VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

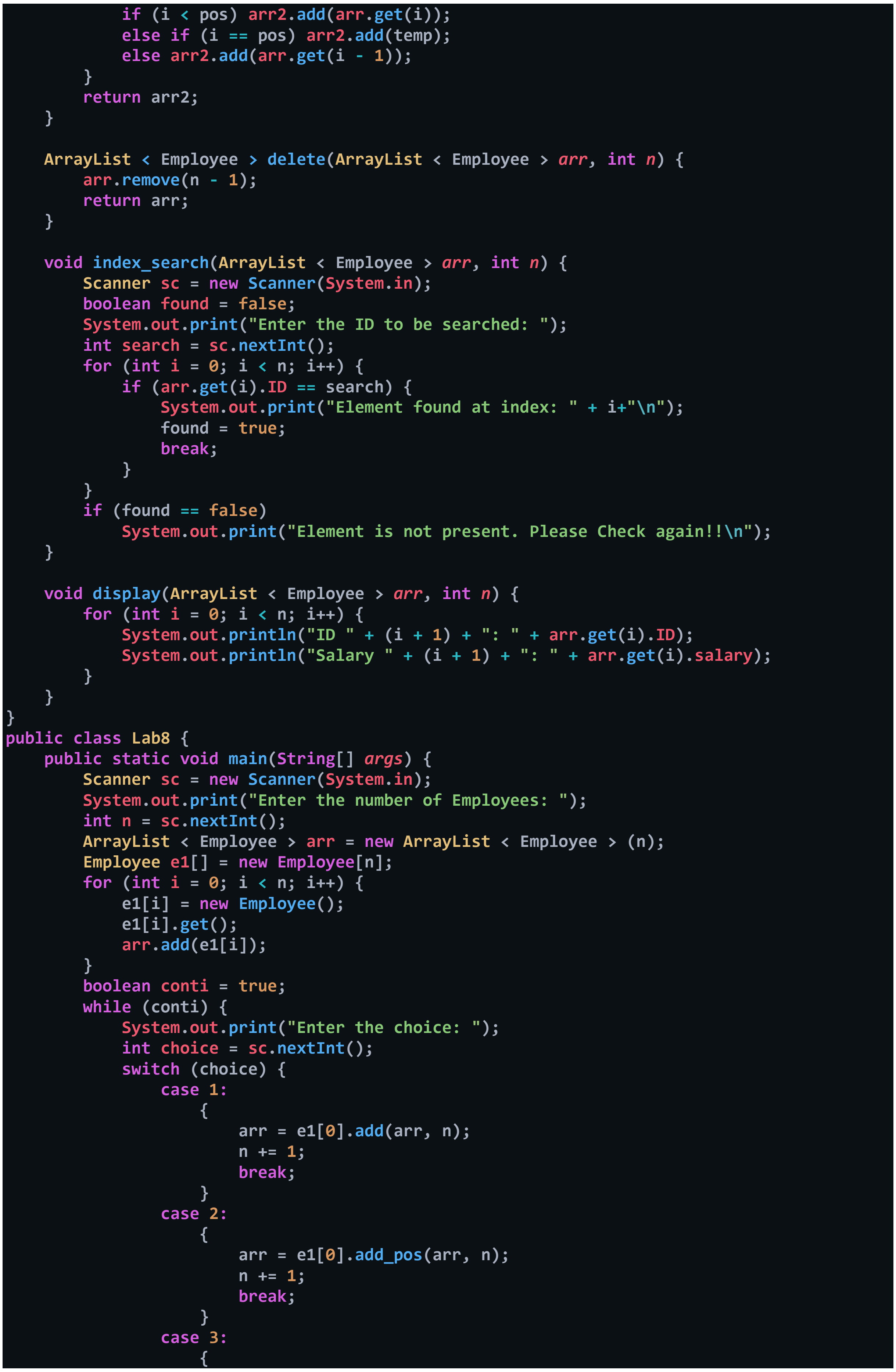
Programme	:	**BTech. CSE Core**	Semester	:	**Win 2021-22**
Course	:	**Java Programming**	Code	:	**CSE1007**
Faculty	:	**Dr. Pradeep K**	Slot	:	**L9+L10**
Name	:	**Hariket Sukesh Kumar Sheth**	Register No.	:	**20BCE1975**

1. Create a class named Employee. Data fields for Employee include an Integer ID number and a salary. Methods include a constructor that requires values for both data fields as well as get and set methods for each of the data fields.

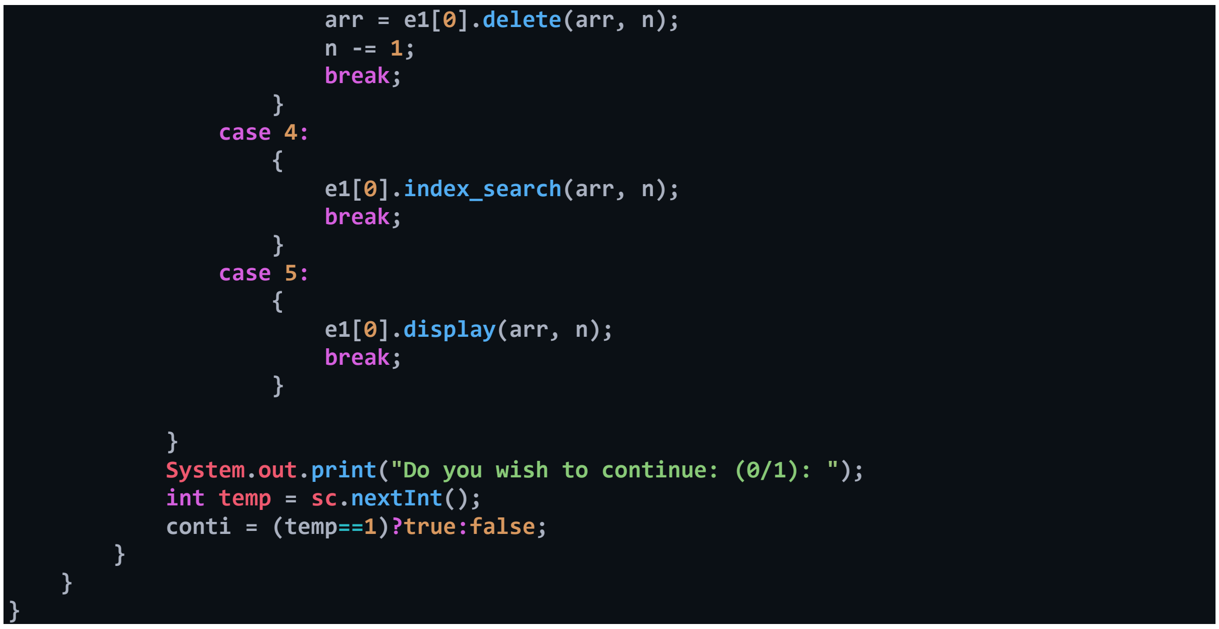
Now create an Application that allows you to store an ArrayList that acts as a Database of any number of Employee Objects. While the user decides to continue, offer the following options:

