IT5512- WEB TECHNOLOGY LAB-SESSION-1

DATE: 13/09/2021 FN

NAME: A.S. PRUTHIEV

REG NO.2019506067

ARRAY AND LIST JAVA PROGRAMS

1) AIM:

To write a Array and List Java code for the calculation of student details like adding a student, search a student with his/her name,Find the student with maximum cgpa , Find the student with minimum cgpa,Sort the student by name,Sort the student by cgpa.

.

1) a

AIM:

To Write a java program to perform various operations in student class using List interface in Collection library.

ALGORITHM:

- ✓ Create a class Student with private fields like name,cgpa,age,roll number
- ✓ Create a constructor which accepts and intializes the above field varaibles
- ✓ Use getter methods for each fields to return them as they are declared as private
- ✓ Overload the toString method to get out own desired way of displaying
- ✓ In the main method,Create a ArrayList and allocate memory
- ✓ Create a menu driven program in which
    - Choice 1 → Add Student
    - Choice 2 → Remove Student by name
    - Choice 3 → Search Student by name
    - Choice 4 → Find Student with maximum cgpa
    - Choice 5 → Find Student with minimum cgpa

- Choice 6 → Sort the student list by name
- Choice 7 → Sort eh student list by cgpa
- Choice 8 → Display the student details

- ✓ For every Choice creat a method and implement the logic
- ✓ The implementation of above methods are done by the inbuitl List methods and methods in Collections class and stream features
- ✓ This process goes untill the choice given by the user is negative which will make the program to stop.

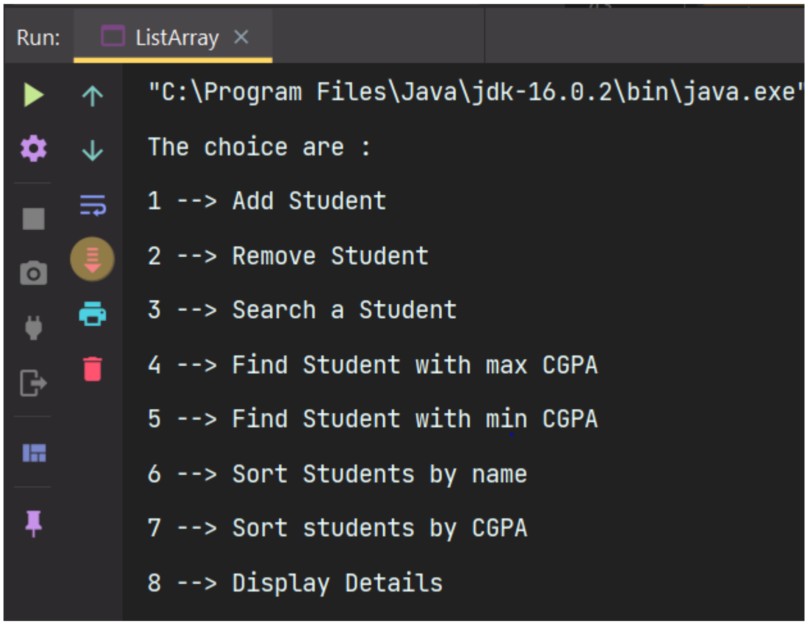
PROGRAM CODE:

```
package Java.Lab.lab2;
import java.util.*;
class Student{
   private String name,rollNumber;
   int curSem,age;
   double cgpa;
   public Student(String name, String rollNumber, int curSem, int age, double cgpa) {
      this.name = name;
      this.rollNumber = rollNumber;
      this.curSem = curSem;
      this.age = age;
      this.cgpa = cgpa;
   }
   public String getName() {
      return name;
   }
   public String getRollNumber() {
      return rollNumber;
   }
   public int getCurSem() {
      return curSem;
   }
```

