# LAB-3

# Array and Structure Operations in C

Q1) Write a program to find the second largest element in an integer array without sorting the array. Consider the array may contain duplicate values.

```
#include <stdio.h>
#include <limits.h>

int main(){
   int n;
   long long x;
   long long max=LLONG_MIN;
   long long secondmax=LLONG_MIN;
   scanf("%d",&n);
   for(int i=0;i<n;i++){
      scanf("%lld",&x);
      if(x>max){
         secondmax=max;
         max=x;
      }
      else if(secondmax<x && secondmax!=max){
         secondmax=x;
      }
   }
   printf("The Second maximum element in array is : %lld",secondmax);
   return 0;
}
```

Sample Input

```
4

12 34 20 6
```

Your Output

```
The Second maximum element in array is : 20
```

Q2) Given an array of integers, write a program to check if there exists a contiguous subarray whose sum equals a given value S. If it exists, print the starting and ending indices of such subarray.

```
#include <stdio.h>
#include <stdbool.h>
#include <limits.h>

int main(){
   int n;
   long long S;
   scanf("%d%lld",&n,&S);

   long long arr[n];
   for(int i=0;i<n;i++){
      long long x;
      scanf("%lld",&x);
      arr[i]=x;
   }

   int ans[2];
   bool isit=false;

   for(int i=0;i<n;i++){
      long long sum=0;
      for(int j=i;j<n;j++){
         sum+=arr[j];
         if(sum==S){
            isit=true;
            ans[0]=i;
            ans[1]=j;
            break;
         }
      }
      if(isit){
         break;
      }
   }

   if(isit){
      printf("Yes There is contiguous subarray whose sum equals to S\n");
      printf("The following starting and ending indices of subarray : %d %d\n",
```

```
ans[0], ans[1]);
   }
   else{
      printf("No Such subarray found\n");
   }

   return 0;
}
```

Sample Input

```
5
54
20 34 6 12 24
```

Your Output

```
Yes There is contiguous subarray whose sum equals to S
The following starting and ending indices of subarray : 0 1
```

Q3) Write a function to rotate an array of n elements to the right by k positions. For example, for array [1, 2, 3, 4, 5] and k = 2, the rotated array should be [4, 5, 1, 2, 3]. Try to do in O(n) time complexity

```
#include <stdio.h>
#include <stdbool.h>
#include <limits.h>

int main(){
   int n;
   long long k;
   scanf("%d%lld",&n,&k);

   long long arr[n];
   for(int i=0;i<n;i++){
      long long x;
      scanf("%lld",&x);
      arr[i]=x;
   }
   long long RotatedArr[n];
```

```
    int j=0;
    for(int i=(n-k);i<n;i++){
        RotatedArr[j++]=arr[i];
    }
    for(int i=0;i<(n-k);i++){
        RotatedArr[j++]=arr[i];
    }
    printf("The Rotated Array after k positions are : ");
    for(int i=0;i<n;i++){
        printf("%lld ", RotatedArr[i]);
    }
    printf("\n");
    return 0;
}
```

Sample Input

```
5
3
20 34 6 12 24
```

Your Output

```
The Rotated Array after k positions are : 6 12 24 20 34
```

Q4) Define a structure Student with fields name, rollNumber, and marks. Write a program to input details of n students and then display the details of the student with the highest marks.

```
#include <stdio.h>
#include <string.h>

struct Student{
    char name[50];
    int rollNumber;
    float marks;
};

int main(){
    int n;
```

```
    printf("Enter number of students: ");
    scanf("%d",&n);

    struct Student students[n];

    for(int i=0;i<n;i++){
        printf("Enter details of student %d:\n", i+1);
        printf("Name: ");
        scanf(" %[^\n]s", students[i].name);
        printf("Roll Number: ");
        scanf("%d", &students[i].rollNumber);
        printf("Marks: ");
        scanf("%f", &students[i].marks);
    }

    int maxIndex = 0;

    for(int i=1;i<n;i++){
        if(students[i].marks > students[maxIndex].marks){
            maxIndex = i;
        }
    }

    printf("\nStudent with highest marks:\n");
    printf("Name: %s\n", students[maxIndex].name);
    printf("Roll Number: %d\n", students[maxIndex].rollNumber);
    printf("Marks: %.2f\n", students[maxIndex].marks);

    return 0;
}
```

