Name: ANYA AGARWAL

Section: B (13)

Roll no: 2021122

1. Given a sorted array of positive integers containing few duplicate elements, design an algorithm and implement it using a program to find whether the given key element is present in the array or not. If present, then also find the number of copies of given key. (Time Complexity = O (log n))

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

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;

inputFile >> n;

vector<int> arr(n);

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

inputFile >> arr[i];

}

int x = 0;

inputFile >> x;

int low = 0;

int high = n - 1;

int flag = 0;

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

while (low <= high) {

if (arr[mid] == x && (mid == 0 || arr[mid - 1] != x)) {

flag = 1;

break;

} else if (arr[mid] >= x) {

high = mid - 1;

} else {

low = mid + 1;

}

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

}

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

if (!outputFile) {

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

return 1;

}

if (flag == 0) {

outputFile << "Not present" << endl;

} else {

int startingIndex = mid;

int start = mid;

int end = n - 1;

while (start < end) {

if (arr[mid] == x && (mid == n - 1 || arr[mid + 1] != x)) {

end = mid;

break;

} else if (arr[mid] <= x) {

start = mid + 1;

} else {

end = mid - 1;

}

mid = start + (end - start) / 2;

}

outputFile << "present " << end - startingIndex + 1 << endl;

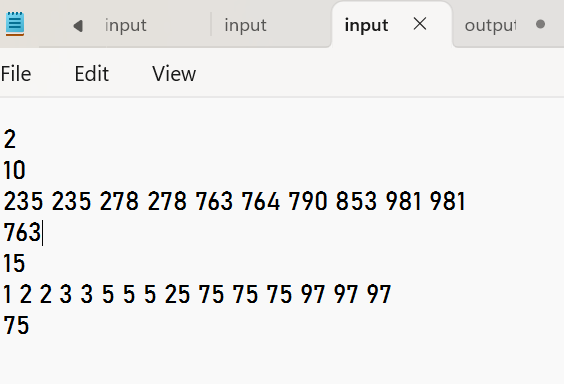
}

}

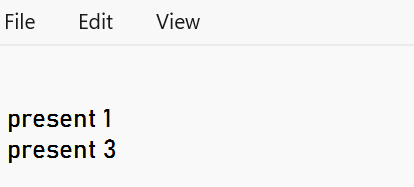
return 0;

}

INPUT



OUTPUT



Name: ANYA AGARWAL

Section: B (13)

Roll no: 2021122

2. Given a sorted array of positive integers, design an algorithm and implement it using a program to find three indices i, j, k such that arr[i] + arr[j] = arr[k].

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

ifstream inputFile("input.txt");

ofstream outputFile("output.txt");

if (!inputFile || !outputFile) {

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

return 1;

}

int T;

inputFile >> T;

while (T--) {

int n;

inputFile >> n;

vector<int> arr(n);

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

inputFile >> arr[i];

}

int i, j, k;

bool found = false;

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

for (j = i + 1; j < n; j++) {

for (k = j + 1; k < n; k++) {

if (arr[i] + arr[j] == arr[k]) {

found = true;

break;

}

}

if (found) {

break;

}

}

if (found) {

break;

}

}

if (!outputFile) {

cerr << "Error writing to output file." << endl;

return 1;

}

if (!found) {

outputFile << "No sequence found." << endl;

} else {

outputFile << i + 1 << "," << j + 1 << "," << k + 1 << endl;

}

}

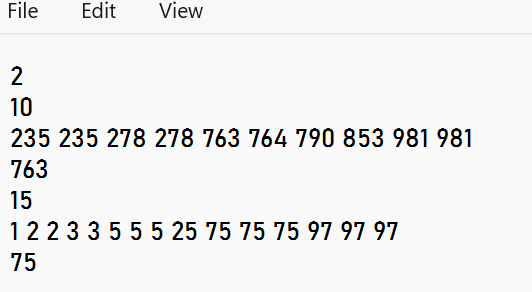
inputFile.close();

outputFile.close();

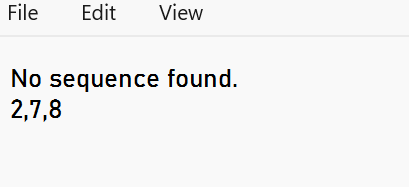
return 0;

}

INPUT



OUTPUT



Name: ANYA AGARWAL

Section: B (13)

Roll no: 2021122

3. Given an array of nonnegative integers, design an algorithm and a program to count the number of pairs of integers such that their difference is equal to a given key, K.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

ifstream inputFile("input.txt");

ofstream outputFile("output.txt");

if (!inputFile || !outputFile) {

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

return 1;

}

int T;

inputFile >> T;

while (T--) {

int n;

inputFile >> n;

vector<int> arr(n);

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

inputFile >> arr[i];

}

int key;

inputFile >> key;

int count = 0;

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

for (int j = i + 1; j < n; j++) {

int diff = arr[i] - arr[j];

if (diff < 0) {

diff = -diff;

}

if (diff == key) {

count++;

}

}

}

if (!outputFile) {

cerr << "Error writing to output file." << endl;

return 1;

}

outputFile << count << endl;

}

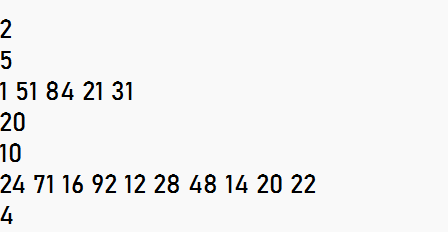
inputFile.close();

outputFile.close();

return 0;

}

INPUT



OUTPUT