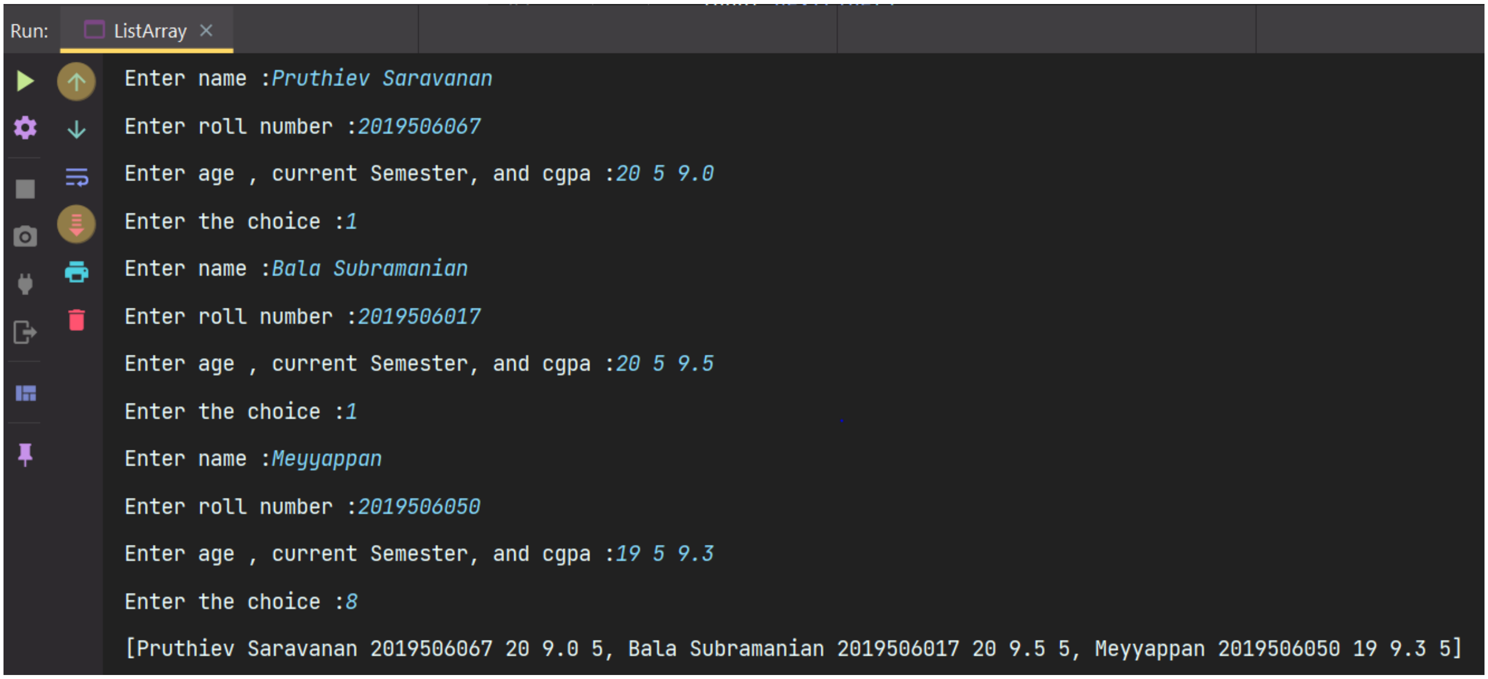
```
    public int getAge() {
        return age;
    }
    public boolean isNull(){
        return false;
    }
    public double getCgpa() {
        return cgpa;
    }
    @Override
    public String toString() {
        return name + " " + rollNumber + " " + age + " " + cgpa + " " + curSem;
    }
}
public class ListArray {
    private static Scanner input = new Scanner(System.in);
    public static void add(ArrayList<Student>students){
        int age,curSem;
        String name,rollNumber;
        double cgpa;
        input.nextLine();
        System.out.print("Enter name :");
        name = input.nextLine();
        System.out.print("Enter roll number :");
        rollNumber = input.nextLine();
        System.out.print("Enter age , current Semester, and cgpa :");
        age = input.nextInt();
        curSem = input.nextInt();
        cgpa = input.nextDouble();
        students.add(new Student(name,rollNumber,curSem,age,cgpa));
    }
    public static void remove(ArrayList<Student>students){
        input.nextLine();
        String name;
        System.out.print("Enter student name :");
        name = input.nextLine();
        if(students.removeIf(x -> x.getName().equals(name)))System.out.println("Student
name removed successfully");
        else System.out.println("Student name not exists");
    }
    public static void search(ArrayList<Student>students){
        input.nextLine();
        String name;
        System.out.print("Enter student name :");
        name = input.nextLine();
        if((students.stream().filter(x -> x.getName().equals(name)).findAny().orElse(null) !=
null)){
            System.out.println("Student exists");
        }
        else System.out.println("Student details not exists");
    }
    public static Student findMaxCGPA(ArrayList<Student>students){
        double maxCgpa = students.stream().mapToDouble(Student::getCgpa).max().orElse(-
1);
        return students.stream().filter(x -> x.getCgpa() == maxCgpa).findAny().orElse(null);
    }
    public static Student findMinCGPA(ArrayList<Student>students){
        double maxCgpa = students.stream().mapToDouble(Student::getCgpa).min().orElse(-
1);
        return students.stream().filter(x -> x.getCgpa() == maxCgpa).findAny().orElse(null);
    }
    public static void sortByName(ArrayList<Student>students){
        students.sort(Comparator.comparing(Student::getName));
    }
    public static void sortByCGPA(ArrayList<Student>students){
        students.sort(Comparator.comparing(Student :: getCgpa).reversed());
    }
    public static void main(String[] args) {
        ArrayList<Student>students = new ArrayList<>();
        int choice;
    System.out.println("The choice are :");
    System.out.println("1 --> Add Student\n2 --> Remove Student\n3 --> Search a
Student\n4 --> Find Student with max CGPA" +
        "\n5 --> Find Student with min CGPA\n6 --> Sort Students by name\n7 --> Sort
students by CGPA\n8 --> Display Details\n");
    do {
      System.out.print("Enter the choice :");
      choice = input.nextInt();
      switch (choice) {
        case 1 -> add(students);
        case 2 -> remove(students);
        case 3 -> search(students);
        case 4 -> {
          Student st = findMaxCGPA(students);
          if (st != null) {
            System.out.println(st);
          }
        }
        case 5 -> {
          Student st1 = findMinCGPA(students);
          if (st1 != null) {
            System.out.println(st1);
          }
        }
        case 6 -> {
          sortByName(students);
          System.out.println(students);
        }
        case 7 -> {
          sortByCGPA(students);
          System.out.println(students);
        }
        case 8 -> {
          System.out.println(students);
        }
```

```
                default -> System.out.println("Enter a valid choice !!!");
            }
        }while(choice >= 0);
    }
}
```

