Manav Rachna International Institute of Research and Studies

Bachelor's in computer applications

Data Structures using C

Submitted by: Urvashi Pahuja

Department: School of Computer Applications

Course: Bachelor's in computer applications

Roll No: 24/SCA/BCA/087

Semester: 2nd

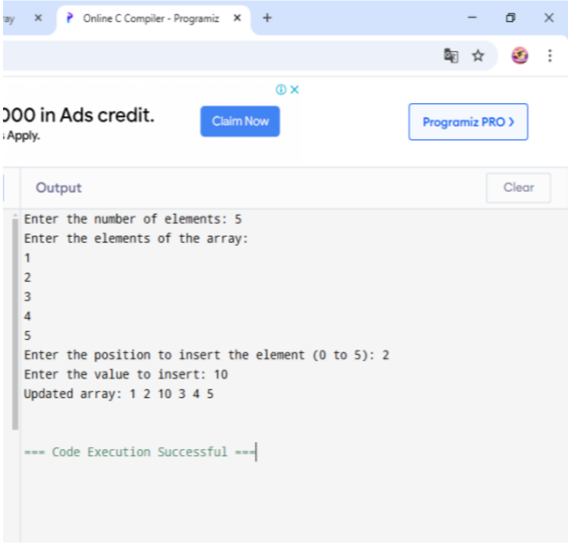
Subject: Data Structures using C

AIM-1. - WRITE A PROGRAM IN C TO IMPLEMENT INSERTION IN 1-D ARRAYS.

PROGRAM-

```
#include <stdio.h>

void insertElement(int arr[], int *size, int pos, int value) {
    // Shift elements to the right to make space for the new element
    for (int i = *size - 1; i >= pos; i--) {
     arr[i + 1] = arr[i];
    }

    // Insert the new element at the desired position
    arr[pos] = value;

    // Increment the size of the array
    (*size)++;
}
```

```
int main() {
    int arr[100], n, pos, value;

    // Input the size of the array
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    // Input the elements of the array
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Input the position and value to insert
    printf("Enter the position to insert the element (0 to %d): ", n);
    scanf("%d", &pos);
    printf("Enter the value to insert: ");
    scanf("%d", &value);

    // Check if the position is valid
    if (pos < 0 || pos > n) {
        printf("Invalid position!\n");
    } else {
        // Perform the insertion
        insertElement(arr, &n, pos, value);

        // Print the updated array
        printf("Updated array: ");
        for (int i = 0; i < n; i++) {
            printf("%d ", arr[i]);
        }
        printf("\n");
    }

    return 0;
}
```

OUTPUT-: 

```
Enter the number of elements: 5
Enter the elements of the array:
1
2
3
4
5
Enter the position to insert the element (0 to 5): 2
Enter the value to insert: 10
Updated array: 1 2 10 3 4 5


=== Code Execution Successful ===
```

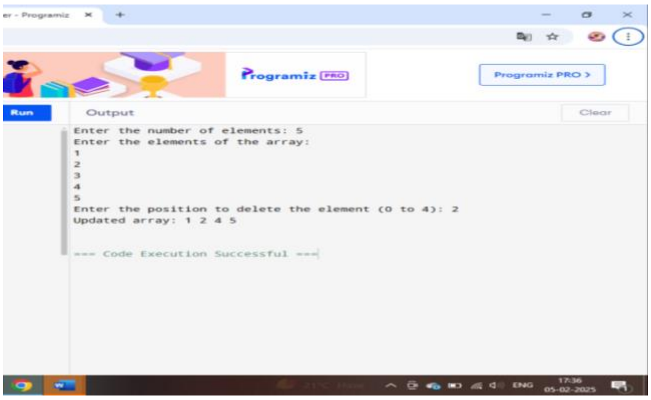
LEARNING OBJECTIVE- learned regarding the concept of arrays and insertion program of the arrays.