Choice 1: Add()
Choice 2: Add with position()
Choice 3: delete()
Choice 4: index search()
Choice 5: display()

```java
package lab8;
import java.util.*;
import java.io.*;

class Employee {
    int ID, salary;
    void get() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the ID: ");
        ID = sc.nextInt();
        System.out.print("Enter the salary: ");
        salary = sc.nextInt();
    }

    int getID(ArrayList < Employee > arr) {
        return ID;
    }
    ArrayList < Employee > add(ArrayList < Employee > arr, int n) {
        ArrayList < Employee > arr2 = arr;
        Employee temp = new Employee();
        temp.get();
        arr2.add(temp);
        return arr2;
    }
    ArrayList < Employee > add_pos(ArrayList < Employee > arr, int n) {
        Scanner sc = new Scanner(System.in);
        ArrayList < Employee > arr2 = new ArrayList < Employee > (n);
        System.out.print ("Enter the position at which data have to be inserted: ");
        int pos = sc.nextInt();
        Employee temp = new Employee();
        temp.get();
        for (int i = 0; i < n + 1; i++) {
```

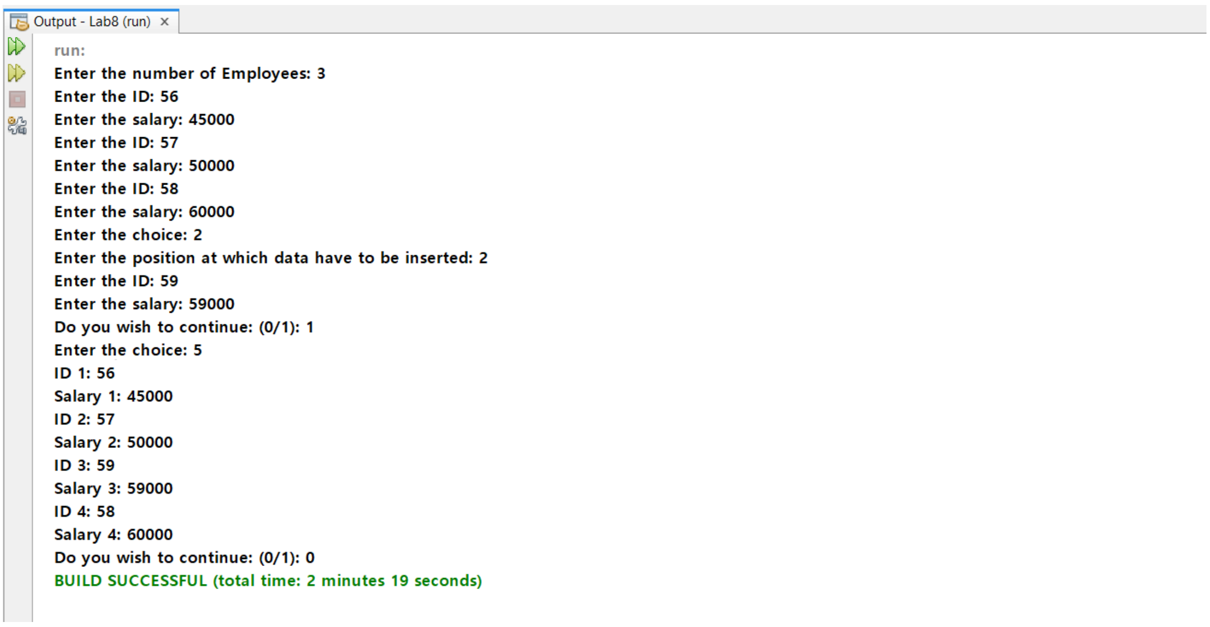


```java
            if (i < pos) arr2.add(arr.get(i));
            else if (i == pos) arr2.add(temp);
            else arr2.add(arr.get(i - 1));
        }
        return arr2;
    }

    ArrayList < Employee > delete(ArrayList < Employee > arr, int n) {
        arr.remove(n - 1);
        return arr;
    }

    void index_search(ArrayList < Employee > arr, int n) {
        Scanner sc = new Scanner(System.in);
        boolean found = false;
        System.out.print("Enter the ID to be searched: ");
        int search = sc.nextInt();
        for (int i = 0; i < n; i++) {
            if (arr.get(i).ID == search) {
                System.out.print("Element found at index: " + i+"\n");
                found = true;
                break;
            }
        }
        if (found == false)
            System.out.print("Element is not present. Please Check again!!\n");
    }

    void display(ArrayList < Employee > arr, int n) {
        for (int i = 0; i < n; i++) {
            System.out.println("ID " + (i + 1) + ": " + arr.get(i).ID);
            System.out.println("Salary " + (i + 1) + ": " + arr.get(i).salary);
        }
    }
}
public class Lab8 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of Employees: ");
        int n = sc.nextInt();
        ArrayList < Employee > arr = new ArrayList < Employee > (n);
        Employee e1[] = new Employee[n];
        for (int i = 0; i < n; i++) {
            e1[i] = new Employee();
            e1[i].get();
            arr.add(e1[i]);
        }
        boolean conti = true;
        while (conti) {
            System.out.print("Enter the choice: ");
            int choice = sc.nextInt();
            switch (choice) {
                case 1:
                    {
                        arr = e1[0].add(arr, n);
                        n += 1;
                        break;
                    }
                case 2:
                    {
                        arr = e1[0].add_pos(arr, n);
                        n += 1;
                        break;
                    }
                case 3:
                    {
```