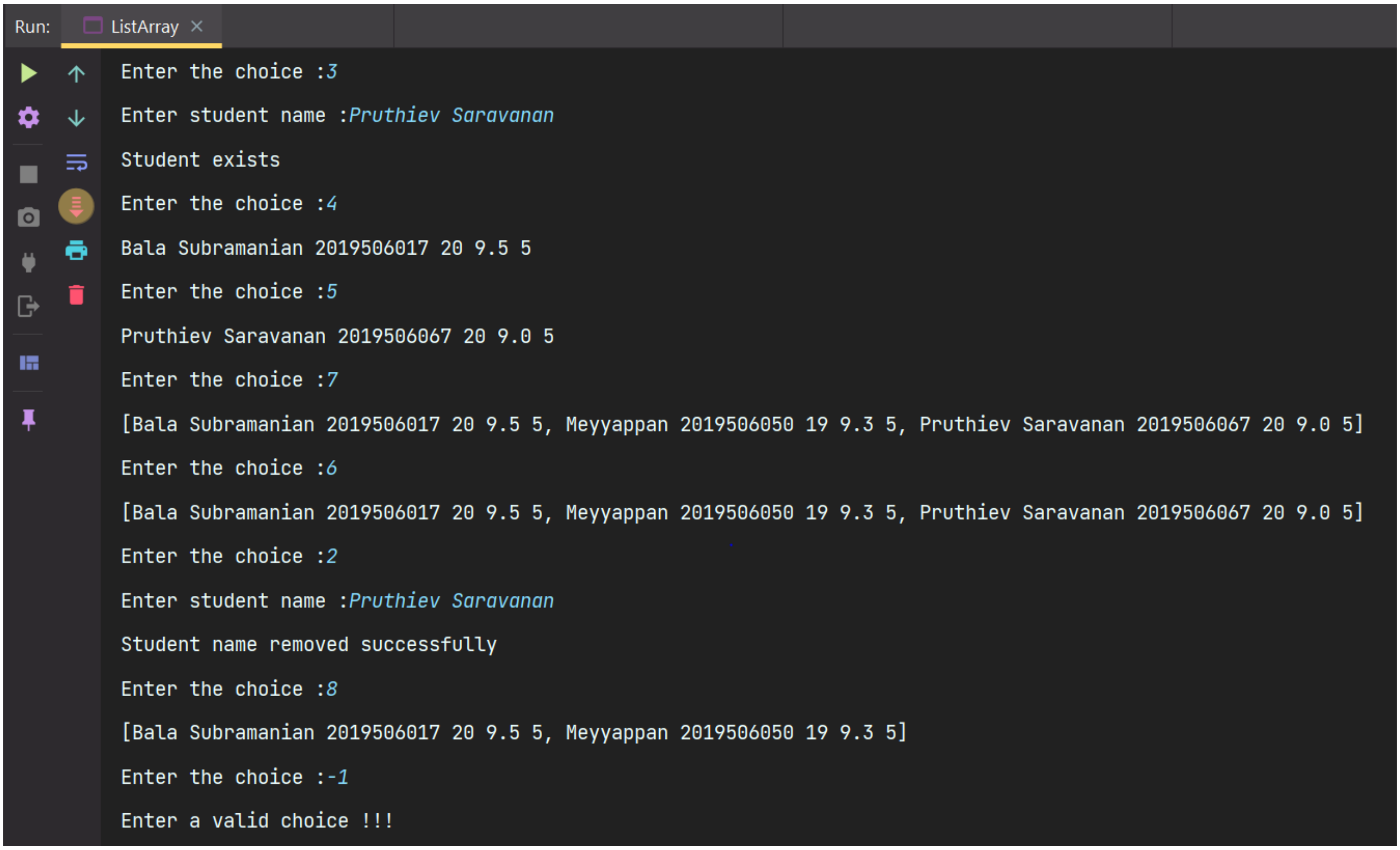
OUTPUT:



```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe"
The choice are :
1 --> Add Student
2 --> Remove Student
3 --> Search a Student
4 --> Find Student with max CGPA
5 --> Find Student with min CGPA
6 --> Sort Students by name
7 --> Sort students by CGPA
8 --> Display Details
```



```
Enter name :Pruthiev Saravanan
Enter roll number :2019506067
Enter age , current Semester, and cgpa :20 5 9.0
Enter the choice :1
Enter name :Bala Subramanian
Enter roll number :2019506017
Enter age , current Semester, and cgpa :20 5 9.5
Enter the choice :1
Enter name :Meyyappan
Enter roll number :2019506050
Enter age , current Semester, and cgpa :19 5 9.3
Enter the choice :8
[Pruthiev Saravanan 2019506067 20 9.0 5, Bala Subramanian 2019506017 20 9.5 5, Meyyappan 2019506050 19 9.3 5]
```



```
Enter the choice :3
Enter student name :Pruthiev Saravanan
Student exists
Enter the choice :4
Bala Subramanian 2019506017 20 9.5 5
Enter the choice :5
Pruthiev Saravanan 2019506067 20 9.0 5
Enter the choice :7
[Bala Subramanian 2019506017 20 9.5 5, Meyyappan 2019506050 19 9.3 5, Pruthiev Saravanan 2019506067 20 9.0 5]
Enter the choice :6
[Bala Subramanian 2019506017 20 9.5 5, Meyyappan 2019506050 19 9.3 5, Pruthiev Saravanan 2019506067 20 9.0 5]
Enter the choice :2
Enter student name :Pruthiev Saravanan
Student name removed successfully
Enter the choice :8
[Bala Subramanian 2019506017 20 9.5 5, Meyyappan 2019506050 19 9.3 5]
Enter the choice :-1
Enter a valid choice !!!
```

