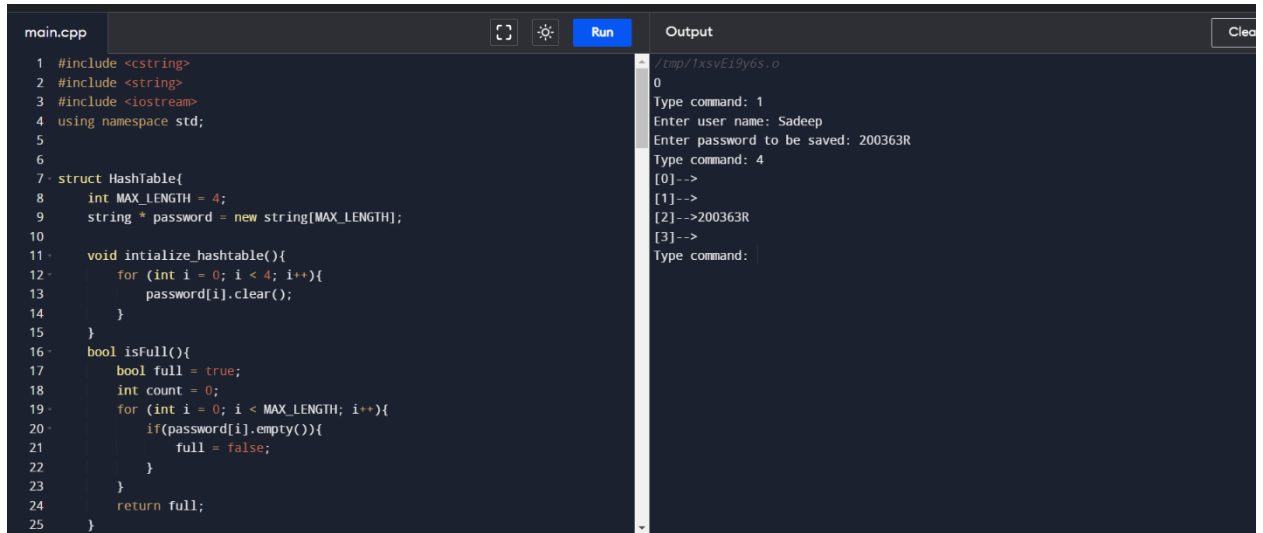


Section 01 :

1)

2)



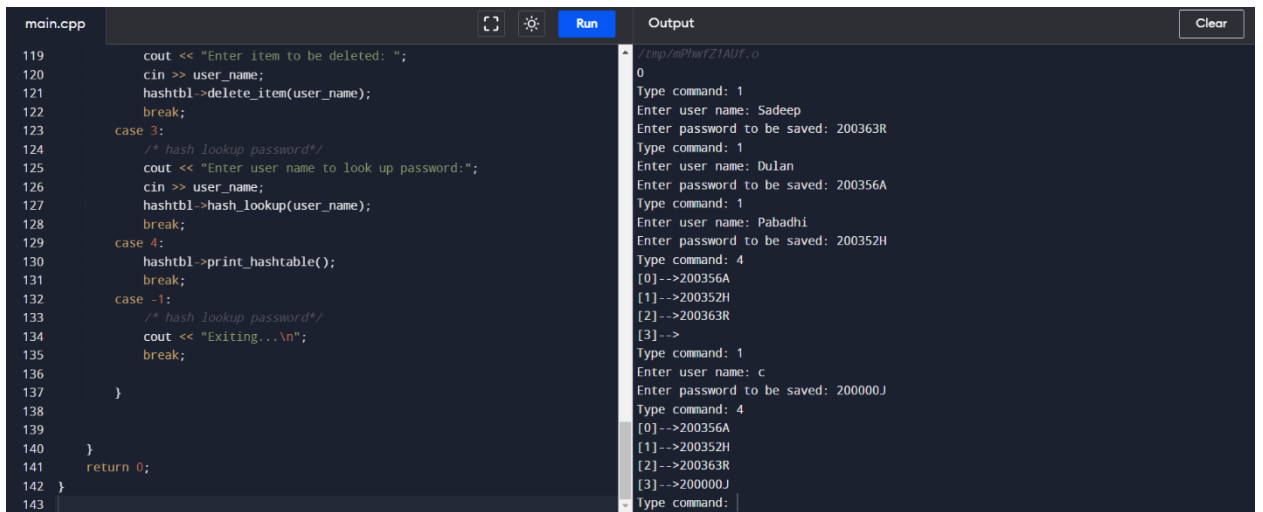
The screenshot shows a C++ IDE with a file named `main.cpp`. The code defines a `HashTable` structure with a `MAX_LENGTH` of 4 and an array of `string` objects. It includes a `initialize_hashtable()` function that clears the array. The output window shows the program's execution, including prompts for user name and password, and the state of the hash table after initialization.

```
main.cpp 1 #include <cstring>
2 #include <string>
3 #include <iostream>
4 using namespace std;
5
6
7 struct HashTable{
8     int MAX_LENGTH = 4;
9     string * password = new string[MAX_LENGTH];
10
11     void initialize_hashtable(){
12         for (int i = 0; i < 4; i++){
13             password[i].clear();
14         }
15     }
16     bool isFull(){
17         bool full = true;
18         int count = 0;
19         for (int i = 0; i < MAX_LENGTH; i++){
20             if(password[i].empty()){
21                 full = false;
22             }
23         }
24         return full;
25     }
26 }
```