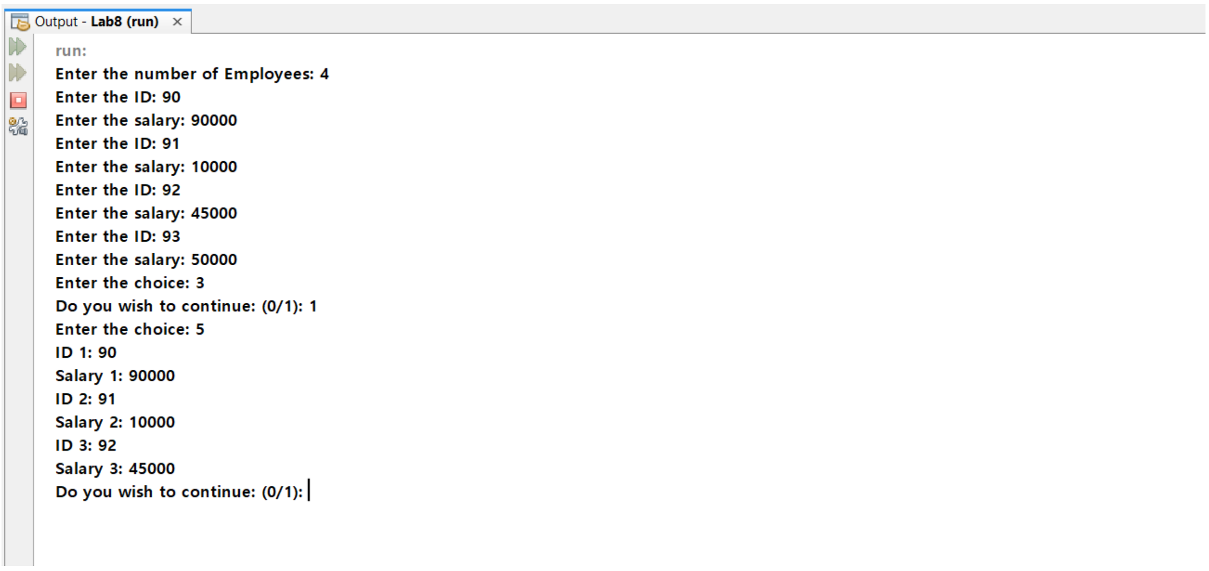


```
                        arr = e1[0].delete(arr, n);
                        n -= 1;
                        break;
                    }
                case 4:
                    {
                        e1[0].index_search(arr, n);
                        break;
                    }
                case 5:
                    {
                        e1[0].display(arr, n);
                        break;
                    }

            }
            System.out.print("Do you wish to continue: (0/1): ");
            int temp = sc.nextInt();
            conti = (temp==1)?true:false;
        }
    }
}
```

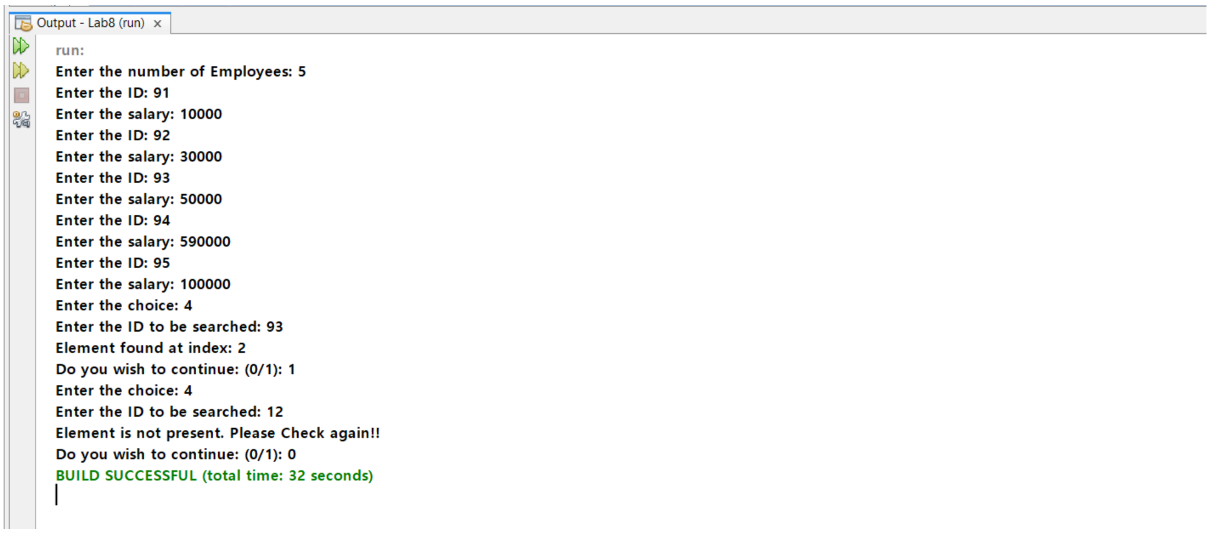
OUTPUT:



```
Output - Lab8 (run)
run:
Enter the number of Employees: 3
Enter the ID: 56
Enter the salary: 45000
Enter the ID: 57
Enter the salary: 50000
Enter the ID: 58
Enter the salary: 60000
Enter the choice: 2
Enter the position at which data have to be inserted: 2
Enter the ID: 59
Enter the salary: 59000
Do you wish to continue: (0/1): 1
Enter the choice: 5
ID 1: 56
Salary 1: 45000
ID 2: 57
Salary 2: 50000
ID 3: 59
Salary 3: 59000
ID 4: 58
Salary 4: 60000
Do you wish to continue: (0/1): 0
BUILD SUCCESSFUL (total time: 2 minutes 19 seconds)
```



```
run:
Enter the number of Employees: 4
Enter the ID: 90
Enter the salary: 90000
Enter the ID: 91
Enter the salary: 10000
Enter the ID: 92
Enter the salary: 45000
Enter the ID: 93
Enter the salary: 50000
Enter the choice: 3
Do you wish to continue: (0/1): 1
Enter the choice: 5
ID 1: 90
Salary 1: 90000
ID 2: 91
Salary 2: 10000
ID 3: 92
Salary 3: 45000
Do you wish to continue: (0/1):
```