RESULT:

Thus, the program has been executed successfully.

1) b.

AIM:

To Write a java program perform various operations in student class using Array data type

ALGORITHM:

- ✓ Enter the number of students as in case of array and read it in n
- ✓ In the main method,Create a Student Array[] and allocate the size as n.
- ✓ Create a menu driven program in which
    - Choice 1 → Add Student
    - Choice 2 → Remove Student by name
    - Choice 3 → Search Student by name
    - Choice 4 → Find Student with maximum cgpa
    - Choice 5 → Find Student with minimum cgpa
    - Choice 6 → Sort the student list by name
    - Choice 7 → Sort eh student list by cgpa
    - Choice 8 → Display the student details

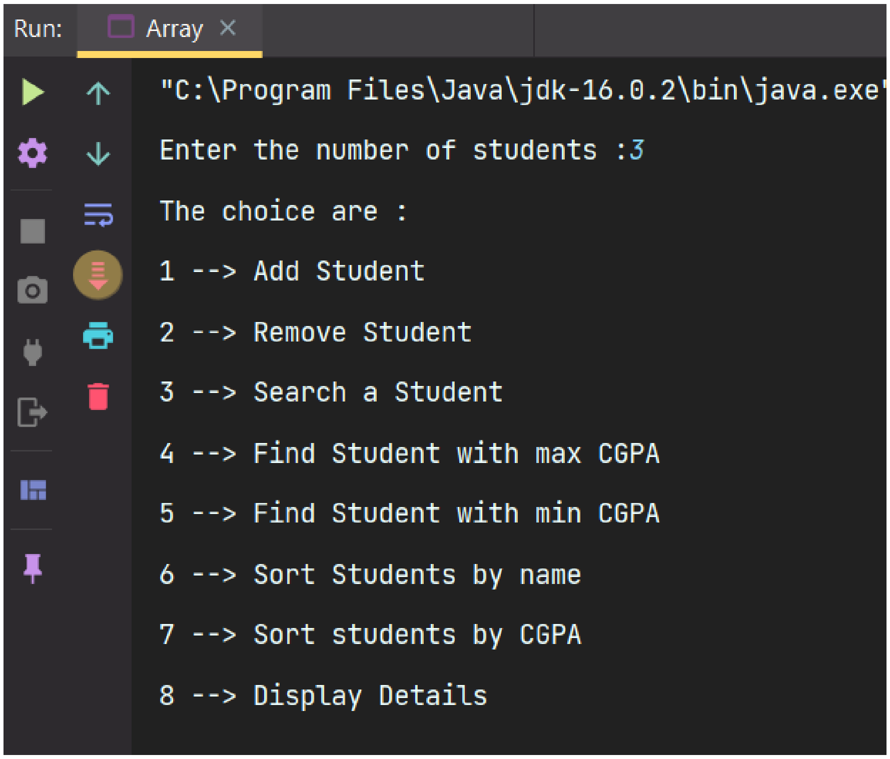
- ✓ For every Choice creat a method and implement the logic
- ✓ These methods are implemeted by the in built methods available in Arrays class and stream features.
- ✓ This process goes untill the choice given by the user is negative which will make the program to stop.

