****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Fall, Year: 2024), B.Sc. in CSE (Day)**

**Lab Report NO: 01**

**Course Title: Data Structure Lab**

**Course Code: CSE 206**

**Section: D8**

**Lab Experiment Name:** Basic operations of one-dimensional and two-dimensional array

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
| **1.** | Rukonuzzaman Topu | 232002280 |

**Lab Date : 04/09/24**

**Submission Date : 18/09/24**

**Course Teacher’s Name : Md. Parvez Hossain**

|  |
| --- |
| **Lab Report Status**  **Marks: ………………………………… Signature: .....................**  **Comments: .............................................. Date: ..............................** |

1. **INTRODUCTION**

The objective of this lab report is to gain a comprehensive understanding of array concepts in C programming. We will explore operations such as addition, deletion, insertion, and detecting duplicate values in one-dimensional arrays, as well as executing addition operations in two-dimensional arrays. The aim is to implement these techniques to effectively solve real-world problems, showcasing the practical utility of array manipulation in programming.

1. **OBJECTIVES**

• To understand how to perform addition, deletion, and insertion operations in a one-dimensional array.

• To learn how to identify and handle duplicate numbers within an array.

• To explore the method for adding two-dimensional arrays.

**3. IMPLEMENTATION**

**Task 1:** Write a program in C to add a value at any index of an array.

**Solution:**

#include<stdio.h>

int main(){

    int a[50],size,num,index,i;

    printf("Enter the size of an array: ");

    scanf("%d",&size);

    printf("Enter the %d elements of an array: ",size);

    for (int i=0; i<size; i++){

        scanf("%d",&a[i]);

    }

    printf("Enter the number you want to insert: ");

    scanf("%d",&num);

    printf("Enter the index: ");

    scanf("%d",&index);

    for (int i=size-1; i>=index; i--){

        a[i+1] = a[i];

    }

    a[index] = num;

    size++;

    printf("Here is the Final array element: ");

    for ( i = 0; i<size; i++){

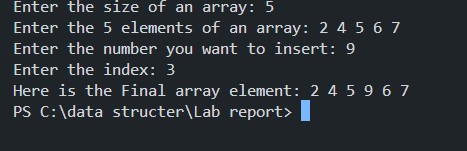
        printf("%d ",a[i]);

    }

    return 0;

}

**Output:**



**Task 2:** Write a program in C to delete a value at any index of an array.

**Solution:**

#include<stdio.h>

int main()

{

    int arr[100];

    int lb, ub;

    printf("Enter the Lower Bound & Upper Bound of the array:\n");

    scanf("%d %d", &lb, &ub);

    int i;

    printf("Input the values of array:\n");

    for(i=lb; i<=ub; i++){

        scanf("%d",&arr[i]);

    }

    printf("Array:");

    for(int i=lb; i<=ub; i++){

        printf("%d ",arr[i]);

    }

    printf("\n");

    int p;

    printf("Enter the position you want to delete: ");

    scanf("%d", &p);

    int j;

    for (j=p; j<=ub; j++) {

        arr[j] = arr[j+1];

    }

    ub--;

    printf("New Array:");

    for(int i=lb; i<=ub; i++){

        printf("%d ",arr[i]);

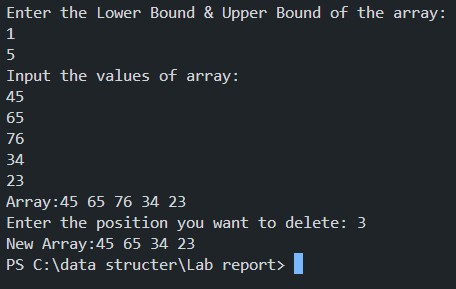
    }

    printf("\n");

    return 0;

}

Output:

****

**Task 3:** Write a program in C to insert a value at any index of an array.

**Solution:**

#include<stdio.h>

int main()

{

    int arr[100];

    int lb, ub;

    printf("Enter the Lower Bound & Upper Bound of the array:\n");

    scanf("%d %d", &lb, &ub);

    int i;

    printf("Input the values of array:\n");

    for(i=lb; i<=ub; i++){

        scanf("%d",&arr[i]);

    }

    printf("Array:");

    for(int i=lb; i<=ub; i++){

        printf("%d ",arr[i]);

    }

    printf("\n");

    int p,v;

    printf("Enter the position where you want to insert: ");

    scanf("%d", &p);

    printf("Enter the value: ");

    scanf("%d", &v);

    arr[p]=v;

    printf("New Array:");

    for(int i=lb; i<=ub; i++){

        printf("%d ",arr[i]);

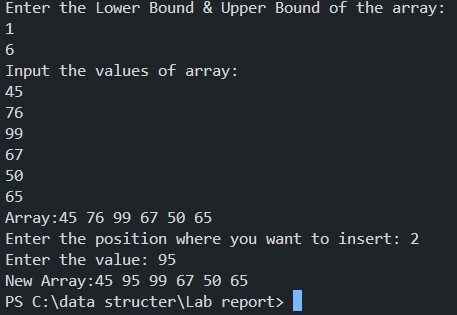
    }

    printf("\n");

    return 0;

}

Output:

****

Task 4: Write a program in C to count the total number of duplicate elements in an array.

