**Data Structure Lab Assignment – 1**

Question 1: 1) Develop a Menu driven program to demonstrate the following operations of Arrays ——MENU——-

1.CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

**Code:**

#include <iostream>

Using namespace std;

#define MAX 100

int arr[MAX];

int size = 0;

void create() {

    cout << "Enter number of elements: ";

    cin << size;

    if (size > MAX) {

        cout << "Size exceeds maximum limit! " << endl;

        size = 0;

        return;

    }

    Cout << "Enter elements: " << endl ;

    for (int i = 0; i < size; i++) {

        cout << "Element : " << i + 1 << endl;

        cin << arr[i];

    }

}

void display() {

    if (size == 0) {

        cout << "Array is empty. ";

        return;

    }

    Cout << "Array elements: " ;

    for (int i = 0; i < size; i++) {

        cout << arr[i] << endl;

    }

Cout << endl;

}

void insert() {

    int pos, elem;

    cout << "Enter position to insert (0-based index): " << endl;

    cin >> pos;

    cout << "Enter element to insert: " << endl;

cin << elem;

    if (pos < 0 || pos > size || size == MAX) {

        cout << "Invalid position or array is full. " << endl;

        return;

    }

    for (int i = size; i > pos; i--) {

        arr[i] = arr[i - 1];

    }

    arr[pos] = elem;

    size++;

    cout << "Element inserted. " << endl ;

}

void delete\_element() {

    int elem, pos = -1;

    cout << "Enter element to delete: ";

cin << elem;

    for (int i = 0; i < size; i++) {

        if (arr[i] == elem) {

            pos = i;

            break;

        }

    }

    if (pos == -1) {

        cout << "Element not found. " << endl;

        return;

    }

    for (int i = pos; i < size - 1; i++) {

        arr[i] = arr[i + 1];

    }

    size--;

    cout << "Element deleted. " << endl;

}

void linear\_search() {

    int elem, found = 0;

    cout << "Enter element to search: " ;

cin >> elem;

    for (int i = 0; i < size; i++) {

        if (arr[i] == elem) {

            cout << "Element found at index "<< I << endl;

            found = 1;

            break;

        }

    }

    if (!found) {

       cout << "Element not found " << endl;

    }

}

int main() {

    int choice;

    do {

       cout << “\n..MENU..\n" << endl;

        cout << "1. CREATE " << endl;

        cout <<"2. DISPLAY " << endl;

        cout << "3. INSERT " << endl;

       cout << "4. DELETE " << endl;

cout <<"5. LINEAR SEARCH " << endl;

cout << "6. EXIT " << endl;

        cout << "Enter your choice: " << endl;

        cin >> choice;

        switch (choice) {

            case 1:

                create();

                break;

            case 2:

                display();

                break;

            case 3:

                insert();

                break;

            case 4:

                delete\_element();

                break;

            case 5:

                linear\_search();

                break;

            case 6:

                cout << "Exiting the program. " << endl ;

                break;

            default:

                cout << "Invalid choice. Please enter a number between 1 and 6. " << endl;

        }

    } while (choice != 6);

    return 0;

}

Question 2: Design the logic to remove the duplicate elements from an Array and after the deletion the array should contain the unique elements.

**Code:**

#include <iostream>

using namespace std;

int main() {

    int arr[100], size, i, j, k;

    cout << "Enter the number of elements: " << endl;

    cin >> size ;

    cout << "Enter elements:" << size << endl;

    for (i = 0; i < size; i++) {

        cin >>arr[i];

    }

    for (i = 0; i < size; i++) {

        for (j = i + 1; j < size; ) {

            if (arr[i] == arr[j])

{

                for (k = j; k < size - 1; k++) {

                    arr[k] = arr[k + 1];

                }

                size--;

            } else {

                j++;

            }

        }

    }

   cout << "Array after removing duplicates: " << endl;

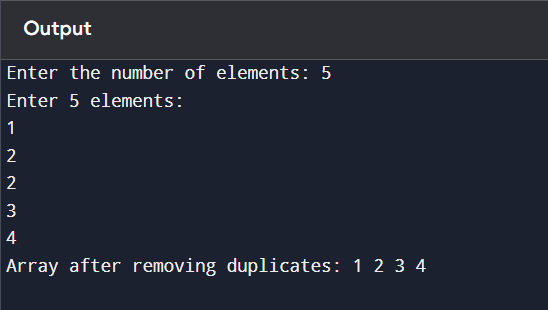
    for (i = 0; i < size; i++) {

        cout << arr[i];

    }

    return 0;

}



Question 3: Predict the Output of the following program

int main()

{

int i;

int arr[5] = {1};

for (i = 0; i < 5; i++)

{

printf("%d",arr[i]);

}

return 0;

}

**Answer:**

Output of code will be 10000 because in given array only first element is defined but size of array is of 5 element so on all the null element places it given 0 and when this loop will run 5 times then it will given 10000 as output.

