

**Name: Chandana Ramesh Galgali Batch: B1 Roll No.: 16010422234**

**Experiment No.: 1**

**Aim:**  Implement and demonstrate the use of arrays, array of structure and pointers using C.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Resources needed:** Turbo C/C++ editor and C compiler (Online/Offline)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Theory:**

1. **Arrays:**

Arrays are a collection of elements of the same data type, stored in contiguous memory locations. They allow for efficient storage and retrieval of multiple values using a single variable name.

**Examples:**

***int* numbers[5] = {1, 2, 3, 4, 5};**

1. **Structures:**

Structures are user-defined data types that allow you to group different variables of different data types under a single name. They provide a way to represent a complex entity with multiple attributes.

**Examples:**

***struct* *Person* {**

***char* name[20];**

***int* age;**

***float* height;**

**};**

***struct* *Person* *person1*;**

1. **Array of Structures:**

An array of structures is an array where each element is a structure. It allows you to store multiple instances of a structure in a sequential manner.

**Examples:**

***struct* *Person* *people*[3] = {**

**{"Alice", 25, 1.65},**

**{"Bob", 30, 1.75},**

**{"Charlie", 35, 1.80}**

**};**

1. **Pointers and Pointers to Structures:**

Pointers are variables that store memory addresses. They allow you to indirectly access and manipulate data by pointing to the memory location where the data is stored.

**Examples:**

***int* number = 10;**

***int* \*ptr = &number;**

Pointers to structures are variables that store the memory address of a structure. They allow you to access and modify the structure's members indirectly.

**Examples:**

***struct* *Person* \**ptr* = &person1;**

1. **Functions and Function signature:**

Functions are blocks of code that perform a specific task. They allow you to modularize your code and reuse it. Functions can have input parameters and return values.

**Examples:**

***int* *add*(*int* a, *int* b) {**

***return* a + b;**

**}**

A function signature is a declaration that specifies the function's name, return type, and parameter types. It helps the compiler understand how to call and use the function correctly.

**Examples:**

***int* *add*(*int* a, *int* b);**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Activity :** Implementing a C program to create a roll call list of a class **using array of structure concept**. It has the details of students as roll number and name. Program should support following operations.

1. **Insert into last position.**
2. **Delete from last position.**
3. **Search specific student.**
4. **Display complete list of student with details.**

**Results:** A C program depicting the correct behaviour of mentioned concept and capable of handling all possible exceptional conditions/inputs and the same is reflecting clearly in the output.

**Program and Output:**

**Code:**

#include<stdio.h>

struct roll\_call{

int roll\_no;

char name[50], surname[50];

}r[100],t;

int n,i,j;

void main()

{

int o;

printf("\n1. Insert into last position.\n2. Delete from last position.\n3. Search specific student.\n4. Display complete list of students with details.\n5.Exit.\n");

printf("Enter your choice.");

scanf("%d",&o);

switch(o)

{

case 1:

{

insert();

}

break;

case 2:

{

delete();

}

break;

case 3:

{

search();

}

break;

case 4:

{

display();

}

break;

case 5:

break;

default:

{

printf("Choose a valid option");

}

break;

}

}

void insert()

{

printf("Enter the total number of the students: ");

scanf("%d",&n);

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

{

printf("Enter the roll number of the student: ");

scanf("%d",&r[i].roll\_no);

printf("Enter the full name of the student(Name\_space\_Surname): ");

scanf("%s %s",&r[i].name,&r[i].surname);

}

printf("\*\*\*\*\*Roll-Call List of Students\*\*\*\*\*");

printf("\nRoll Number\t\tName\n");

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

{

printf("\n%d\t\t\t%s %s\n",r[i].roll\_no,r[i].name,r[i].surname);

}

main();

}

void delete()

{

int del\_roll\_no;

printf("Enter the roll number of the student: ");

scanf("%d",&del\_roll\_no);

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

{

if(del\_roll\_no!=r[i].roll\_no)

{

printf("\n%d\t\t\t%s %s\n",r[i].roll\_no,r[i].name,r[i].surname);

}

}

main();

}

void search()

{

int search\_roll\_no;

printf("Enter the roll number of the student: ");

scanf("%d",&search\_roll\_no);

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

{

if(search\_roll\_no==r[i].roll\_no)

{

printf("\n%d\t\t\t%s %s\n",r[i].roll\_no,r[i].name,r[i].surname);

}

}

main();

}

void display()

{

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

{

for(j=1;j<n;j++)

{

if(r[j].roll\_no>r[j+1].roll\_no)

{

t=r[j];

r[j]=r[j+1];

r[j+1]=t;

}

}

}

printf("\*\*\*\*\*Roll-Call List of Students\*\*\*\*\*");

printf("\nRoll Number\t\tName\n");