PROGRAM CODE

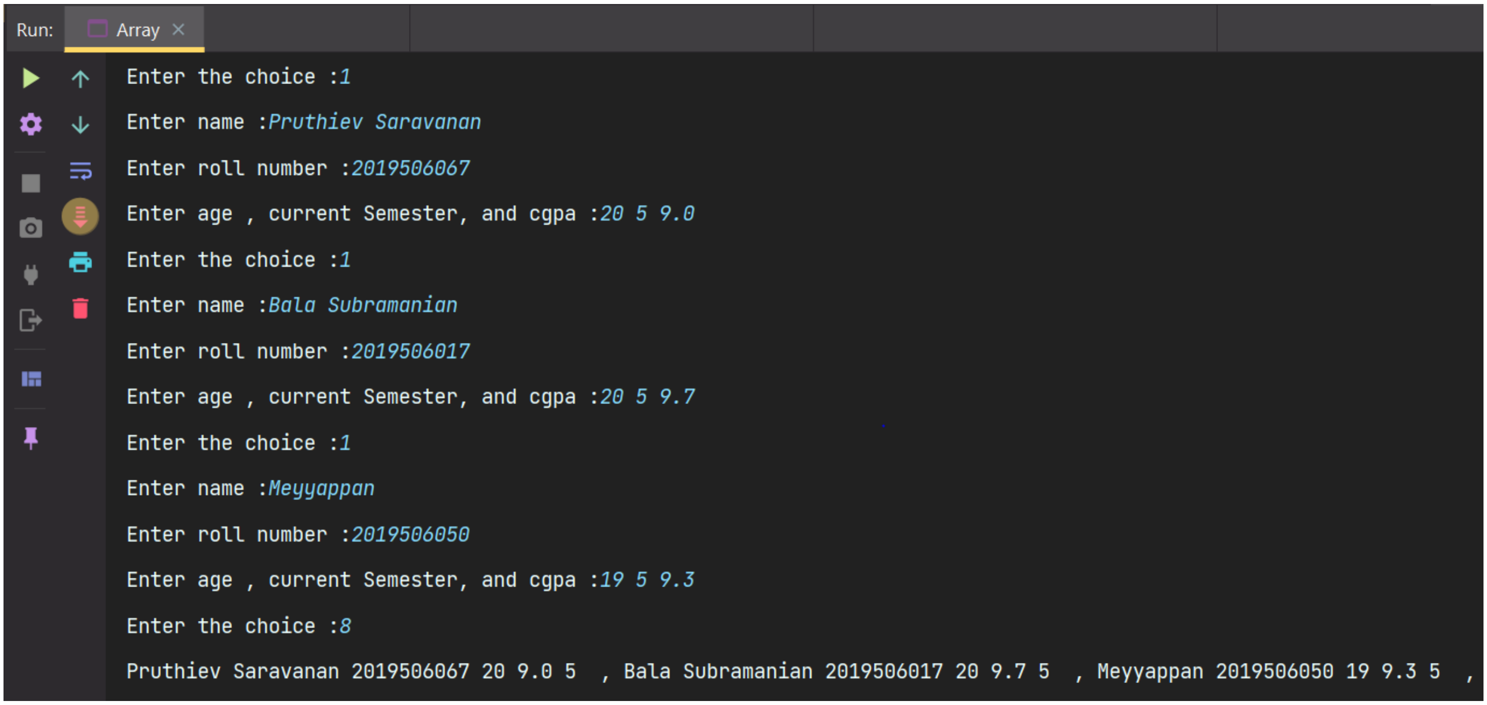
```
package Java.Lab.lab2;

import java.util.*;

public class Array {
    private static Scanner input = new Scanner(System.in);
    static int count = 0;
    public static void add(Student[]students){
        int age,curSem;
        String name,rollNumber;
        double cgpa;
        input.nextLine();
        System.out.print("Enter name :");
        name = input.nextLine();
        System.out.print("Enter roll number :");
        rollNumber = input.nextLine();
        System.out.print("Enter age , current Semester, and cgpa :");
        age = input.nextInt();
        curSem = input.nextInt();
        cgpa = input.nextDouble();
        if(count >= students.length)System.out.println("Array is full ");
        else students[count++] = new Student(name,rollNumber,curSem,age,cgpa);
    }
    public static void remove(Student[]students){
        input.nextLine();
        String name;
        System.out.print("Enter student name :");
        name = input.nextLine();
        int index = -1;
        for(int i = 0 ; i < students.length ; i++){
            if(students[i].getName().equals(name)){
                index = i;
                break;
```

```
         }
      }
      if(index != -1){
         for(int i = index ; i < students.length-1 ; i++){
            students[i] = students[i+1];
         }
         students[count-1] = null;
         count --;
         System.out.println("The student details are removed successfully");
      }
      else{
         System.out.println("Student does not exists");
      }

   }
   public static void search(Student[]students){
      input.nextLine();
      String name;
      System.out.print("Enter student name :");
      name = input.nextLine();
      Student findStudent = Arrays.stream(students).filter(x ->
(x.getName().equals(name))).findAny().orElse(null);
      if(findStudent != null)System.out.println("The student detail exists");
      else System.out.println("The student details does not exists");
   }
   public static Student findMaxCGPA(Student[]students){
      return Arrays.stream(students).max(Comparator.comparing(Student ::
getCgpa)).orElse(null);
   }
   public static Student findMinCGPA(Student[]students){
      return Arrays.stream(students).min(Comparator.comparing(Student ::
getCgpa)).orElse(null);
   }
   public static void sortByName(Student[]students){