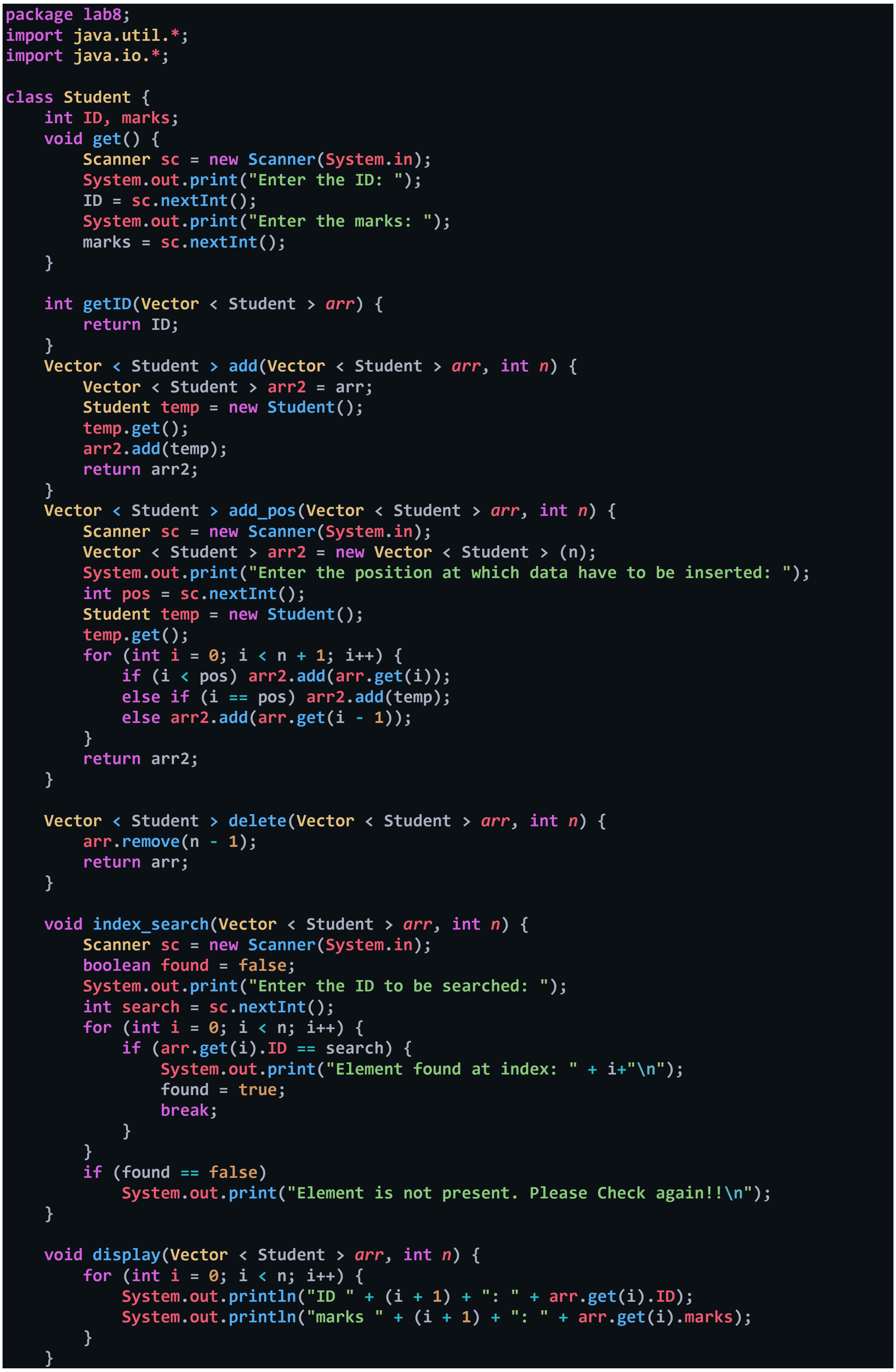


```
run:
Enter the number of Employees: 5
Enter the ID: 91
Enter the salary: 10000
Enter the ID: 92
Enter the salary: 30000
Enter the ID: 93
Enter the salary: 50000
Enter the ID: 94
Enter the salary: 590000
Enter the ID: 95
Enter the salary: 100000
Enter the choice: 4
Enter the ID to be searched: 93
Element found at index: 2
Do you wish to continue: (0/1): 1
Enter the choice: 4
Enter the ID to be searched: 12
Element is not present. Please Check again!!
Do you wish to continue: (0/1): 0
BUILD SUCCESSFUL (total time: 32 seconds)
```

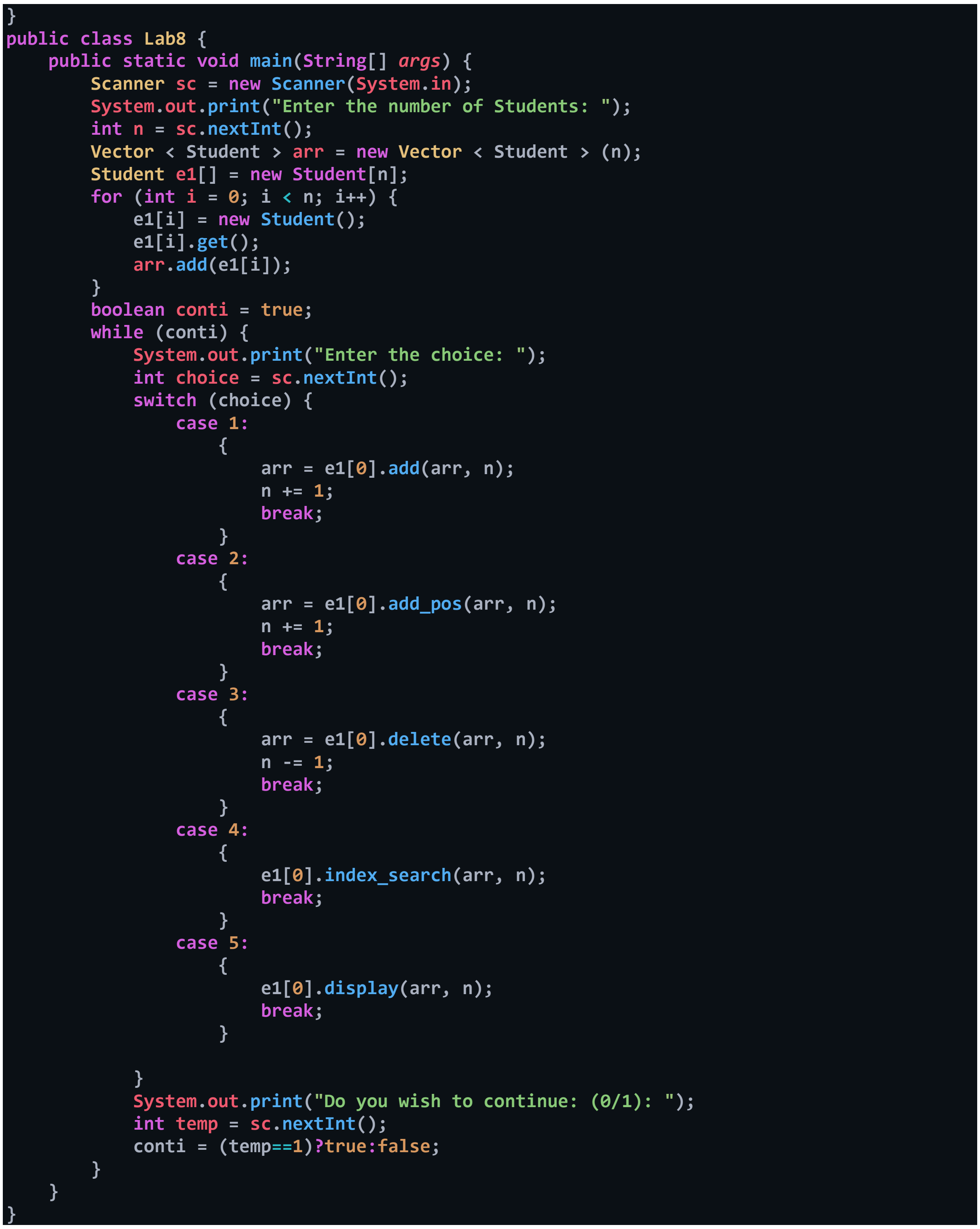
2. Create a class named Student. Data fields for Student include an integer ID number and a Total marks. Methods include a constructor that requires values for both data fields, as well as get and set methods for each of the data fields.