AIM-2.- WRITE A PROGRAM IN C TO IMPLEMENT DELETION IN 1-D ARRAYS.

PROGRAM- #include <stdio.h >

```
void deleteElement(int arr[], int *size, int pos) {
    // Check if the position is valid
    if (pos < 0 || pos >= *size) {
        printf("Invalid position!\n");
        return;
    }

    // Shift elements to the left to fill the gap
    for (int i = pos; i < *size - 1; i++) {
        arr[i] = arr[i + 1];
    }

    // Decrease the size of the array
    (*size)--;
}

int main() {
    int arr[100], n, pos;

    // Input the size of the array
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    // Input the elements of the array
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Input the position to delete
    printf("Enter the position to delete the element (0 to %d): ", n - 1);
    scanf("%d", &pos);

    // Perform the deletion
    deleteElement(arr, &n, pos);

    // Print the updated array
    printf("Updated array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

OUTPUT:-



```
Enter the number of elements: 5
Enter the elements of the array:
1
2
3
4
5
Enter the position to delete the element (0 to 4): 2
Updated array: 1 2 4 5


=== Code Execution Successful ===
```

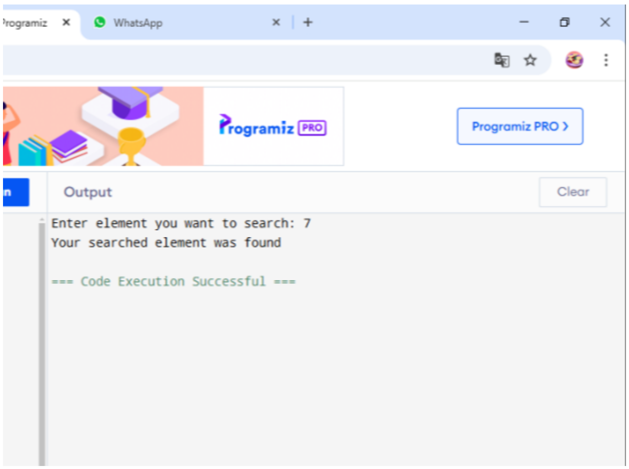
LEARNING OBJECTIVE:- learned regarding the concept of arrays and deletion program of the arrays.

AIM-3.- WRITE A PROGRAM IN C TO IMPLEMENT LINEAR SEARCHING IN 1-D ARRAYS.

PROGRAM-

```
#include <stdio.h>

int main() {
    int arr[8] = {1, 2, 3, 4, 5, 6, 7, 8};
    int search;
    printf("Enter element you want to search: ");
    scanf("%d", &search);
    int found = 0;


    for (int i = 0; i < 8; i++) {
        if (arr[i] == search) {
            found = 1;
            break;  // Exit the loop once the element is found
        }
    }


     if (found == 1) {
        printf("Your searched element was found");
    } else {
        printf("not found---->");
    }

    return 0;
```

```
}
```

OUTPUT- 

```
Enter element you want to search: 7
Your searched element was found

=== Code Execution Successful ===
```

LEARNING OBJECTIVE- learned regarding the concept of arrays and linear searching of the arrays.

AIM-4.- WRITE A PROGRAM TO IMPLEMENT SORTING IN 1-D ARRAYS.

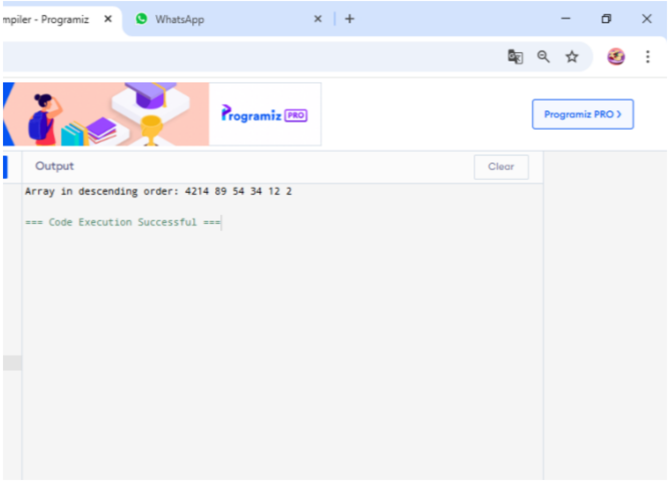
PROGRAM-