        Arrays.sort(students,Comparator.nullsLast(Comparator.comparing(Student
::getName)));
    }
    public static void sortByCGPA(Student[]students){
        Arrays.sort(students,Comparator.nullsLast(Comparator.comparing(Student
::getCgpa).reversed()));
    }
    public static void main(String[] args) {
        int n;
        System.out.println("Enter the number of students :");
        n = input.nextInt();
        Student[]students = new Student[n];
        int choice;
        System.out.println("The choice are :");
        System.out.println("1 --> Add Student\n2 --> Remove Student\n3 --> Search a
Student\n4 --> Find Student with max CGPA" +
                    "\n5 --> Find Student with min CGPA\n6 --> Sort Students by name\n7 --> Sort
students by CGPA\n8 --> Display Details\n");
        do {
            System.out.print("Enter the choice :");
            choice = input.nextInt();
            switch (choice) {
                case 1 -> add(students);
                case 2 -> remove(students);
                case 3 -> search(students);
                case 4 -> {
                    Student st = findMaxCGPA(students);
                    if (st != null) {
                        System.out.println(st.toString());
                    }
                }
                case 5 -> {
                    Student st1 = findMinCGPA(students);
                    if (st1 != null) {
                        System.out.println(st1.toString());
                    }
                }
                case 6 -> {
                    sortByName(students);
                    Arrays.stream(students).filter(Objects::nonNull).forEach((x)                ->
System.out.print(x.toString() + " , "));
                    System.out.println();
                }
                case 7 -> {
                    sortByCGPA(students);
                    Arrays.stream(students).filter(Objects::nonNull).forEach((x)  ->  System.out.print
(x.toString() + " , "));
                    System.out.println();
                }
                case 8 -> {
                    Arrays.stream(students).filter(Objects::nonNull).forEach((x)                ->
System.out.print(x.toString() + "  , "));
                    System.out.println();
                }
                default -> System.out.println("Enter a valid choice !!!");
            }
        }while(choice >= 0);
    }

}
```