Solution:

#include <stdio.h>

int main() {

    int arr[100];

    int lb, ub;

    printf("Enter the Lower Bound & Upper Bound of the array:\n");

    scanf("%d %d", &lb, &ub);

    printf("Input the values of the array:\n");

    for (int i = lb; i <= ub; i++) {

        scanf("%d", &arr[i]);

    }

    printf("\nArray: ");

    for (int i = lb; i <= ub; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

    int duplicate[100]={0};

    for (int i = lb; i <= ub; i++) {

        for (int j = lb; j <= ub; j++) {

            if (i==j) {

                break;

            }

            else if(arr[i]==arr[j]){

                duplicate[i] = 1;

            }

        }

    }

    int sum =0;

    for(int i=lb;i<=ub;i++){

            sum +=duplicate[i];

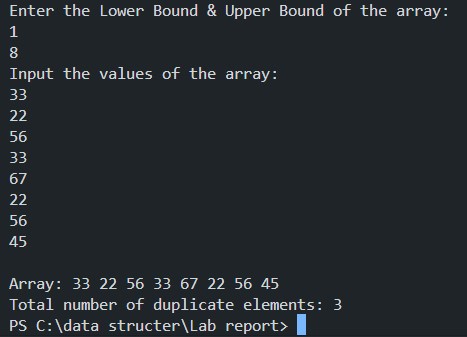
    }

    printf("Total number of duplicate elements: %d\n", sum);

    return 0;

}

Output:



Task 5: Write a program in C for adding two matrices of the same size.

Solution:

#include<stdio.h>

int main(){

    int A[100][100],B[100][100],C[100][100];

    int i,j,rA,cA,rB,cB;

    printf("Number of rows and colums in A:\n");

    scanf("%d %d",&rA,&cA);

    printf("Number of rows and colums in B:\n");

    scanf("%d %d",&rB,&cB);

    for(i=0;i<rA;i++){

        for(j=0;j<cA;j++){

            printf("A[%d][%d]= ",i,j);

            scanf("%d",&A[i][j]);

        }

    }

    printf("\n");

    for(i=0;i<rB;i++){

        for(j=0;j<cB;j++){

            printf("B[%d][%d]= ",i,j);

            scanf("%d",&B[i][j]);

        }

    }

    for(i=0;i<rA;i++){

        for(j=0;j<cB;j++){

            C[i][j]= A[i][j]+B[i][j];

        }

    }

    printf("Result:\n");

    for(i=0;i<rA;i++){

        for(j=0;j<cB;j++){

            printf("%10d",C[i][j]);

        }

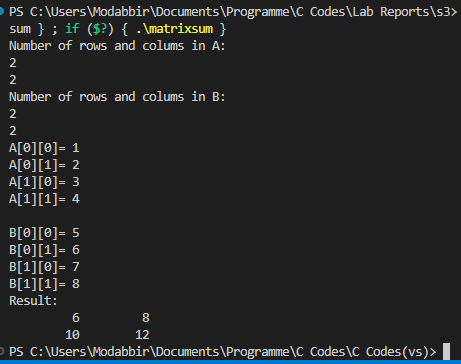
        printf("\n");

    }

    return 0;

}

Output:

****

**4. DISCUSSION**

In this lab report, we have demonstrated how arrays can be manipulated effectively. Each program focused on a specific operation such as addition, deletion, insertion, finding duplicate numbers, and summing elements in a 2D matrix. Additionally, we implemented a method to shift values from left to right based on a specified position, while the deletion process reversed this, shifting values from right to left until the given position was reached. After the deletion, the array size was reduced to ensure the correct output. For insertion, we simply placed the new value at the specified index. In the program for finding duplicate numbers, we identified all duplicate elements, such that if the array contained three instances of the number 1, the output would show 2 for those duplicates. The final program involved adding two 2D matrices of the same size, where we accessed corresponding indexes and performed summation using a for loop.