|  |  |
| --- | --- |
| ASSIGNMENT 1 Solutions | Ishaan Mishra    1024030346 |

Lab Assignment 1 Solutions

# Question 1

// Question 1: Menu driven program for array operations  
#include <iostream>

using namespace std;

void createArray(int arr[], int &n) {

cout << "Enter number of elements (max 6): ";

cin >> n;

if (n > 6) n = 6;

cout << "Enter " << n << " elements: ";

for (int i = 0; i < n; i++) cin >> arr[i];

}

void displayArray(int arr[], int n) {

if (n == 0) {

cout << "Array is empty\n";

return;

}

cout << "Array elements: ";

for (int i = 0; i < n; i++) cout << arr[i] << " ";

cout << endl;

}

void insertElement(int arr[], int &n) {

if (n == 6) {

cout << "Array is full\n";

return;

}

int pos, val;

cout << "Enter position (1 to " << n+1 << "): ";

cin >> pos;

if (pos < 1 || pos > n+1) {

cout << "Invalid position!\n";

return;

}

cout << "Enter value: ";

cin >> val;

for (int i = n; i >= pos; i--) arr[i] = arr[i-1];

arr[pos-1] = val;

n++;

cout << "Element inserted.\n";

}

void deleteElement(int arr[], int &n) {

if (n == 0) {

cout << "Array is empty!\n";

return;

}

int pos;

cout << "Enter position (1 to " << n << "): ";

cin >> pos;

if (pos < 1 || pos > n) {

cout << "Invalid position!\n";

return;

}

for (int i = pos-1; i < n-1; i++) arr[i] = arr[i+1];

n--;

cout << "Element deleted.\n";

}

void linearSearch(int arr[], int n) {

if (n == 0) {

cout << "Array is empty!\n";

return;

}

int key, found = 0;

cout << "Enter value to search: ";

cin >> key;

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

if (arr[i] == key) {

cout << "Element found at position " << i+1 << endl;

found = 1;

break;

}

}

if (!found) cout << "Element not found.\n";

}

int main() {

int arr[6], n = 0, choice;

do {

cout << "\n------ MENU ------\n";

cout << "1. CREATE\n";

cout << "2. DISPLAY\n";

cout << "3. INSERT\n";

cout << "4. DELETE\n";

cout << "5. LINEAR SEARCH\n";

cout << "6. EXIT\n";

cout << "Enter choice: ";

cin >> choice;

switch (choice) {

case 1: createArray(arr, n); break;

case 2: displayArray(arr, n); break;

case 3: insertElement(arr, n); break;

case 4: deleteElement(arr, n); break;

case 5: linearSearch(arr, n); break;

case 6: cout << "Exiting...\n"; break;

default: cout << "Invalid choice!\n";

}

} while (choice != 6);

return 0;

}

# Question 2

// Question 2: Remove duplicates from array  
#include <iostream>

using namespace std;

int main() {

int arr[50], n;

cout << "Enter size of array: ";

cin >> n;

cout << "Enter elements: ";

for (int i = 0; i < n; i++) cin >> arr[i];

int temp[50;

int j = 0;

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

bool duplicate = false;

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

if (arr[i] == temp[k]) {

duplicate = true;

break;

}

}

if (!duplicate) {

temp[j] = arr[i];

j++;

}

}

cout << "Array after removing duplicates: ";

for (int i = 0; i < j; i++) cout << temp[i] << " ";

cout << endl;

return 0;

}

# Question 3

// Question 3: Predict Output  
  
output: 10000

# Question 4

#include <iostream>

using namespace std;

void reverseArray() {

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

int n = 5;

cout << "Reversed array: ";

for (int i = n - 1; i >= 0; i--) {

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

}

cout << endl;

}

void matrixMultiply() {

int a[2][2] = {{1, 2}, {3, 4}};

int b[2][2] = {{5, 6}, {7, 8}};

int c[2][2] = {0}; // initialize with 0

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

for (int k = 0; k < 2; k++) {

c[i][j] += a[i][k] \* b[k][j];

}

}

}

cout << "Result of multiplication:\n";

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

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

}

cout << endl;

}

}

void matrixTranspose() {

int a[2][3] = {{1, 2, 3}, {4, 5, 6}};

int r = 2, c = 3;

cout << "Transpose of matrix:\n";

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

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

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

}

cout << endl;

}

}

int main() {

int choice;

while (true) {

cout << "\nMenu:\n";

cout << "1. Reverse Array\n";

cout << "2. Matrix Multiplication\n";

cout << "3. Matrix Transpose\n";

cout << "4. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: reverseArray(); break;

case 2: matrixMultiply(); break;

case 3: matrixTranspose(); break;

case 4: return 0;

default: cout << "Invalid choice\n";

}

}

}

# Question 5

// Question 5: Row and Column Sum  
#include <iostream>

using namespace std;

int main(){

int A[4][4]={{3,5,7,2},{0,4,6,3},{5,4,6,5},{4,1,9,0}};

int sum,i,j;

for(i=0;i<4;i++){

sum=0;

for(j=0;j<4;j++){

sum+=A[i][j];

}

cout << "sum(Row " << i+1 << ") = " << sum << endl;

}

for(i=0;i<4;i++){

sum=0;

for(j=0;j<4;j++){

sum+=A[j][i];

}

cout << "sum(Colmun " << i+1 << ") = " << sum << endl;

}

return 0;

}