2. Searching Techniques:

**Linear and Binary Search**

#include<iostream.h> #include<conio.h>

int linearSearch(int arr[], int n, int key) { for (int i = 0; i < n; i++) {

if (arr[i] == key) {

return i; // Return the index where the key is found

}}

return -1; // Return -1 if the key is not found

}

int binarySearch(int arr[], int low, int high, int key) { while (low <= high) {

int mid = (low + high) / 2; if (arr[mid] == key) {

return mid; // Return the index where the key is found

} else if (arr[mid] < key) { low = mid + 1;

} else {

high = mid - 1;

}}

return -1; // Return -1 if the key is not found

}

void displayResult(int index) { if (index != -1) {

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

} else {

cout << "Element not found in the array." << endl;

}}

int main() { clrscr(); int n;

cout << "Enter the number of elements in the array: "; cin >> n;

int arr[100];

cout << "Enter the elements of the sorted array: ";

Page no. 12

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

}

int choice, key; while (1) {

cout << "\nChoose Search Algorithm:\n"; cout << "1. Linear Search\n";

cout << "2. Binary Search\n";

cout << "Enter your choice (1 or 2): "; cin >> choice;

switch (choice) { case 1:

cout << "Enter the key to search: "; cin >> key;

displayResult(linearSearch(arr, n, key)); break;

case 2:

cout << "Enter the key to search: "; cin >> key;

displayResult(binarySearch(arr, 0, n - 1, key)); break;

default:

cout << "Invalid choice!"; return 1;

}

char cont;

cout << "\nDo you want to continue? (y/n): "; cin >> cont;

if (cont != 'y' && cont != 'Y') { break;

}}

getch(); return 0;

}

Page no. 13