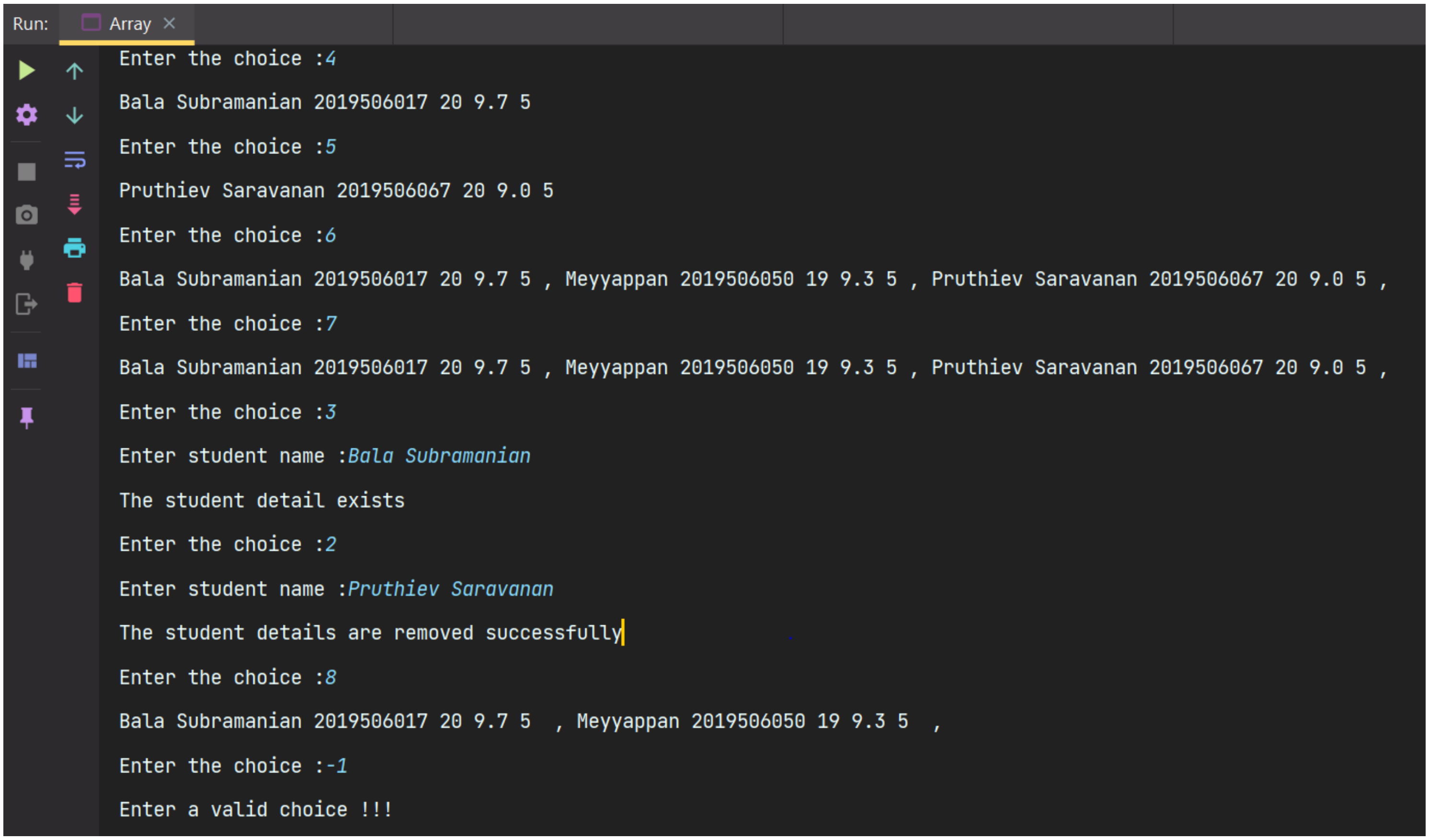
OUTPUT:



```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe"
Enter the number of students :3
The choice are :
1 --> Add Student
2 --> Remove Student
3 --> Search a Student
4 --> Find Student with max CGPA
5 --> Find Student with min CGPA
6 --> Sort Students by name
7 --> Sort students by CGPA
8 --> Display Details
```



```
Enter the choice :1
Enter name :Pruthiev Saravanan
Enter roll number :2019506067
Enter age , current Semester, and cgpa :20 5 9.0
Enter the choice :1
Enter name :Bala Subramanian
Enter roll number :2019506017
Enter age , current Semester, and cgpa :20 5 9.7
Enter the choice :1
Enter name :Meyyappan
Enter roll number :2019506050
Enter age , current Semester, and cgpa :19 5 9.3
Enter the choice :8
Pruthiev Saravanan 2019506067 20 9.0 5  , Bala Subramanian 2019506017 20 9.7 5  , Meyyappan 2019506050 19 9.3 5  ,
```



```
Enter the choice :4
Bala Subramanian 2019506017 20 9.7 5
Enter the choice :5
Pruthiev Saravanan 2019506067 20 9.0 5
Enter the choice :6
Bala Subramanian 2019506017 20 9.7 5 , Meyyappan 2019506050 19 9.3 5 , Pruthiev Saravanan 2019506067 20 9.0 5 ,
Enter the choice :7
Bala Subramanian 2019506017 20 9.7 5 , Meyyappan 2019506050 19 9.3 5 , Pruthiev Saravanan 2019506067 20 9.0 5 ,
Enter the choice :3
Enter student name :Bala Subramanian
The student detail exists
Enter the choice :2
Enter student name :Pruthiev Saravanan
The student details are removed successfully
Enter the choice :8
Bala Subramanian 2019506017 20 9.7 5  , Meyyappan 2019506050 19 9.3 5  ,
Enter the choice :-1
Enter a valid choice !!!
```