   Now, create an application that allows you to store an vector that acts as a database of any number of Student objects. While the user decides to continue, offer the following options

   Choice 1: Add()
   Choice 1: Add with position()
   Choice 2: delete()
   Choice 3: Search()
   Choice 4: index search()
   Choice 5: display()

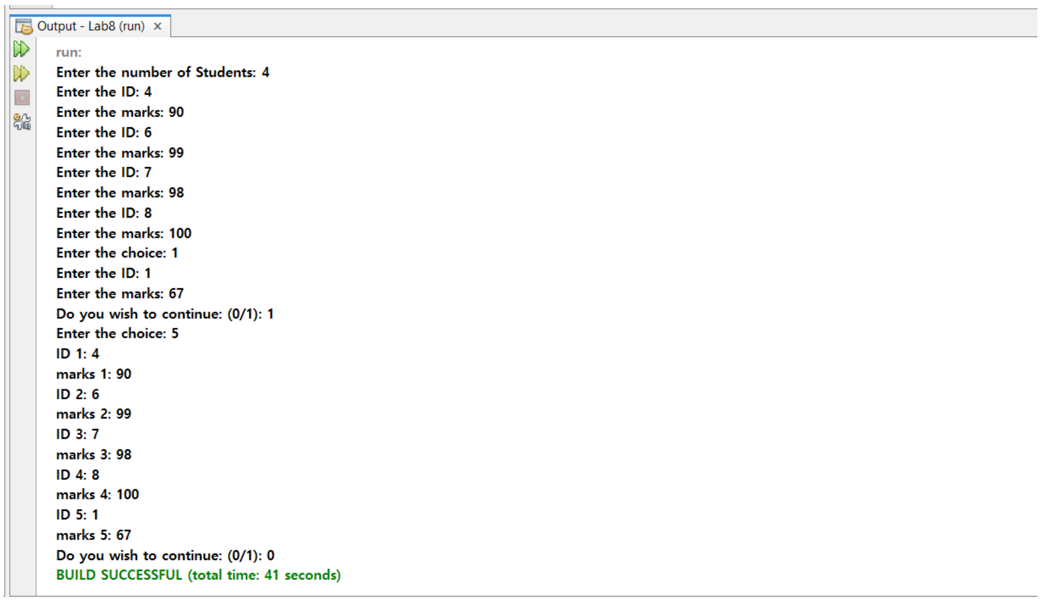


```java
package lab8;
import java.util.*;
import java.io.*;

class Student {
    int ID, marks;
    void get() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the ID: ");
        ID = sc.nextInt();
        System.out.print("Enter the marks: ");
        marks = sc.nextInt();
    }

    int getID(Vector < Student > arr) {
        return ID;
    }
    Vector < Student > add(Vector < Student > arr, int n) {
        Vector < Student > arr2 = arr;
        Student temp = new Student();
        temp.get();
        arr2.add(temp);
        return arr2;
    }
    Vector < Student > add_pos(Vector < Student > arr, int n) {
        Scanner sc = new Scanner(System.in);
        Vector < Student > arr2 = new Vector < Student > (n);
        System.out.print("Enter the position at which data have to be inserted: ");
        int pos = sc.nextInt();
        Student temp = new Student();
        temp.get();
        for (int i = 0; i < n + 1; i++) {
            if (i < pos) arr2.add(arr.get(i));
            else if (i == pos) arr2.add(temp);
            else arr2.add(arr.get(i - 1));
        }
        return arr2;
    }

    Vector < Student > delete(Vector < Student > arr, int n) {
        arr.remove(n - 1);
        return arr;
    }

    void index_search(Vector < Student > arr, int n) {
        Scanner sc = new Scanner(System.in);
        boolean found = false;
        System.out.print("Enter the ID to be searched: ");
        int search = sc.nextInt();
        for (int i = 0; i < n; i++) {
            if (arr.get(i).ID == search) {
                System.out.print("Element found at index: " + i+"\n");
                found = true;
                break;
            }
        }
        if (found == false)
            System.out.print("Element is not present. Please Check again!!\n");
    }

    void display(Vector < Student > arr, int n) {
        for (int i = 0; i < n; i++) {
            System.out.println("ID " + (i + 1) + ": " + arr.get(i).ID);
            System.out.println("marks " + (i + 1) + ": " + arr.get(i).marks);
        }
    }
```



