

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
    for( int i = 0, year = 2016; i < n; i++){
        if(minId == i) return year;
        else year++;
    }
    return -1;
}
```

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

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

    // finding year on the basis of index and returning it
    for( int i = 0, year = 2016; i < n; i++){
        if(maxId == i) return year;
        else year++;
    }
    return -1;
}

// 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 = -1;

    // finding index
    for(int i = 0, year = 2016; i < n; i++){
        if(year == targetYear){
            id = i;
            break;
        }
        else year++;
    }
    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 << "\n4. Attendance for " << year << " is: " << specificAttendance(year)
<< endl;
    cout << "5. Total years with attendance lower than 3000 are: " <<
yearWithLowAttend(3000) << endl;

```

```
    cout << "6. Percentage of years with attendance higher than 5000 is: " <<
highAttendYearPer(5000) << endl;
    cout << "7. Average attendance is: " << predictAttendance() << endl;
}
```

Output:

```
Enter attendance:
For 2016: 3
For 2017: 5
For 2018: 2
For 2019: 5
For 2020: 8
For 2021: 5
For 2022: 8
For 2023: 4
For 2024: 8
For 2025: 2

***Attendance Summary***

1. Year with minimum number of spectators is: 2018
2. Year with maximum number of attendance is: 2020
3. Total attendance across all years is: 50

Enter year to find the attendance: 2025

4. Attendance for 2025 is: 2
5. Total years with attendance lower than 3000 are: 10
6. Percentage of years with attendance higher than 5000 is: 0
7. Average attendance is: 5
PS D:\Hasan\DSA\University\Lab 01 - Arrays> █
```

Task 02:

Sample input:

[Anum, Maryam, Daniyal, Bilal, Saad]

Sample Output:

[Saad, Anum, Maryam, Daniyal, Bilal]

Code:

```
/*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: better
```

```
5: amazing
```

```
Original line: Random good bad better amazing
```

```
Updated line: amazing Random good bad better
```

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		

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

Code:

```
/*Event Scheduling System*/
```

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
string eventName[10] = {"Seminar", "Food Festival", "Concert", "Trade Show",  
"Birthday", "Workshop", "Wedding anniversary", "Graduation"};
```

```
string eventDate[10] = {"1-09-2024", "3-09-2024", "5-09-2024", "11-09-2024", "18-  
09-2024", "19-09-2024", "21-09-2024", "23-09-2024"};
```

```
int n = 10;
```

```
void insertEvent(string name, string date, int id){
```

```
    // first of all shifting
```

```
    for(int i = n-1; i > id; i--){  
        eventName[i] = eventName[i+1];  
        eventDate[i] = eventDate[i+1];  
    }
```

```
    // inserting
```

```
    eventName[id] = name;  
    eventDate[id] = date;
```

```
}
```

```
void deleteEvent(int id){
```

```
    for(int i = id; i < n-1; i++){  
        eventName[i] = eventName[i+1];  
        eventDate[i] = eventDate[i+1];  
    }
```

```
    eventName[n-1] = "";
```

```

    eventDate[n-1] = "";
}

int main(){
    cout << "Original Schedule: " << endl;
    cout << "Events: ";
    for(string s : eventName) cout << s << " | ";
    cout << "\nDates: ";
    for(string s : eventDate) cout << s << " | ";
    cout << endl;

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

    cout << "\nAfter adding an Orientation: " << endl;
    cout << "Events: ";
    for(string s : eventName) cout << s << " | ";
    cout << "\nDates: ";
    for(string s : eventDate) cout << s << " | ";
    cout << endl;

    // b. Delete the event of Trade Show from list.
    deleteEvent(4);
    cout << "\nAfter deleting Trade Show: " << endl;
    cout << "Events: ";
    for(string s : eventName) cout << s << " | ";
    cout << "\nDates: ";
    for(string s : eventDate) cout << s << " | ";
    cout << endl;
}

```

Output:

```

Original Schedule:
Events: Seminar | Food Festival | Concert | Trade Show | Birthday | Workshop | Wedding anniversary | Graduation | | |
Dates: 1-09-2024 | 3-09-2024 | 5-09-2024 | 11-09-2024 | 18-09-2024 | 19-09-2024 | 21-09-2024 | 23-09-2024 | | |

After adding an Orientation:
Events: Seminar | Food Festival | Concert | Orientation | Trade Show | Birthday | Workshop | Wedding anniversary | Graduation | |
Dates: 1-09-2024 | 3-09-2024 | 5-09-2024 | 10-09-2024 | 11-09-2024 | 18-09-2024 | 19-09-2024 | 21-09-2024 | 23-09-2024 | |

After deleting Trade Show:
Events: Seminar | Food Festival | Concert | Orientation | Birthday | Workshop | Wedding anniversary | Graduation | | |
Dates: 1-09-2024 | 3-09-2024 | 5-09-2024 | 10-09-2024 | 18-09-2024 | 19-09-2024 | 21-09-2024 | 23-09-2024 | | |
PS D:\Hasan\DSA\University\Lab 01 - Arrays>

```


Task 04:**Input**

250000	420000	310000
230000	400000	295000
270000	450000	330000

Output

250000	230000	270000
420000	400000	450000
310000	295000	330000

Code:

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

int main(){

    int prices[3][3] = {
        {250000, 420000, 310000},
        {230000, 400000, 295000},
        {270000, 450000, 330000}
    };

    cout << "Original 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 << "After 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:

```
Original Matrix:
250000 420000 310000
230000 400000 295000
270000 450000 330000
```

```
After Transpose:
250000 230000 270000
420000 400000 450000
310000 295000 330000
```

```
PS D:\Hasan\DSA\University\Lab 01 - Arrays> █
```