Sample Input

```
3
Ohi
34
92
Tiger
37
86
```

Veer

48

68

Your Output

Enter number of students: Enter details of student 1:

Name: Roll Number: Marks: Enter details of student 2:

Name: Roll Number: Marks: Enter details of student 3:

Name: Roll Number: Marks:

Student with highest marks:

Name: Ohi

Roll Number: 34

Marks: 92.00

Q5) Define a structure Employee with fields id, name, salary, and department. Write a program to increase the salary of all employees in the "IT" department by 10%.

```
#include <stdio.h>
#include <string.h>

struct Employee{
   int id;
   char name[50];
   float salary;
   char department[20];
};

int main(){
   int n;
   printf("Enter number of employees: ");
   scanf("%d",&n);

   struct Employee employe[n];

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

```
    printf("Enter details of Employee %d:\n", i+1);
    printf("Employee ID: ");
    scanf("%d", &employe[i].id);
    printf("Name: ");
    scanf(" %[^\n]s", employe[i].name);
    printf("Salary: ");
    scanf("%f", &employe[i].salary);
    printf("Department: ");
    scanf(" %[^\n]s", employe[i].department);
  }

  for(int i=0;i<n;i++){
    if(strcmp(employe[i].department, "IT") == 0){
      employe[i].salary *= 1.10;
    }
  }

  printf("\nEmployee details after salary update:\n");
  for(int i=0;i<n;i++){
    printf("ID: %d\n", employe[i].id);
    printf("Name: %s\n", employe[i].name);
    printf("Salary: %.2f\n", employe[i].salary);
    printf("Department: %s\n\n", employe[i].department);
  }

  return 0;
}
```

Sample Input

```
3
34
Ohi
92000
AI
37
Tiger
88000
CSE
48
```

Veer

68000

IT

Your Output

Enter number of employees: Enter details of Employee 1:

Employee ID: Name: Salary: Department: Enter details of Employee 2:

Employee ID: Name: Salary: Department: Enter details of Employee 3:

Employee ID: Name: Salary: Department:

Employee details after salary update:

ID: 34

Name: Ohi

Salary: 92000.00

Department: AI

ID: 37

Name: Tiger

Salary: 88000.00

Department: CSE

ID: 48

Name: Veer

Salary: 74800.00

Department: IT

Q6) Define a structure Book with fields title, author, and price. Input details for n books and then display all books whose price is above a given value.

```
#include <stdio.h>
#include <string.h>

struct Book {
   char title[100];
```

```
    char author[50];
    float price;
};

int main() {
    int n;
    scanf("%d", &n);
    printf("Enter number of books: %d",n);

    struct Book books[n];
    printf("\nEnter details of Book\n");
    for(int i = 0; i < n; i++) {
        scanf(" %[^\n]", books[i].title);
        scanf(" %[^\n]", books[i].author);
        scanf("%f", &books[i].price);
    }

    float limit;
    printf("\nEnter price limit");
    scanf("%f", &limit);

    printf("\nBooks with price above %.2f:\n", limit);
    int found = 0;
    for(int i = 0; i < n; i++) {
        if(books[i].price > limit) {
            printf("Title: %s\n", books[i].title);
            printf("Author: %s\n", books[i].author);
            printf("Price: %.2f\n\n", books[i].price);
            found = 1;
        }
    }

    if(!found) {
        printf("No books found above %.2f\n", limit);
    }

    return 0;
}
```

Sample Input

```
3
```

Arc Fault Dedection

Ohi

200

IOT Optimization

Tiger

220

Drone Technology

Veer

180

199

Your Output

Enter number of books: 3

Enter details of Book

Enter price limit

Books with price above 199.00:

Title: Arc Fault Dedection

Author: Ohi

Price: 200.00

Title: IOT Optimization

Author: Tiger

Price: 220.00

Q7) Define a structure Date with fields day, month, and year. Write a program to input two dates and determine which date is earlier.

