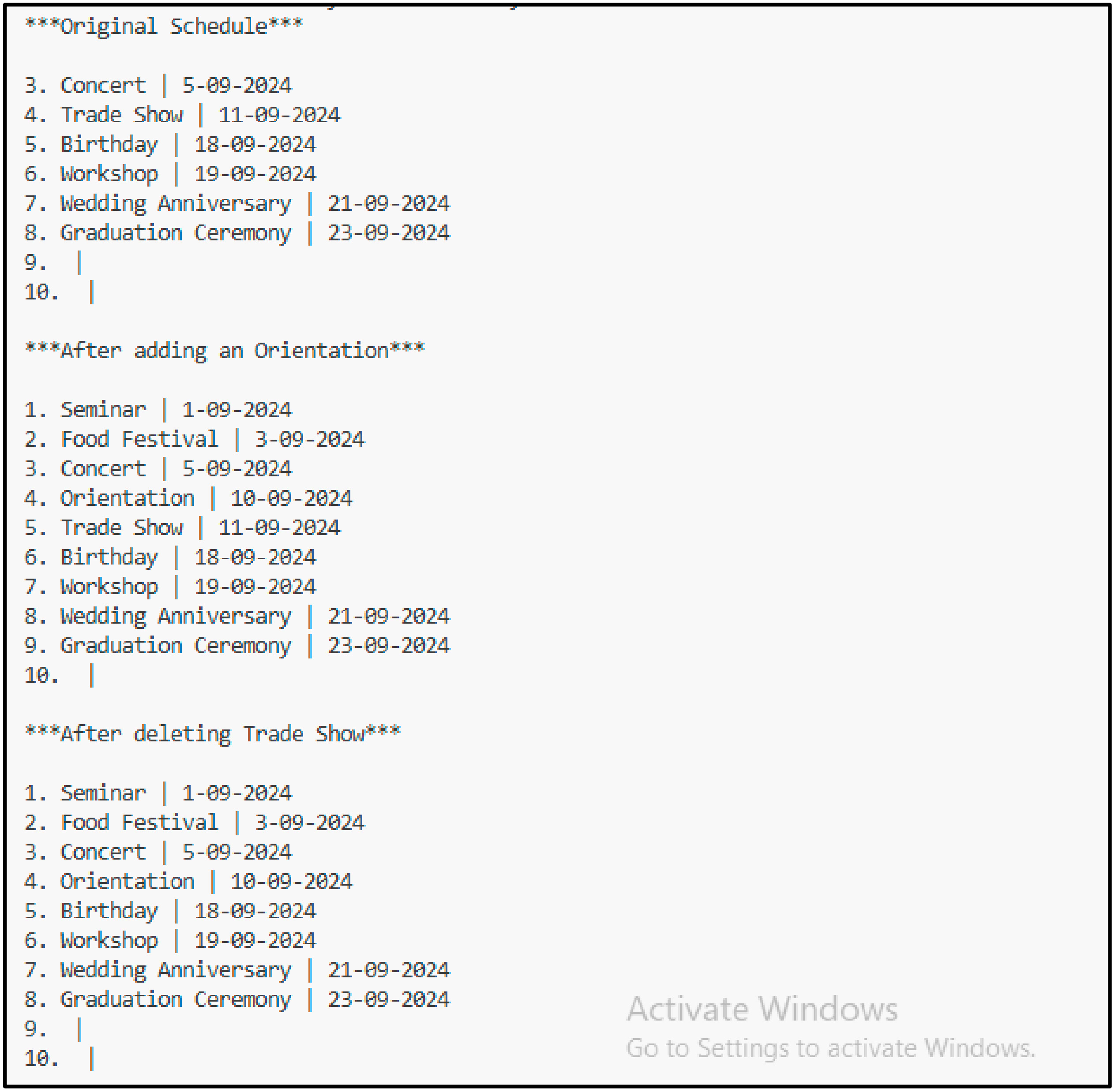
a. Insert a new event “Orientation” date: 10-09-2024
b. Delete the event of Trade Show from list.
c. Show the list of events after each of the above tasks.