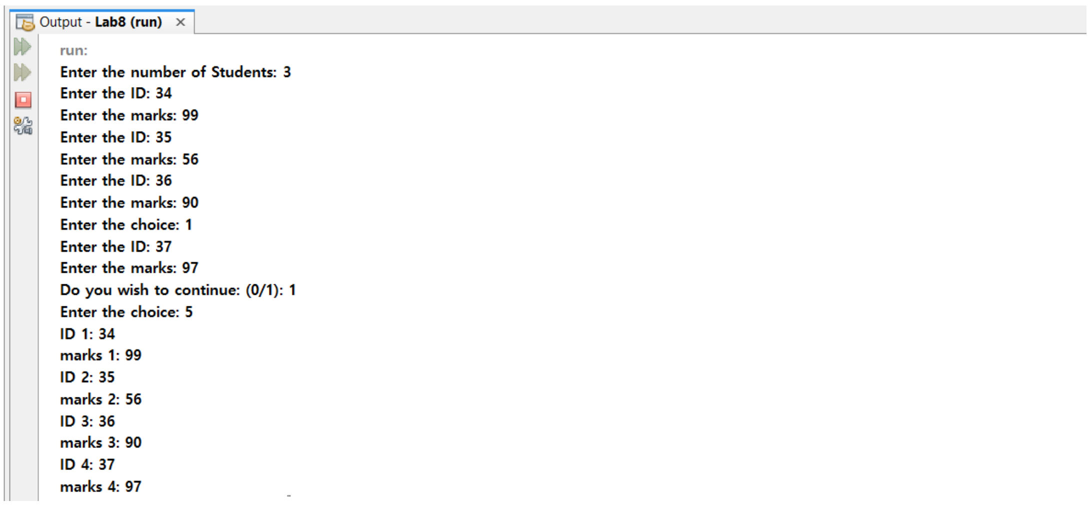
```java
}
public class Lab8 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of Students: ");
        int n = sc.nextInt();
        Vector < Student > arr = new Vector < Student > (n);
        Student e1[] = new Student[n];
        for (int i = 0; i < n; i++) {
            e1[i] = new Student();
            e1[i].get();
            arr.add(e1[i]);
        }
        boolean conti = true;
        while (conti) {
            System.out.print("Enter the choice: ");
            int choice = sc.nextInt();
            switch (choice) {
                case 1:
                    {
                        arr = e1[0].add(arr, n);
                        n += 1;
                        break;
                    }
                case 2:
                    {
                        arr = e1[0].add_pos(arr, n);
                        n += 1;
                        break;
                    }
                case 3:
                    {
                        arr = e1[0].delete(arr, n);
                        n -= 1;
                        break;
                    }
                case 4:
                    {
                        e1[0].index_search(arr, n);
                        break;
                    }
                case 5:
                    {
                        e1[0].display(arr, n);
                        break;
                    }

            }
            System.out.print("Do you wish to continue: (0/1): ");
            int temp = sc.nextInt();
            conti = (temp==1)?true:false;
        }
    }
}
```

OUTPUT:



Output - Lab8 (run)

```
run:
Enter the number of Students: 4
Enter the ID: 4
Enter the marks: 90
Enter the ID: 6
Enter the marks: 99
Enter the ID: 7
Enter the marks: 98
Enter the ID: 8
Enter the marks: 100
Enter the choice: 1
Enter the ID: 1
Enter the marks: 67
Do you wish to continue: (0/1): 1
Enter the choice: 5
ID 1: 4
marks 1: 90
ID 2: 6
marks 2: 99
ID 3: 7
marks 3: 98
ID 4: 8
marks 4: 100
ID 5: 1
marks 5: 67
Do you wish to continue: (0/1): 0
BUILD SUCCESSFUL (total time: 41 seconds)
```

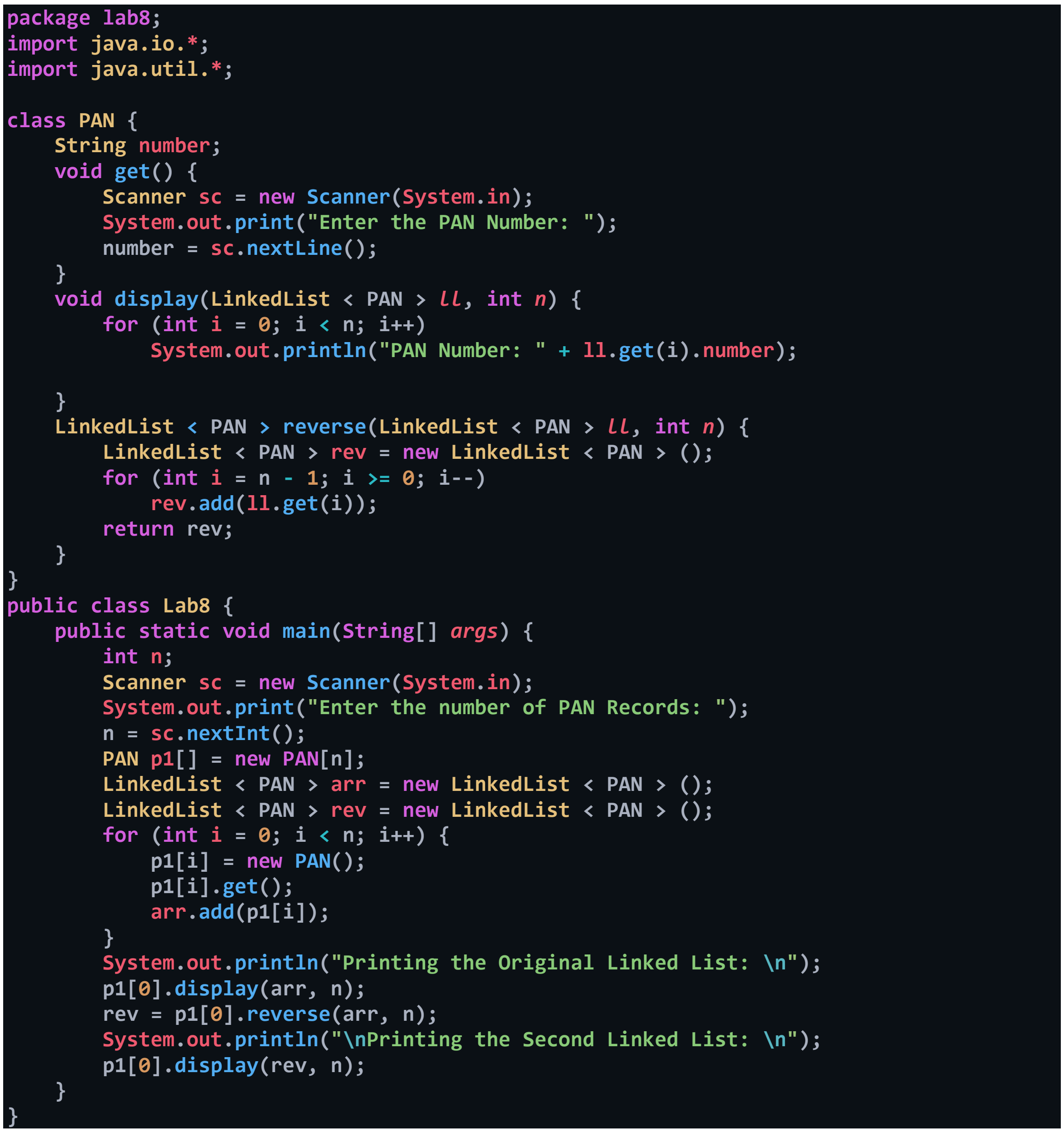


Output - Lab8 (run)