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

{

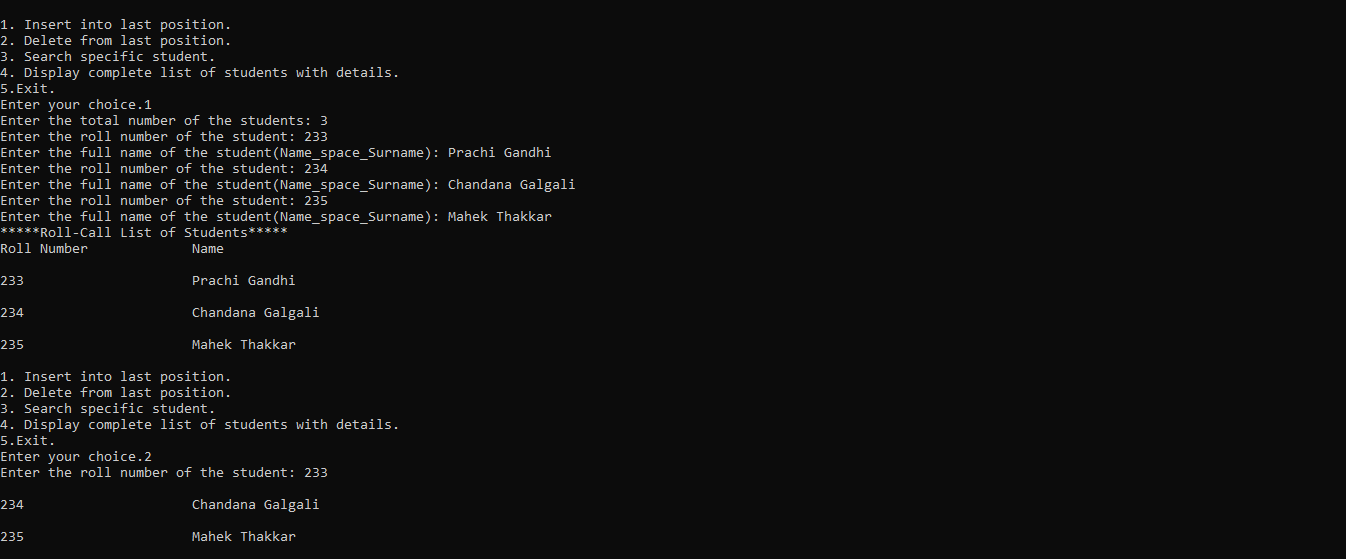
printf("\n%d\t\t\t%s %s\n",r[i].roll\_no,r[i].name,r[i].surname);

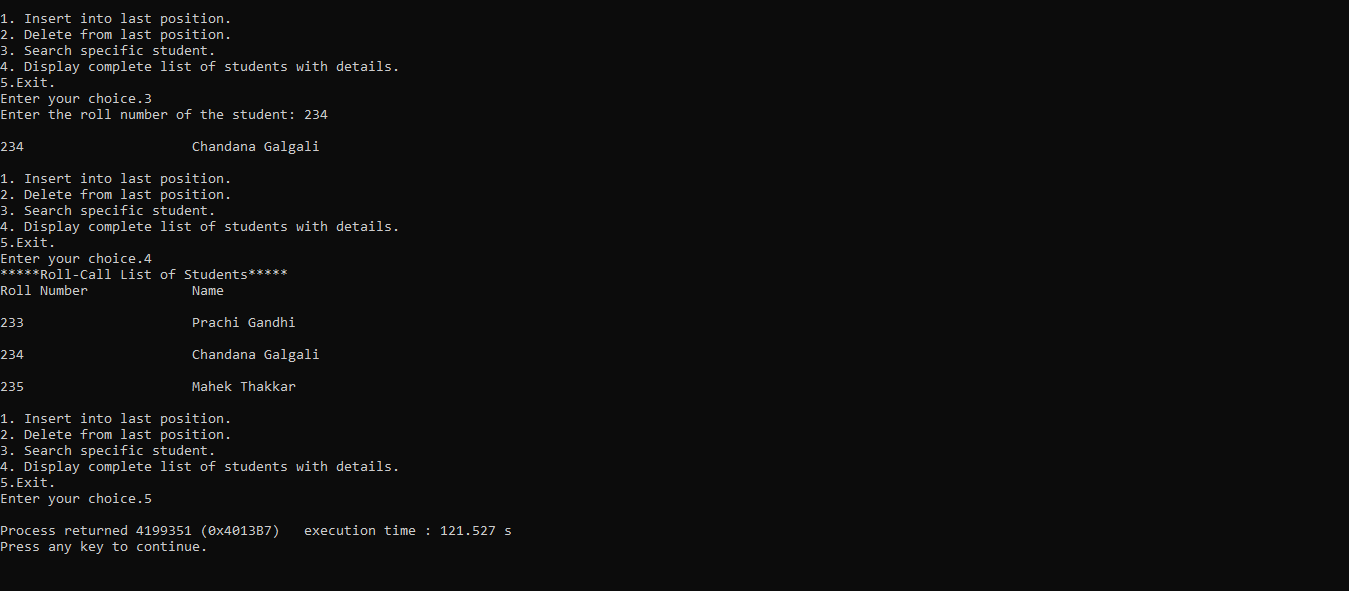
}

main();

}

**Output:**

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Course Outcomes:**

**Explain the different data structures used in problem solving.**

**Conclusion:**

We could successfully implement an array of structures and functions in the given problem definition.

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of faculty in-charge with date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**References:**

**Books/ Journals/ Websites:**

* Y. Langsam, M. Augenstin and A. Tannenbaum, “**Data Structures using C**”, Pearson Education Asia, 1st Edition, 2002
* **Data Structures A Psedocode Approach with C**, Richard F. Gilberg&Behrouz A. Forouzan, secondedition, CENGAGE Learning