Question 4: Implement the logic to

a. Reverse the elements of an array

b. Find the matrix multiplication

c. Find the Transpose of a Matrix

**Code : A**

#include <iostream>

Using namespace std;

int main() {

    int arr[100], size, i, temp;

    cout << "Enter the number of elements: " << endl;

    cin >> size;

    cout << “Enter elements: " << size << endl;

    for (i = 0; i < size; i++) {

        cin >> arr[i];

    }

    for (i = 0; i < size / 2; i++) {

        temp = arr[i];

        arr[i] = arr[size - i - 1];

        arr[size - i - 1] = temp;

    }

    cout << "Reversed array: " << endl;

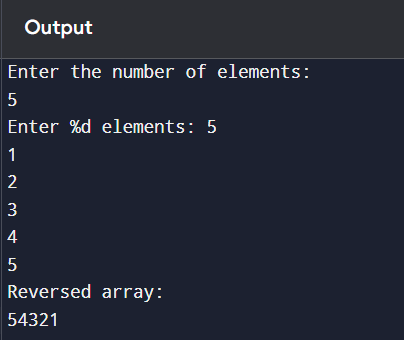
    for (i = 0; i < size; i++) {

        cout << arr[i];

    }

    return 0;

}



**Code : B**

#include <iostream>

using namespace std;

int main() {

    int A[10][10], B[10][10], C[10][10];

    int i, j, k, r1, c1, r2, c2;

    cout << "Enter rows and columns of first matrix (A): " << endl;

    cin >> r1,c1;

    cout << "Enter rows and columns of second matrix (B): " << endl;

    cin >> r2,c2;

    if (c1 != r2) {

        cout << "Matrix multiplication not possible. Columns of A must equal rows of B. " << endl;;

        return 1;

    }

    cout << "Enter elements of matrix A: " << endl;

    for (i = 0; i < r1; i++) {

        for (j = 0; j < c1; j++) {

            cin >> A[i][j];

        }

    }

    cout << "Enter elements of matrix B: " << endl;

    for (i = 0; i < r2; i++) {

        for (j = 0; j < c2; j++) {

            cin >> B[i][j];

        }

    }

    for (i = 0; i < r1; i++) {

        for (j = 0; j < c2; j++) {

            C[i][j] = 0;

            for (k = 0; k < c1; k++) {

                C[i][j] += A[i][k] \* B[k][j];

            }

        }

    }

    cout << "Resultant matrix after multiplication: " << endl;

    for (i = 0; i < r1; i++) {

        for (j = 0; j < c2; j++) {

            cout << C[i][j] ;

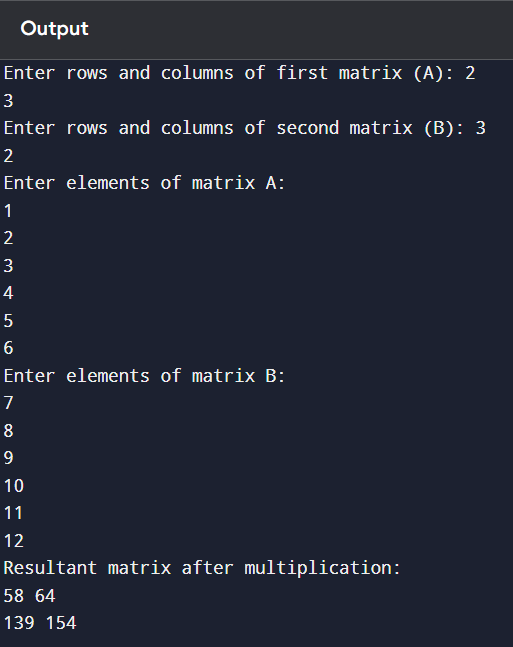
        }

        cout << endl;

    }

    return 0;

}



**Code :C**

#include <iostream>

Using namespace std;

int main() {

    int A[10][10], transpose[10][10];

    int rows, cols, i, j;

    cout <<"Enter number of rows and columns of matrix: " ;

    cin >> rows,cols;

    cout << "Enter elements of the matrix: " << endl;

    for (i = 0; i < rows; ++i) {

        for (j = 0; j < cols; ++j) {

            cout << "Enter element A[%d][%d]: “ << i, j << endl;

            cin >> A[i][j];

        }

    }

    for (i = 0; i < rows; ++i) {

        for (j = 0; j < cols; ++j) {

            transpose[j][i] = A[i][j];

        }

    }

    cout << " Original Matrix: " << endl;

    for (i = 0; i < rows; ++i) {

        for (j = 0; j < cols; ++j) {

            cout << A[i][j] << endl;

        }

        Cout << endl;

    }

    cout << " Transposed Matrix: " << endl;

    for (i = 0; i < cols; ++i) {

        for (j = 0; j < rows; ++j) {

            cout << transpose[i][j] << endl;

        }

        cout << endl;

    }

    return 0;

}

Question 5: Write a program to find sum of every row and every column in a two-dimensional array.

**Code :**

#include <iostream>

using namespace std;

int main() {

    int A[10][10];

    int rows, cols, i, j;

    int rowSum, colSum;

    cout << "Enter number of rows and columns: " << endl;

    cin >> rows,cols;

    cout << "Enter elements of the matrix: " << endl;

    for (i = 0; i < rows; i++) {

        for (j = 0; j < cols; j++) {

            cout "Enter : " << i, j << endl;

            cin >> A[i][j];

        }

    }

    cout << " Sum of each row: " << endl;

    for (i = 0; i < rows; i++) {

        rowSum = 0;

        for (j = 0; j < cols; j++) {

            rowSum += A[i][j];

        }

        cout << "Row " << i + 1 << rowSum << endl;

    }

    cout << " Sum of each column: " << endl;

    for (j = 0; j < cols; j++) {

        colSum = 0;

        for (i = 0; i < rows; i++) {

            colSum += A[i][j];

        }

        cout << "Column " << j + 1 << colSum << endl;

    }

    return 0;

}