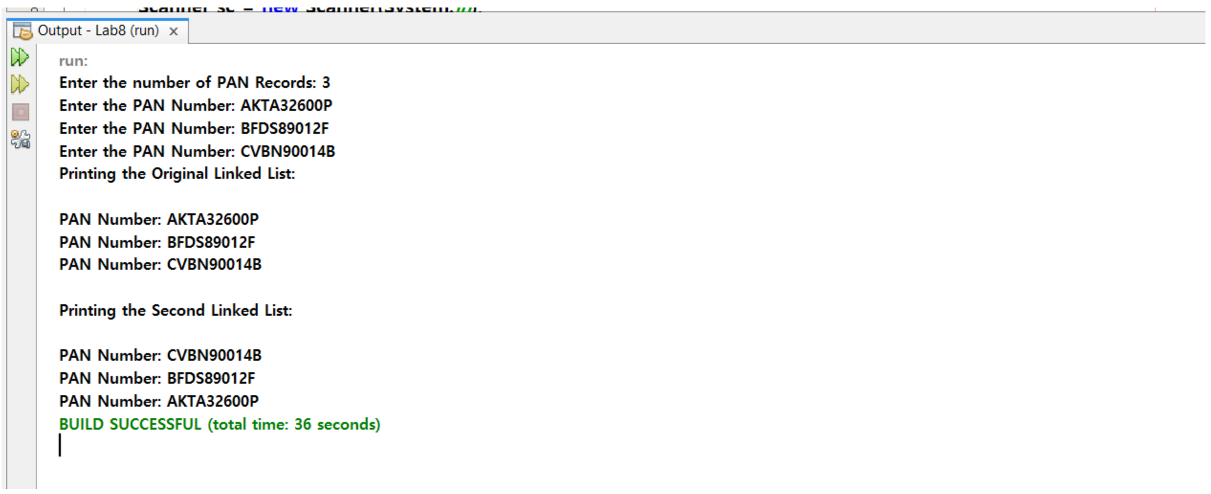
```
run:
Enter the number of Students: 3
Enter the ID: 34
Enter the marks: 99
Enter the ID: 35
Enter the marks: 56
Enter the ID: 36
Enter the marks: 90
Enter the choice: 1
Enter the ID: 37
Enter the marks: 97
Do you wish to continue: (0/1): 1
Enter the choice: 5
ID 1: 34
marks 1: 99
ID 2: 35
marks 2: 56
ID 3: 36
marks 3: 90
ID 4: 37
marks 4: 97
```

```
Enter the choice: 3
Do you wish to continue: (0/1): 1
Enter the choice: 5
ID 1: 34
marks 1: 99
ID 2: 35
marks 2: 56
ID 3: 36
marks 3: 90
Do you wish to continue: (0/1): 1
Enter the choice: 3
Do you wish to continue: (0/1): 1
Enter the choice: 5
ID 1: 34
marks 1: 99
ID 2: 35
marks 2: 56
Do you wish to continue: (0/1):
```

3. Write a program that creates a LinkedList which consists the PAN number as object of 10 Alpha Numeric values, then creates a second LinkedList object containing a copy of the first list, but in reverse order and then followed by the Digits in the first list. Use ListIterators wherever it is necessary. Write the same input and output of this program under all possible test cases. (Ex. PAN Number: APTPP3299K).



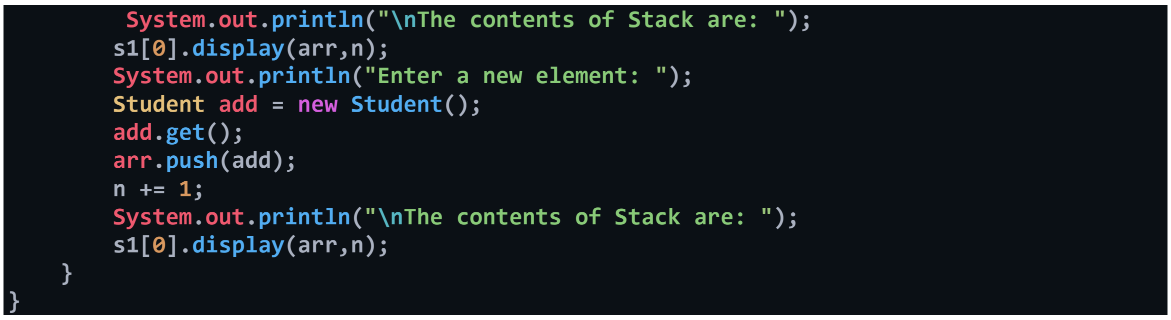
```
package lab8;
import java.io.*;
import java.util.*;

class PAN {
    String number;
    void get() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the PAN Number: ");
        number = sc.nextLine();
    }
    void display(LinkedList < PAN > ll, int n) {
        for (int i = 0; i < n; i++)
            System.out.println("PAN Number: " + ll.get(i).number);

    }
    LinkedList < PAN > reverse(LinkedList < PAN > ll, int n) {
        LinkedList < PAN > rev = new LinkedList < PAN > ();
        for (int i = n - 1; i >= 0; i--)
            rev.add(ll.get(i));
        return rev;
    }
}
public class Lab8 {
    public static void main(String[] args) {
        int n;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of PAN Records: ");
        n = sc.nextInt();
        PAN p1[] = new PAN[n];
        LinkedList < PAN > arr = new LinkedList < PAN > ();
        LinkedList < PAN > rev = new LinkedList < PAN > ();
        for (int i = 0; i < n; i++) {
            p1[i] = new PAN();
            p1[i].get();
            arr.add(p1[i]);
        }
        System.out.println("Printing the Original Linked List: \n");
        p1[0].display(arr, n);
        rev = p1[0].reverse(arr, n);
        System.out.println("\nPrinting the Second Linked List: \n");
        p1[0].display(rev, n);
    }
}
```



```
run:
Enter the number of PAN Records: 3
Enter the PAN Number: AKTA32600P
Enter the PAN Number: BFDS89012F
Enter the PAN Number: CVBN90014B
Printing the Original Linked List:

PAN Number: AKTA32600P
PAN Number: BFDS89012F
PAN Number: CVBN90014B

Printing the Second Linked List:

PAN Number: CVBN90014B
PAN Number: BFDS89012F
PAN Number: AKTA32600P
BUILD SUCCESSFUL (total time: 36 seconds)
```

