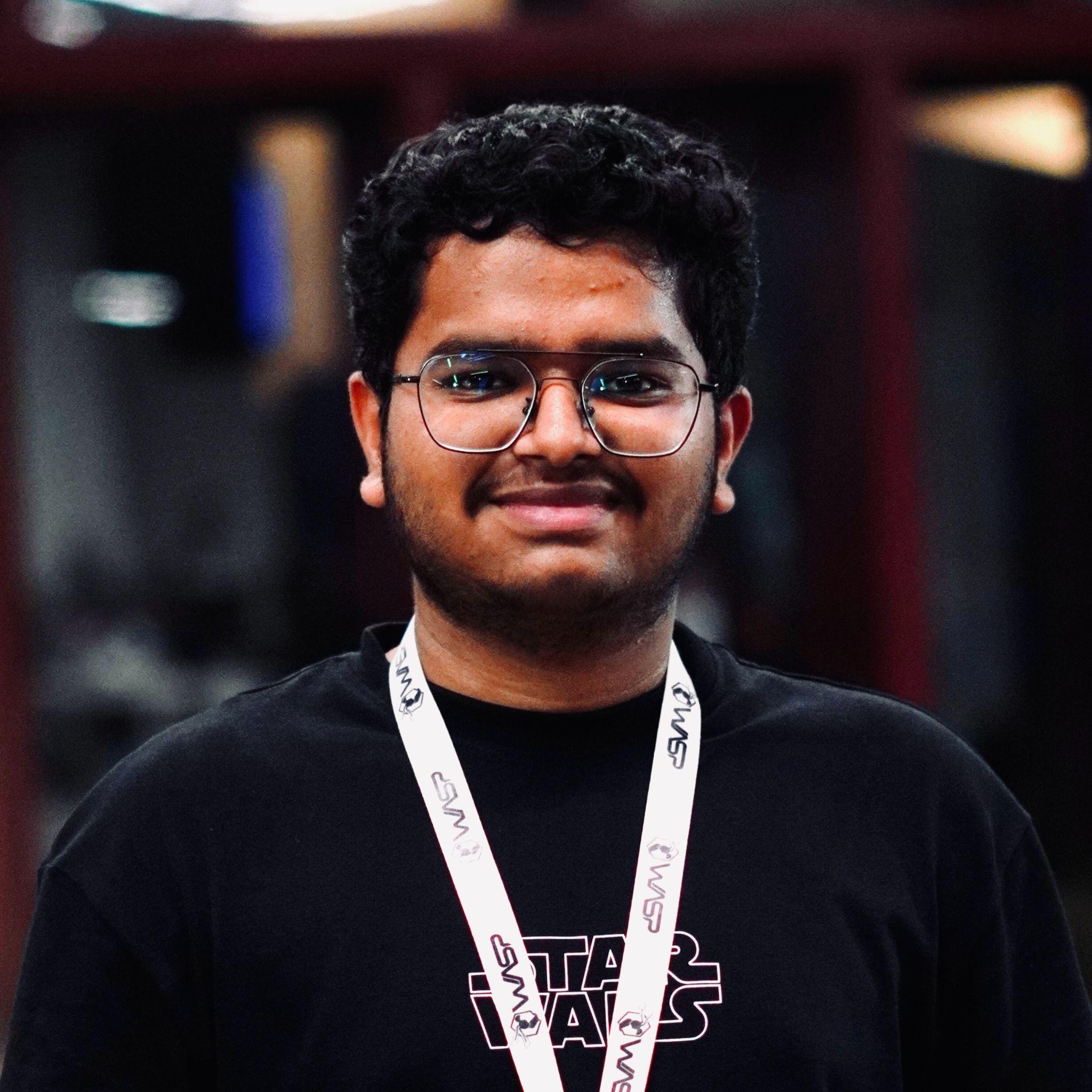
Name: Dhruv Mittal

Roll no: 1024030364

Subgroup: 2C24



Q-2: /\* 2) Design the logic to remove the duplicate elements from an Array and after the deletion the array should contain the unique elements. \*/

#include <iostream>

using namespace std ;

void remove\_duplicates(int arr[] , int size){

for(int i=0 ; i<size ; i++){

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

if(arr[i] == arr[j]){

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

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

}

size-- ;

j-- ;

}

}

}

cout << "Array after deletion : " ;

for(int i=0 ; i<size ; i++){

cout << arr[i] << " " ;

}

}

int main(){

int x ;

cout << "Enter the size of array you want : " ;

cin >> x ;

int arr[x];

cout << "Enter the array : " ;

for(int i=0 ; i<x ; i++){

cin >> arr[i] ;

