|  |
| --- |
| **SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES** |
| **COMPUTER SCIENCE AND ENGINEERING PROGRAMME** |

**SUB CODE: CSA0392 SUB NAME: Data Structures for Hashing Techniques**

**LIST OF PROGRAMS**

**DAY 3 : 27.07.2024**

**Lab Questions to be practiced with test cases**

1. Write a C program to implement Linear Search Algorithm.

Answer:

#include <stdio.h>

// Function to perform linear search

int linearSearch(int arr[], int size, int target) {

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

if (arr[i] == target) {

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

}

}

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

}

int main() {

int size, target;

// Input size of the array

printf("Enter the number of elements in the array: ");

scanf("%d", &size);

int arr[size]; // Declare an array of the specified size

// Input elements of the array

printf("Enter %d elements:\n", size);

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

scanf("%d", &arr[i]);

}

// Input the target value to search

printf("Enter the element to search for: ");

scanf("%d", &target);

// Perform linear search

int result = linearSearch(arr, size, target);

// Output result

if (result != -1) {

printf("Element %d found at index %d.\n", target, result);

} else {

printf("Element %d not found in the array.\n", target);

}

return 0;

}

1. Write a C program to implement Binary Search Algorithm.

Answer:

#include <stdio.h>

// Function to perform binary search

int binarySearch(int arr[], int size, int target) {

int left = 0;

int right = size - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

// Check if target is present at mid

if (arr[mid] == target) {

return mid; // Target found at index mid

}

// If target is greater, ignore left half

if (arr[mid] < target) {

left = mid + 1;

}

// If target is smaller, ignore right half

else {

right = mid - 1;

}

}

// Target is not present in array

return -1;

}

int main() {

int size, target;

// Input size of the array

printf("Enter the number of elements in the sorted array: ");

scanf("%d", &size);

int arr[size]; // Declare an array of the specified size

// Input elements of the array

printf("Enter %d elements (sorted in ascending order):\n", size);

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

scanf("%d", &arr[i]);

}

// Input the target value to search

printf("Enter the element to search for: ");

scanf("%d", &target);

// Perform binary search

int result = binarySearch(arr, size, target);

// Output result

if (result != -1) {

printf("Element %d found at index %d.\n", target, result);

} else {

printf("Element %d not found in the array.\n", target);

}

return 0;

}