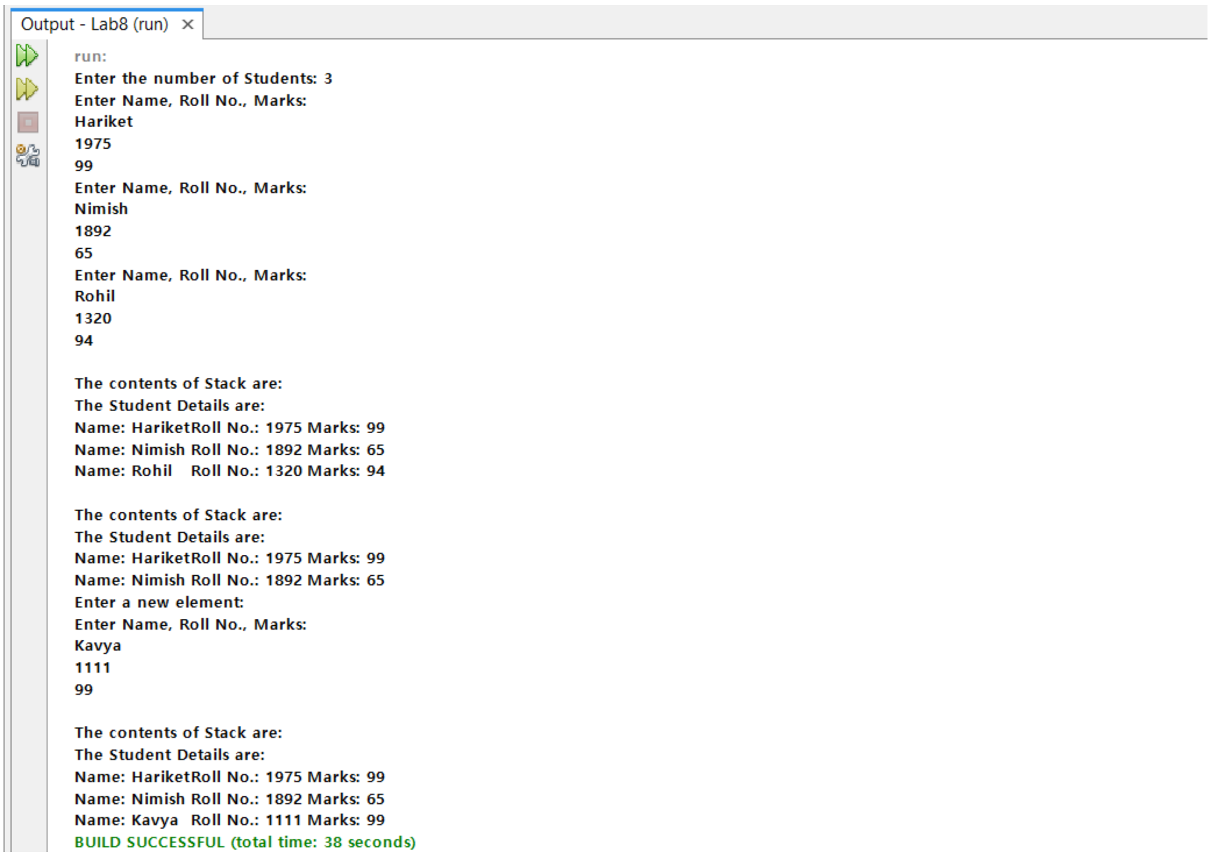
4. Write a java program Stack using collection framework (push, pop methods).

```java
package lab8;
import java.io.*;
import java.util.*;

class Student {
    int roll_no, marks;
    String name;

    void get(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Name, Roll No., Marks: ");
        name = sc.nextLine();
        roll_no = sc.nextInt();
        marks = sc.nextInt();
    }

    void display(Stack<Student> ll, int n){
        System.out.println("The Student Details are:");
        for(int i=0; i<n; i++)
            System.out.println("Name: "+ll.get(i).name+"\tRoll No.:
"+ll.get(i).roll_no+"\tMarks: "+ll.get(i).marks);
    }
}
public class Lab8{
    public static void main(String[] args){
        int n;
        System.out.print("Enter the number of Students: ");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        Stack<Student> arr = new Stack<Student>();
        Student s1[] = new Student[n];
        for(int i=0; i<n; i++){
            s1[i] = new Student();
            s1[i].get();
            arr.push(s1[i]);
        }

        System.out.println("\nThe contents of Stack are: ");
        s1[0].display(arr,n);
        arr.pop();
        n-=1;
```



```
         System.out.println("\nThe contents of Stack are: ");
        s1[0].display(arr,n);
        System.out.println("Enter a new element: ");
        Student add = new Student();
        add.get();
        arr.push(add);
        n += 1;
        System.out.println("\nThe contents of Stack are: ");
        s1[0].display(arr,n);
    }
}
```

OUTPUT:



Output - Lab8 (run)

```
run:
Enter the number of Students: 3
Enter Name, Roll No., Marks:
Hariket
1975
99
Enter Name, Roll No., Marks:
Nimish
1892
65
Enter Name, Roll No., Marks:
Rohil
1320
94

The contents of Stack are:
The Student Details are:
Name: HariketRoll No.: 1975 Marks: 99
Name: Nimish Roll No.: 1892 Marks: 65
Name: Rohil   Roll No.: 1320 Marks: 94

The contents of Stack are:
The Student Details are:
Name: HariketRoll No.: 1975 Marks: 99
Name: Nimish Roll No.: 1892 Marks: 65
Enter a new element:
Enter Name, Roll No., Marks:
Kavya
1111
99

The contents of Stack are:
The Student Details are:
Name: HariketRoll No.: 1975 Marks: 99
Name: Nimish Roll No.: 1892 Marks: 65
Name: Kavya  Roll No.: 1111 Marks: 99
BUILD SUCCESSFUL (total time: 38 seconds)
```