```
#include <stdio.h>

int main() {
```

```
int arr[6] = {34, 12, 54, 2,  89, 4214};


 int temp;



for (int i = 0; i < 6; i++) {
   for (int j = 0; j < 5; j++) {
      if (arr[j] < arr[j+1]) {

         temp = arr[j];
         arr[j] = arr[j+1];
         arr[j+1] = temp;
      }
   }
}



 printf("Array in descending order: ");
for (int i = 0; i < 6; i++) {
   printf("%d ", arr[i]);
```

```
    }

    return 0;
}
```

OUTPUT-



```
Array in descending order: 4214 89 54 34 12 2

=== Code Execution Successful ===
```

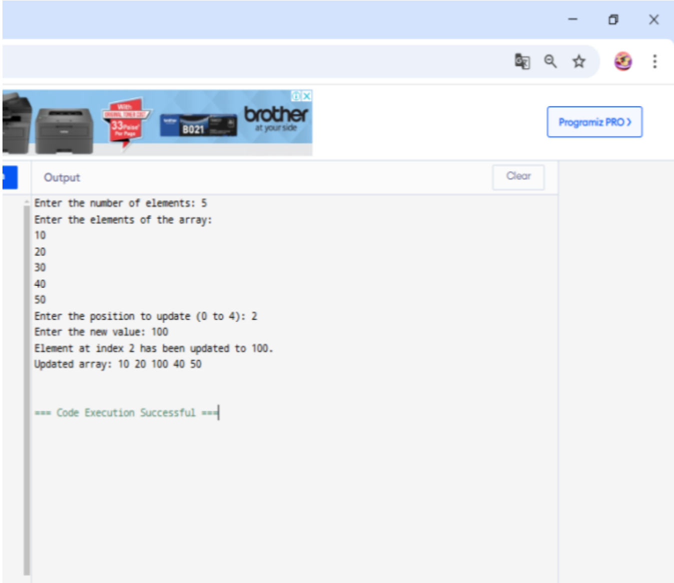
LEARNING OBJECTIVE- learned regarding the concept of arrays and sorting implementation of the arrays.

AIM-5.- WRITE A PROGRAM IN C TO IMPLEMENT UPDATION IN 1-D ARRAYS.

PROGRAM-

```
#include <stdio.h>

// Function to update an element at a given position
void updateElement(int arr[], int size, int pos, int new_value) {
    if (pos < 0 || pos >= size) {
        printf("Invalid position!\n");
    } else {
        arr[pos] = new_value;  // Update the element at the given position
        printf("Element at index %d has been updated to %d.\n", pos, new_value);
    }
}

// Function to print the array
```

```
void printArray(int arr[], int size) {
    printf("Updated array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int arr[100], n, pos, new_value;

    // Input the number of elements
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    // Input the elements of the array
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    // Ask for the position and new value to update
    printf("Enter the position to update (0 to %d): ", n - 1);
    scanf("%d", &pos);
    printf("Enter the new value: ");
    scanf("%d", &new_value);

    // Update the element at the specified position
    updateElement(arr, n, pos, new_value);

    // Print the updated array
    printArray(arr, n);

    return 0;
}
```

OUTPUT- 

```
Enter the number of elements: 5
Enter the elements of the array:
10
20
30
40
50
Enter the position to update (0 to 4): 2
Enter the new value: 100
Element at index 2 has been updated to 100.
Updated array: 10 20 100 40 50


=== Code Execution Successful ===
```

LEARNING OBJECTIVE- learned regarding the concept of arrays and updating implementation of the arrays.