Code:

```
/*Event Scheduling System*/

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

struct Event{
    string name, date;
};

Event events[10] = {
    {"Seminar", "1-09-2024"},
    {"Food Festival", "3-09-2024"},
    {"Concert", "5-09-2024"},
    {"Trade Show", "11-09-2024"},
    {"Birthday", "18-09-2024" },
    {"Workshop", "19-09-2024"},
    {"Wedding Anniversary", "21-09-2024"},
    {"Graduation Ceremony", "23-09-2024"}
};
int n = 10;
```

```cpp
void insertEvent(string name, string date, int id){
    // first of all shifting
    for(int i = n-1; i > id;  i--){
        events[i].name = events[i-1].name;
        events[i].date = events[i-1].date;
    }

    // inserting
    events[id].name = name;
    events[id].date = date;
}

void deleteEvent(int id){
    for(int i = id; i < n-1; i++){
        events[i].name = events[i+1].name;
        events[i].date = events[i+1].date;
    }

    events[n-1].name = "";
    events[n-1].name = "";
}

int main(){

    cout << "***Original Schedule***" << endl << endl;
    for(int i = 0; i < n; i++){
        cout << i+1 << ". " << events[i].name << " | " << events[i].date << endl;
    }

    // a.   Insert a new event “Orientation” date: 10-09-2024
    insertEvent("Orientation", "10-09-2024", 3);

    cout << "\n***After adding an Orientation***" << endl << endl;
    for(int i = 0; i < n; i++){
        cout << i+1 << ". " << events[i].name << " | " << events[i].date << endl;
    }

    // b.   Delete the event of Trade Show from list.
    deleteEvent(4);
    cout << "\n***After deleting Trade Show***" << endl << endl;
    for(int i = 0; i < n; i++){
        cout << i+1 << ". " << events[i].name << " | " << events[i].date << endl;
    }
}
```

Output:



```
***Original Schedule***

3. Concert | 5-09-2024
4. Trade Show | 11-09-2024
5. Birthday | 18-09-2024
6. Workshop | 19-09-2024
7. Wedding Anniversary | 21-09-2024
8. Graduation Ceremony | 23-09-2024
9.  |
10.  |

***After adding an Orientation***

1. Seminar | 1-09-2024
2. Food Festival | 3-09-2024
3. Concert | 5-09-2024
4. Orientation | 10-09-2024
5. Trade Show | 11-09-2024
6. Birthday | 18-09-2024
7. Workshop | 19-09-2024
8. Wedding Anniversary | 21-09-2024
9. Graduation Ceremony | 23-09-2024
10.  |

***After deleting Trade Show***

1. Seminar | 1-09-2024
2. Food Festival | 3-09-2024
3. Concert | 5-09-2024
4. Orientation | 10-09-2024
5. Birthday | 18-09-2024
6. Workshop | 19-09-2024
7. Wedding Anniversary | 21-09-2024
8. Graduation Ceremony | 23-09-2024
9.  |
10.  |
```