```
#include <stdio.h>

struct Date {
   int day;
   int month;
```

```
    int year;
};

int main() {
    struct Date d1, d2;

    printf("Enter first date (day month year): ");
    scanf("%d %d %d", &d1.day, &d1.month, &d1.year);

    printf("Enter second date (day month year): ");
    scanf("%d %d %d", &d2.day, &d2.month, &d2.year);

    printf("\nFirst Date: %02d-%02d-%04d", d1.day, d1.month, d1.year);
    printf("\nSecond Date: %02d-%02d-%04d\n", d2.day, d2.month, d2.year);

    if(d1.year < d2.year ||
      (d1.year == d2.year && d1.month < d2.month) ||
      (d1.year == d2.year && d1.month == d2.month && d1.day < d2.day)) {
        printf("\nThe earlier date is: %02d-%02d-%04d\n", d1.day, d1.month,
d1.year);
    }
    else if(d1.year == d2.year && d1.month == d2.month && d1.day == d2.day) {
        printf("\nBoth dates are the same.\n");
    }
    else {
        printf("The earlier date is: %02d-%02d-%04d\n", d2.day, d2.month,
d2.year);
    }

    return 0;
}
```

Sample Input

12 5 2023

25 4 2023

Your Output

Enter first date (day month year): Enter second date (day month year):

First Date: 12-05-2023

Second Date: 25-04-2023

The earlier date is: 25-04-2023

Q8) Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

```
#include <stdio.h>
#include <stdbool.h>

int main() {
   int n;
   long long target;

   scanf("%d", &n);
   scanf("%lld", &target);

   int arr[n];
   for(int i = 0; i < n; i++){
      scanf("%d", &arr[i]);
   }

   int start = -1;
   int end = -1;
   bool isit = false;

   int number = 0;
   for(int i = 0; i < n; i++){
      int check = target - arr[i];
      for(int j = 0; j < n; j++){
         if(j != i && arr[j] == check){
            start = i;
            end = j;
            isit = true;
            break;
         }
      }
      if(isit) break;
   }

   if(isit){
      printf("The indices of elements are: %d  , %d\n", start, end);
   }
```

```
   else{
      printf("No such pair exists.\n");
   }
   return 0;
}
```

Sample Input

```
4

54

20 24 6 34
```

Your Output

```
The indices of elements are: 0 , 3
```

Q9) Write a function to find the longest common prefix string amongst an array of strings.

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

int main() {
   int n;
   scanf("%d", &n);

   char arr[n][34];
   for(int i = 0; i < n; i++) {
      scanf(" %[^\n]", arr[i]);
   }

   char str[34] = "";
   char prev[34];
   strcpy(prev, arr[0]);

   int shortestStrLength = strlen(prev);

   for(int i = 0; i < shortestStrLength; i++) {
      char curr = prev[i];
```

```
    bool isit = true;

    for(int j = 1; j < n; j++) {
      if(arr[j][i] != curr) {
        isit = false;
        break;
      }
    }

    if(isit) {
      int len = strlen(str);
      str[len] = curr;
      str[len+1] = '\0';
    } else {
      break;
    }
  }

  if(strlen(str) > 0) {
    printf("%s\n", str);
  } else {
    printf("\n");
  }

  return 0;
}
```

Sample Input

```
4
Ohi
OhiTiger
OhiVeer
OhiUippto
```

Your Output

```
Ohi
```

Q10) Given an integer array nums, return true if any value appears at least twice in the array, and return false if every element is distinct.

```
#include <stdio.h>
#include <stdbool.h>

int main() {
   int n;
   scanf("%d", &n);

   int arr[n];
   for(int i = 0; i < n; i++){
      scanf("%d", &arr[i]);
   }

   bool isit = false;
   int count = 0;

   for(int i = 0; i < n; i++){
      int num = arr[i];
      for(int j = 0; j < n; j++){
         if(i != j && num == arr[j]){
            count++;
         }
         if(count > 0){
            isit = true;
            break;
         }
      }
      count = 0;
      if(isit) break;
   }

   if(isit){
      printf("TRUE");
   }
   else{
      printf("FALSE");
   }
   return 0;
}
```

Sample Input

4

34 20 34 12

Your Output

TRUE