

INPUT

```
1 import java.util.*;
2
3 class Node {
4     long phoneNo;
5     String name;
6     String email;
7     String address;
8     Node link;
9
10    Node(long data, String name, String email, String address) {
11        this.phoneNo = data;
12        this.name = name;
13        this.email = email;
14        this.address = address;
15        this.link = null;
16    }
17 }
18
```

```
19 class HashD extends Thread {
    3 usages
20     static long phone;
    2 usages
21     static String name;
    2 usages
22     static String email;
    2 usages
23     static String address;
    6 usages
24     static Node[] hashT = new Node[100];
    4 usages
25     static Node[] stack = new Node[100];
    7 usages
26     static int top = -1;
27
28     public void run() {
29         Node newnode = new Node(phone, name, email, address);
30         long ind = phone % 100;
31         int index = (int) ind;
32         if (hashT[index] == null) {
33             hashT[index] = newnode;
34         } else {
35             Node ptr;
36             ptr = hashT[index];
37             while (ptr.link != null) {
38                 ptr = ptr.link;
39             }
```

```

40         ptr.link = newnode;
41     }
42     top++;
43     stack[top] = newnode;
44 }
45
46 1 usage
47 public static void undoLastContact() {
48     if (top >= 0) {
49         Node lastContact = stack[top];
50         long ind = lastContact.phoneNo % 100;
51         int index = (int) ind;
52         if (hashT[index] == lastContact) {
53             hashT[index] = null;
54             stack[top]=null;
55         } else {
56             Node ptr = hashT[index];
57             while (ptr.link != lastContact) {
58                 ptr = ptr.link;
59             }
60             ptr.link = null;
61             stack[top]=null;
62         }top--;
63         System.out.println("Last contact undone successfully.");
64     } else {System.out.println("No contacts to undo.");
65     }
66 }

```

```

67     public static void modifyLastContact() {
68         if (top >= 0) {
69             Node lastContact = stack[top];
70             Scanner scanner = new Scanner(System.in);
71             System.out.println("Modify Contact:");
72             System.out.print("Enter Name : ");
73             lastContact.name = scanner.nextLine();
74             System.out.print("Enter Email : ");
75             lastContact.email = scanner.nextLine();
76             System.out.print("Enter Address : ");
77             lastContact.address = scanner.nextLine();
78             System.out.print("Enter phone : ");
79             lastContact.phoneNo = scanner.nextLong();
80             System.out.println("Contact modified successfully.");
81         } else {
82             System.out.println("No contacts to modify.");
83         }
84     }
85 }
86
87 ▶ public class Main {
88     2 usages
89     static LinkedList<String> callLog = new LinkedList<>();
90 ▶
91     public static void main(String[] args) {
92         Scanner sc = new Scanner(System.in);
93
94         while (true) { // Infinite loop until user exits
95             System.out.println("Menu");

```

```
95     System.out.println("1. Add new contact");
96     System.out.println("2. Modify existing contact");
97     System.out.println("3. Display all contacts");
98     System.out.println("4. Make a call");
99     System.out.println("5. Display call log");
100    System.out.println("6. Exit");
101    System.out.print("Enter choice : ");
102    int choice = sc.nextInt();
103    sc.nextLine(); // consume newline
104
105    switch (choice) {
106        case 1:
107            addContact(sc);
108            break;
109        case 2:
110            modifyContact();
111            break;
112        case 3:
113            displayAllContacts();
114            break;
115        case 4:
116            makeCall(sc);
117            break;
118        case 5:
119            displayCallLog();
120            break;
121        case 6:
122            System.out.println("Exiting...");
123            System.exit(status: 0);
124            break;
```



```

125         default:
126             System.out.println("Invalid choice");
127     }
128 }
129 }
130

```

1 usage

```

131 @ public static void addContact(Scanner sc) {
132     System.out.print("Enter Name : ");
133     String name = sc.nextLine();
134     System.out.print("Enter Email : ");
135     String email = sc.nextLine();
136     System.out.print("Enter Address : ");
137     String address = sc.nextLine();
138     System.out.print("Enter phone : ");
139     long phone = sc.nextLong();
140     saveDetails(name, email, phone, address);
141 }
142