}

remove\_duplicates(arr , x) ;

return 0 ;

}

Q-3: /\*

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;

}

\*/

// output will be : 10000

Q-4a: /\*

4) Implement the logic to

a. Reverse the elements of an array

\*/

#include <iostream>

using namespace std ;

void reverse\_array(int arr[] , int size){

int start = 0;

int end = size - 1 ;

while(start <= end){

swap(arr[start] , arr[end]) ;

start++ ;

end -- ;

}

cout << "Reversed array is : " ;

for(int i=0 ; i<size ; i++){

cout << arr[i] << " " ;

}

}

int main(){

int n ;

cout << "Enter the size of the array you want : " ;

cin >> n ;

int arr[n] ;

cout << "Enter the array you want : " ;

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

cin >> arr[i] ;

}

reverse\_array(arr , n) ;

return 0 ;

}

Q-4b : /\*

4) Implement the logic to :

b) Find the matrix multiplication

\*/

#include <iostream>

using namespace std;

const int MAX = 100;

void inputMatrix(int A[][MAX], int B[][MAX], int m, int n, int p) {

cout << "Enter elements of Matrix A (" << m << " x " << n << "):\n";

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

cin >> A[i][j];

cout << "Enter elements of Matrix B (" << n << " x " << p << "):\n";

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

for (int j = 0; j < p; j++)

cin >> B[i][j];

}

void multiplyMatrix(int A[][MAX], int B[][MAX], int C[][MAX], int m, int n, int p) {

for (int i = 0; i < m; i++){

for (int j = 0; j < p; j++){

C[i][j]= 0;

for (int k = 0; k < n; k++){

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

}}}

}

void printMatrix(int A[][MAX], int rows, int cols) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++)

cout << A[i][j] << " ";

cout << endl;

}

}

int main() {

int m, n, p;

cout << "Enter rows and columns of Matrix A: ";

cin >> m >> n;

cout << "Enter number of columns of Matrix B: ";

cin >> p;

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

inputMatrix(A, B, m, n, p);

multiplyMatrix(A, B, C, m, n, p);

cout << "\nMatrix A:\n";

printMatrix(A, m, n);

cout << "\nMatrix B:\n";

printMatrix(B, n, p);

cout << "\nMatrix C (Result of A x B):\n";

printMatrix(C, m, p);

return 0;

}

Q-4c : /\*

c. Find the Transpose of a Matrix

\*/

#include<iostream>

using namespace std;

const int maxSize = 100;

void inputarray(int arr1[][maxSize], int rows, int cols) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cin >> arr1[i][j];

}

}

}

void transposematrix(int arr1[][maxSize], int arr2[][maxSize], int rows, int cols) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

arr2[j][i] = arr1[i][j];

}

}

}

void printarray(int arr[][maxSize], int rows, int cols) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cout << arr[i][j] << " ";

}

cout << endl;

}

}

int main() {

int rows, cols;

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

cin >> rows >> cols;

if (rows > maxSize || cols > maxSize) {

cout << "Matrix size exceeds allowed limit." << endl;

return 1;

}

int arr1[maxSize][maxSize], arr2[maxSize][maxSize];

inputarray(arr1, rows, cols);

transposematrix(arr1, arr2, rows, cols);

cout << "Transposed matrix: " << endl;

printarray(arr2, cols, rows);

return 0;

}

Q-5 : /\*

5) Write a program to find sum of every row and every column in a two-dimensional

array.

\*/

#include<iostream>

using namespace std;

const int maxSize = 100;

void inputArray(int arr[][maxSize], int rows, int cols) {

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

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cin >> arr[i][j];

}

}

}

void sumRows(int arr[][maxSize], int rows, int cols) {

cout << "\nSum of each row:" << endl;

for (int i = 0; i < rows; i++) {

int rowSum = 0;

for (int j = 0; j < cols; j++) {

rowSum += arr[i][j];

}

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

}

}

void sumCols(int arr[][maxSize], int rows, int cols) {

cout << "\nSum of each column:" << endl;

for (int j = 0; j < cols; j++) {

int colSum = 0;

for (int i = 0; i < rows; i++) {

colSum += arr[i][j];

}

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

}

}

int main() {

int rows, cols;

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

cin >> rows >> cols;

if (rows > maxSize || cols > maxSize) {

cout << "Matrix size exceeds the maximum allowed size." << endl;

return 1;

}

int arr[maxSize][maxSize];

inputArray(arr, rows, cols);

sumRows(arr, rows, cols);

sumCols(arr, rows, cols);

return 0;

}