Name: ANYA AGARWAL

Section: B (13)

Roll no: 2021122

1.Given an array of nonnegative integers, design a linear algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int linear­\_search(const vector<int>& arr, int x, int& count) {

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

count++;

if (arr[i] == x)

return i;

}

return -1;

}

int main() {

ifstream inputFile("input.txt");

if (!inputFile) {

cerr << "Error opening input file." << endl;

return 1;

}

int T;

inputFile >> T;

while (T--) {

int n = 0, count = 0;

inputFile >> n;

vector<int> arr(n);

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

inputFile >> arr[i];

}

int x = 0;

inputFile >> x;

int ans = -1;

// Linear search

ans = linear\_search(arr, x, count);

ofstream outputFile("output.txt", std::ios::app);

if (!outputFile) {

cerr << "Error opening output file." << endl;

return 1;

}

if (ans == -1)

outputFile << "Not present " << count << endl;

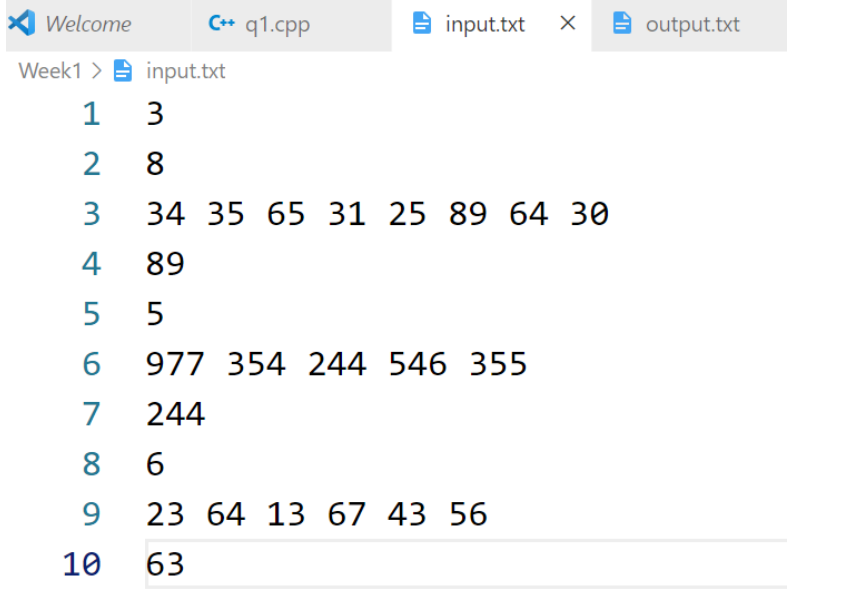
else

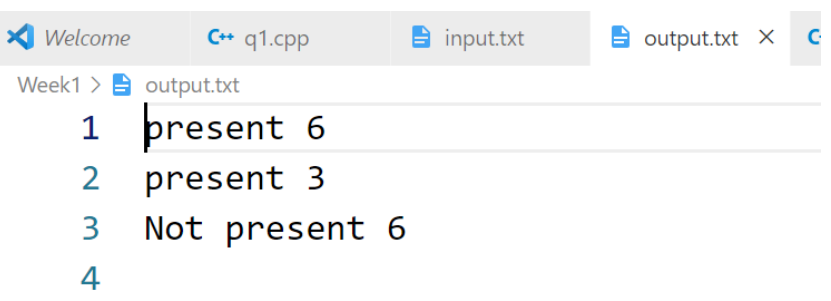
outputFile << "present " << count << endl;

}

return 0;

}





Name: ANYA AGARWAL

Section: B (13)

Roll no: 2021122

2.Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case.

(Time Complexity = O(n logn), where n is the size of input).

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int binary\_search\_file(ifstream& inputFile, int x, int n) {

vector<int> arr(n);

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

inputFile >> arr[i];

}

int low = 0;

int high = n - 1;

int comparison = 0;

while (low <= high) {

int mid = low + (high - low) / 2;

comparison++;

if (arr[mid] == x)

return mid;

else if (arr[mid] < x)

low = mid + 1;

else

high = mid - 1;

}

return -1;

}

int main() {

ifstream inputFile("input.txt");

if (!inputFile) {

cerr << "Error opening input file." << endl;

return 1;

}

int T;

inputFile >> T;

while (T--) {

int n, x;

inputFile >> n;

inputFile >> x;

int result = binary\_search\_file(inputFile, x, n);

ofstream outputFile("output.txt", std::ios::app);

if (!outputFile) {

cerr << "Error opening output file." << endl;

return 1;

}

if (result == -1)

outputFile << "Not present ";

else

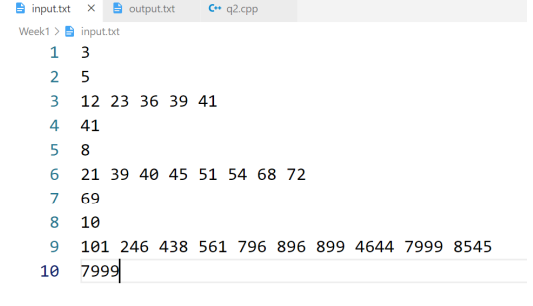
outputFile << "Present ";

}

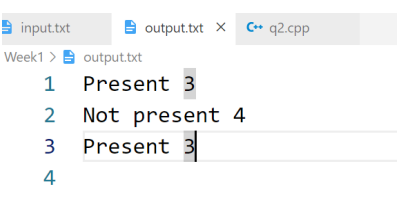
return 0;

}

INPUT



OUTPUT



Name: ANYA AGARWAL

Section: B (13)

Roll no: 2021122

3.Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether a given key element is present in the sorted array or not. For an array arr[n], search at the indexes arr[0], arr[2], arr[4],.....,arr[2k] and so on. Once the interval (arr[2k] < key < arr[ 2k+1] ) is found, perform a linear search operation from the index 2k to find the element key. (Complexity < O(n), where n is the number of elements need to be scanned for searching).

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int jumpSearch(vector<int> vtr, int element, int &comp) {

int i = 0, count = 0;

while (true) {

comp++;

if (vtr[i] >= element || i >= vtr.size()) {

if (vtr[i] == element) {

return i;

}

comp--;

break;

}

if (i == 0) {

i += 2;

} else {

i += i;

}

}

int j = i, mid;

i /= 2;

if (j >= vtr.size()) {

j = vtr.size() - 1;

}

while (i <= j) {

mid = i + (j - i) / 2;

count++;

if (vtr[mid] == element) {

return count;

} else if (vtr[mid] < element) {

i = mid + 1;

} else {

j = mid - 1;

}

}

return -1;

}

int main() {

ifstream inputFile("input.txt");

if (!inputFile) {

cerr << "Error opening input file." << endl;

return 1;

}

int T;

inputFile >> T;

while (T--) {

int n = 0, comparison = 0;

inputFile >> n;

vector<int> arr(n);

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

inputFile >> arr[i];

}

int x = 0;

inputFile >> x;

int result = jumpSearch(arr, x, comparison);

ofstream outputFile("output.txt", std::ios::app);

if (!outputFile) {

cerr << "Error opening output file." << endl;

return 1;

}

if (result == -1) {

outputFile << "Not present " << comparison << endl;

} else {

outputFile << "Present " << comparison << endl;

}

}

return 0;

}

INPUT

A number on a white background

Description automatically generated

OUTPUT

A black text on a white background

Description automatically generated