```

1 usage

```

143 public static void saveDetails(String name, String email, long phoneNo, String address) {
144     synchronized (HashD.class) {
145         HashD.phone = phoneNo;
146         HashD.name = name;
147         HashD.email = email;
148         HashD.address = address;
149         HashD hashing = new HashD();
150         hashing.start();

```

```

151     }
152 }
153
154     1 usage
155     public static void modifyContact() {
156         synchronized (HashD.class) {
157             Scanner sc = new Scanner(System.in);
158             System.out.println("Modify Contact:");
159             System.out.println("1. Undo last contact");
160             System.out.println("2. Modify last contact");
161             System.out.print("Enter choice : ");
162             int choice = sc.nextInt();
163             switch (choice) {
164                 case 1:
165                     HashD.undoLastContact();
166                     break;
167                 case 2:
168                     HashD.modifyLastContact();
169                     break;
170                 default:
171                     System.out.println("Invalid choice");
172             }
173         }
174     }
175
176     1 usage
177     public static void displayAllContacts() {
178         System.out.println("Displaying all contacts:");

```



```

177     for (Node contact : HashD.stack) {
178         if (contact != null) {
179             System.out.println("Name: " + contact.name);
180             System.out.println("Phone: " + contact.phoneNo);
181             System.out.println("Email: " + contact.email);
182             System.out.println("Address: " + contact.address);
183             System.out.println("-----");
184         }
185     }
186 }
187
188 @ 1 usage
189 public static void makeCall(Scanner sc) {
190     System.out.print("Enter the number you want to call: ");
191     String number = sc.next();
192     callLog.add("Called: " + number + " at " + new Date());
193     System.out.println("Call made successfully to " + number);
194 }
195
196 1 usage
197 public static void displayCallLog() {
198     System.out.println("Displaying call log:");
199     for (String call : callLog) {
200         System.out.println(call);
201     }
202 }

```

OUTPUT

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
```

Menu

1. Add new contact
2. Modify existing contact
3. Display all contacts
4. Make a call
5. Display call log
6. Exit

Enter choice : 1

Enter Name : ABHI

Enter Email : sabhilash.0804@gmail.com

Enter Address : akkulam,trivandrum

Enter phone : 1234567890

Menu

1. Add new contact
2. Modify existing contact
3. Display all contacts
4. Make a call
5. Display call log
6. Exit

Enter choice : 1

Enter Name : JITHU

Enter Email : sabhijith.0804@gmail.com

Enter Address : akkulam,trivandrum

Enter phone : 7894561232

Menu

1. Add new contact
2. Modify existing contact
3. Display all contacts
4. Make a call
5. Display call log
6. Exit

Enter choice : 3

Displaying all contacts:

Name: ABHI

Phone: 1234567890

Email: sabhilash.0804@gmail.com

Address: akkulam,trivandrum

Name: JITHU

Phone: 7894561232

Email: sabhijith.0804@gmail.com

Address: akkulam,trivandrum

Menu

1. Add new contact
2. Modify existing contact
3. Display all contacts
4. Make a call
5. Display call log
6. Exit

Enter choice : 4

Enter the number you want to call: 8845679123

Call made successfully to 8845679123

Menu

1. Add new contact
2. Modify existing contact
3. Display all contacts
4. Make a call
5. Display call log
6. Exit

Enter choice : 2

Modify Contact:

1. Undo last contact
2. Modify last contact

Enter choice : 1

Last contact undone successfully.

Menu

1. Add new contact
2. Modify existing contact
3. Display all contacts
4. Make a call
5. Display call log
6. Exit

Enter choice : 3

Displaying all contacts:

Name: ABHI

Phone: 1234567890

Email: sabhilash.0804@gmail.com

Address: akkulam,trivandrum

Menu

1. Add new contact
2. Modify existing contact
3. Display all contacts
4. Make a call
5. Display call log
6. Exit

Enter choice : 5

Displaying call log:

Called: 8845679123 at Sat Feb 17 11:43:52 IST 2024

Menu

1. Add new contact
2. Modify existing contact
3. Display all contacts
4. Make a call
5. Display call log
6. Exit

Enter choice : 6

Exiting...

Process finished with exit code 0