RESULT:

Thus, the program has been executed successfully.

LIST (Collection Library) vs Array(Data type)

LIST	ARRAY
✓ A list is an interface that extends from collection interface	✓ An array is a in-built data type which extends from Object class
✓ It can only store homogeneous elements at any instant of time	✓ It too can only store homogeneous collection of elements
✓ No need of specifying the size before it can resize and add elements dynamically	✓ The size of the array has to be specified before adding elements to it.
✓ It has in-built methods like add , remove,search,isEmpty etc.	✓ No built in menthods as such each and everything has to implemented by the programmer who uses array
✓ Since it is interface the memory is allocated on heap.	✓ Since it is a class it is allocated in heap
✓ It can only store non-primitives	✓ It can store both primitive anhd non primitive data types
✓ We can traverse the array using ,forEach and iterator as List provide the support of iterators as they implemt the iterable interface.	✓ We can traverse using for and enchaned for loop as ther is no support for iterators
✓ We can convert the List to array using toArray methods defined in the Collections class.	✓ We can convert array to List using Arrays.asList() method defined in the Arrays class.
✓ We can perform in-built operations using the interface as well as using the Collections class.	✓ We can perform in-built operations only using Arrays class.
✓ The syntax and data type to be mentioned is	✓ The syntax and data type to mentioned is as follows
List<DataType>l = new ArrayList<>();	DataType arr[] = new DataType[size];