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

#include <iostream>

using namespace std;

int arr[100];

int arrSize = 0;

const int capacity = 100;

void createArray() {

int n;

cout << "\nEnter the number of elements: ";

cin >> n;

if (n > capacity) {

cout << "Array size exceeds capacity of " << capacity << endl;

return;

}

arrSize = n;

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

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

cin >> arr[i];

}

cout << "Array created successfully!" << endl;

}

void displayArray() {

if (arrSize == 0) {

cout << "\nArray is empty!" << endl;

return;

}

cout << "\nArray elements: ";

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

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

}

cout << endl;

cout << "Array size: " << arrSize << endl;

}

void insertElement() {

if (arrSize >= capacity) {

cout << "\nArray is full! Cannot insert more elements." << endl;

return;

}

int element, position;

cout << "\nEnter element to insert: ";

cin >> element;

cout << "Enter position (0 to " << arrSize << "): ";

cin >> position;

if (position < 0 || position > arrSize) {

cout << "Invalid position!" << endl;

return;

}

// Shift elements to the right

for (int i = arrSize; i > position; i--) {

arr[i] = arr[i - 1];

}

arr[position] = element;

arrSize++;

cout << "Element inserted successfully!" << endl;

}

void deleteElement() {

if (arrSize == 0) {

cout << "\nArray is empty! Cannot delete element." << endl;

return;

}

int choice;

cout << "\nDelete by:" << endl;

cout << "1. Position" << endl;

cout << "2. Value" << endl;

cout << "Enter choice: ";

cin >> choice;

if (choice == 1) {

int position;

cout << "Enter position (0 to " << (arrSize - 1) << "): ";

cin >> position;

if (position < 0 || position >= arrSize) {

cout << "Invalid position!" << endl;

return;

}

for (int i = position; i < arrSize - 1; i++) {

arr[i] = arr[i + 1];

}

arrSize--;

cout << "Element at position " << position << " deleted successfully!" << endl;

}

else if (choice == 2) {

int value;

cout << "Enter value to delete: ";

cin >> value;

int position = -1;

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

if (arr[i] == value) {

position = i;

break;

}

}

if (position != -1) {

for (int i = position; i < arrSize - 1; i++) {

arr[i] = arr[i + 1];

}

arrSize--;

cout << "Element " << value << " deleted successfully!" << endl;

} else {

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

}

}

else {

cout << "Invalid choice!" << endl;

}

}

void linearSearch() {

if (arrSize == 0) {

cout << "\nArray is empty!" << endl;

return;

}

int target;

cout << "\nEnter element to search: ";

cin >> target;

bool found = false;

cout << "Element " << target;

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

if (arr[i] == target) {

if (!found) {

cout << " found at position(s): ";

found = true;

}

cout << i << " ";

}

}

if (found) {

cout << endl;

} else {

cout << " not found!" << endl;

}

}

void displayMenu() {

cout << "\n====== MENU ======" << 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 << "==================" << endl;

cout << "Enter your choice: ";

}

int main() {

int choice;

cout << "Array Operations Program" << endl;

cout << "========================" << endl;

do {

displayMenu();

cin >> choice;

switch (choice) {

case 1:

createArray();

break;

case 2:

displayArray();

break;

case 3:

insertElement();

break;

case 4:

deleteElement();

break;

case 5:

linearSearch();

break;

case 6:

cout << "\nThank you for using Array Operations Program!" << endl;

break;

default:

cout << "\nInvalid choice! Please try again." << endl;

}

} while (choice != 6);

return 0;

}

OUTPUT :

Array Operations Program

========================

====== MENU ======

1. CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

==================

Enter your choice: 1

Enter the number of elements: 4

Enter 4 elements: 1

2

3

4

Array created successfully!

====== MENU ======

1. CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

==================

Enter your choice: 2

Array elements: 1 2 3 4

Array size: 4

====== MENU ======

1. CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

==================

Enter your choice: 3

Enter element to insert: 5

Enter position (0 to 4): 3

Element inserted successfully!

====== MENU ======

1. CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

==================

Enter your choice: 2

Array elements: 1 2 3 5 4

Array size: 5

====== MENU ======

1. CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

==================

Enter your choice: 4

Delete by:

1. Position

2. Value

Enter choice: 2

Enter value to delete: 2

Element 2 deleted successfully!

====== MENU ======

1. CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

==================

Enter your choice: 2

Array elements: 1 3 5 4

Array size: 4

====== MENU ======

1. CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

==================

Enter your choice: 5

Enter element to search: 4

Element 4 found at position(s): 3

====== MENU ======

1. CREATE

2. DISPLAY

3. INSERT

4. DELETE

5. LINEAR SEARCH

6. EXIT

==================

Enter your choice: 6

Thank you for using Array Operations Program!