**PROJECT REPORT(CAN204)**

*A report submitted in partial fulfilment of the requirement for the course*

# COMPUTER

# ALGORITHMS

Part of the degree of

BACHELOR

# IN

# Computer Applications



**Session 2024-25**

**Submitted to:**

VAIBHAV PADHAYE

ASSISTANT PROFESSOR

**Submitted by:**

ARYAN

1000020005 B/P1

# SCHOOL OF COMPUTING DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

**Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.**

**ACKNOWLEDGEMENT**

I take this opportunity with much pleasure to thank all the people who have helped me through the course of my journey towards producing this project. I sincerely thank my project guide, Vaibhav Padhaye , for her guidance, help and motivation. Apart from the subject of my research, I learnt a lot from her, which I am sure will be useful in different stages of my life

I am especially grateful to my colleagues for their assistance, criticisms and useful insights. I am thankful to all the other BCA students of DIT UNIVERSITY with whom I share tons of fun memories. I would like to acknowledge the support and encouragement of my friends. My sincere gratitude also goes to all those who instructed and taught me through the years.

Finally, this project would not have been possible without the confidence, endurance and support of my family. My family has always been a source of inspiration and encouragement. I wish to thank my parents, whose love, teachings and support have brought me this far.

ARYAN– 1000020005 B/P1

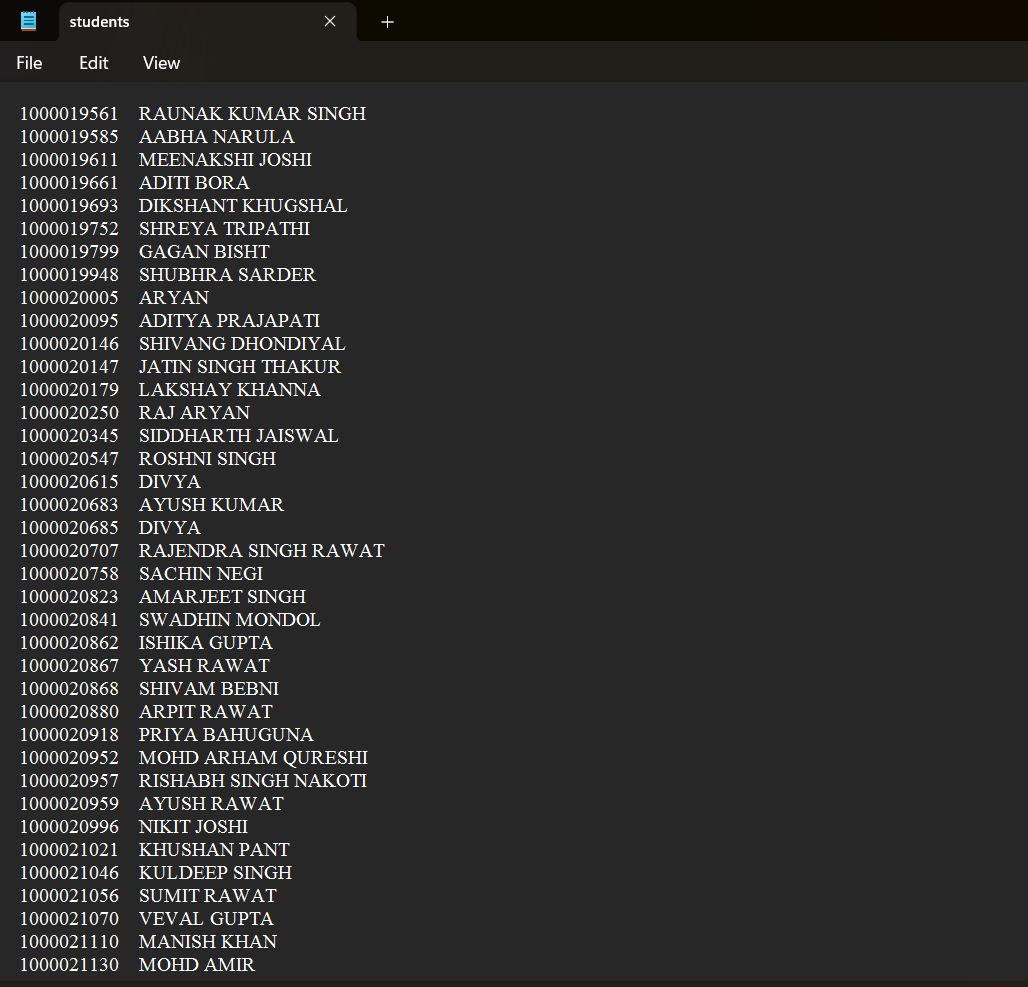
**INTRODUCTION**

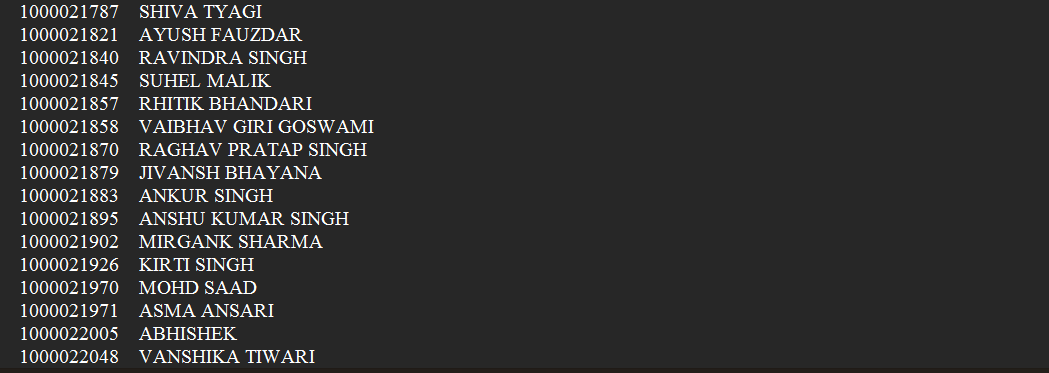
This program is designed to facilitate the retrieval of student information based on their unique SAP IDs. By utilizing a text file to store student data, which includes both SAP IDs and names, the program allows users to efficiently search for their names using a binary search algorithm. The approach ensures quick access to information, making it a practical tool for managing student records. Users simply input their SAP ID, and the program returns the corresponding name if found. This implementation showcases the effectiveness of combining data storage with efficient searching techniques.