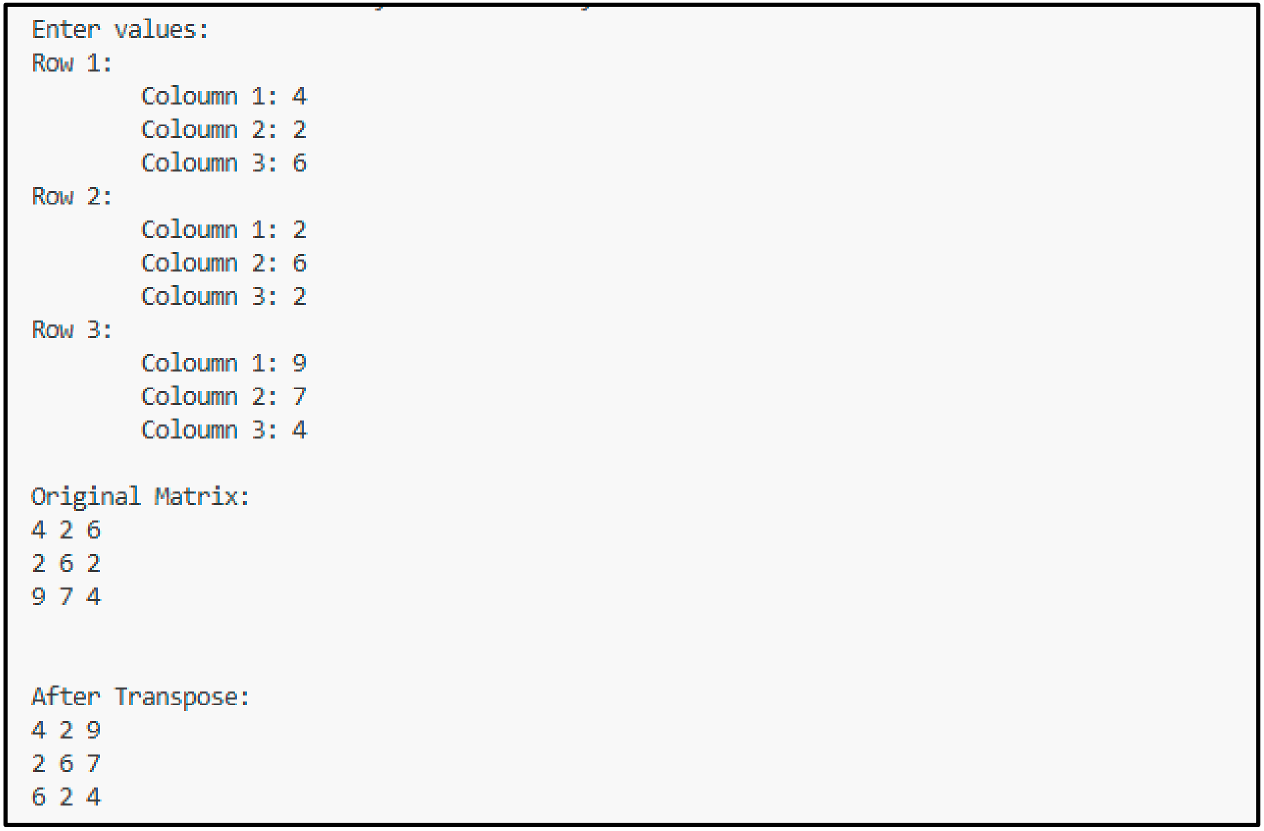
Task 04:

Take transpose of a 3x3 matrix.

Code:

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


int main(){

    int prices[3][3];

    cout << "Enter values: " << endl;
    for(int i = 0; i < 3; i++){
        cout << "Row " << i+1 << ": " << endl;
        for(int j = 0; j < 3; j++){
            cout << "\tColoumn " << j+1 << ": ";
            cin >> prices[i][j];
        }
    }

    cout << "\nOriginal Matrix: " << endl;
    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            cout << prices[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;

    // taking transpose
    int tPrices[3][3];

    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            tPrices[i][j] = prices[j][i];
        }
    }

    cout << "\nAfter Transpose: " << endl;
    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            cout << tPrices[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;
}
```

Output:



```
Enter values:
Row 1:
        Coloumn 1: 4
        Coloumn 2: 2
        Coloumn 3: 6
Row 2:
        Coloumn 1: 2
        Coloumn 2: 6
        Coloumn 3: 2
Row 3:
        Coloumn 1: 9
        Coloumn 2: 7
        Coloumn 3: 4

Original Matrix:
4 2 6
2 6 2
9 7 4


After Transpose:
4 2 9
2 6 7
6 2 4
```