**QUESTION:**

Create a text file that has the list of all the students in your section, the text file should have two attributes, SapID and Name. Using dictionary/hash table like data structure, implement binary search algorithm that searches for your name from the text file, given the SapID as the input parameter.

**Text File:**

****

****

**Source Code(Java):**

package Assignment;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
  
class Student {  
 int sapId;  
 String name;  
  
 public Student(int sapId, String name) {  
 this.sapId = sapId;  
 this.name = name;  
 }  
}  
  
public class BinarySearchStudent {  
 public static void main(String[] args) {  
 try {  
 File file = new File("C:\\Users\\ACER\\OneDrive\\Desktop\\students.txt");  
 Scanner scanner = new Scanner(file);  
  
 int n = 0;  
 Student[] students = new Student[100];  
 while (scanner.hasNext()) {  
 int sapId = scanner.nextInt();  
 String name = scanner.nextLine().trim();  
 students[n++] = new Student(sapId, name);  
 }  
 scanner.close();  
  
 System.*out*.print("Enter the SapID to search: ");  
 Scanner input = new Scanner(System.*in*);  
 int key = input.nextInt();  
 input.close();  
  
 int index = *binarySearch*(students, 0, n - 1, key);  
 if (index != -1) {  
 System.*out*.println("Name: " + students[index].name);  
 } else {  
 System.*out*.println("Student not found");  
 }  
  
 } catch (FileNotFoundException e) {  
 System.*out*.println("File not found.");  
 }  
 }  
  
 public static int binarySearch(Student[] students, int low, int high, int key) {  
 while (low <= high) {  
 int mid = low + (high - low) / 2;  
 if (students[mid].sapId == key) {  
 return mid;  
 } else if (students[mid].sapId < key) {  
 low = mid + 1;  
 } else {  
 high = mid - 1;  
 }  
 }  
 return -1;  
 }  
}

**OUTPUT:**

**Output 1:**

****

**Output 2:**

****