Output

```
/tmp/TxsvE19y6s.o
0
Type command: 1
Enter user name: Sadeep
Enter password to be saved: 200363R
Type command: 4
[0]-->
[1]-->
[2]-->200363R
[3]-->
Type command:
```

3)



The screenshot shows a C++ IDE with a file named `main.cpp`. The code implements a `HashTable` structure with methods for deleting items, looking up passwords, printing the hash table, and exiting. The output window shows the program's execution, including prompts for user name and password, and the state of the hash table after various operations.

```
main.cpp 119     cout << "Enter item to be deleted: ";
120     cin >> user_name;
121     hashtable->delete_item(user_name);
122     break;
123     case 3:
124         /* hash lookup password*/
125         cout << "Enter user name to look up password:";
126         cin >> user_name;
127         hashtable->hash_lookup(user_name);
128         break;
129     case 4:
130         hashtable->print_hashtable();
131         break;
132     case -1:
133         /* hash lookup password*/
134         cout << "Exiting...\n";
135         break;
136     }
137 }
138
139
140 }
141 return 0;
142 }
143 }
```

Output

```
/tmp/mPhetZ1AUf.o
0
Type command: 1
Enter user name: Sadeep
Enter password to be saved: 200363R
Type command: 1
Enter user name: Dulan
Enter password to be saved: 200356A
Type command: 1
Enter user name: Pabadhi
Enter password to be saved: 200352H
Type command: 4
[0]-->200356A
[1]-->200352H
[2]-->200363R
[3]-->
Type command: 1
Enter user name: c
Enter password to be saved: 200000J
Type command: 4
[0]-->200356A
[1]-->200352H
[2]-->200363R
[3]-->200000J
Type command:
```

4)

```
/* insert the new item */
cout << "Enter user name: ";
cin >> user_name;
cout << "Enter password to be saved: ";
cin >> password;
hashtbl->insert(user_name,password);
break;
case 2:
/* delete item */
cout << "Enter item to be deleted: ";
cin >> user_name;
hashtbl->delete_item(user_name);
break;
case 3:
/* hash lookup password */
cout << "Enter user name to look up password:";
cin >> user_name;
hashtbl->hash_lookup(user_name);
break;
case 4:
hashtbl->print_hashtable();
break;
case -1:
/* hash lookup password */
cout << "Exiting...\n";
```

```
Type command: 1
Enter user name: Pabadhi
Enter password to be saved: 200352H
Type command: 4
[0]-->200356A
[1]-->200352H
[2]-->200363R
[3]-->
Type command: 1
Enter user name: c
Enter password to be saved: 200000J
Type command: 4
[0]-->200356A
[1]-->200352H
[2]-->200363R
[3]-->200000J
Type command: 2
Enter item to be deleted: Sadeep
User deleted
Type command: 4
[0]-->200356A
[1]-->200352H
[2]-->
[3]-->200000J
Type command: |
```

5) In this table, we can not save multiple passwords in same locations.

Chains with linked lists can be used as a solution for this problem. Then we can save multiple passwords in same location.

Section 2 :

1)

2)

```
main.cpp
169     cout << "Enter password to be saved: ";
170     cin >> password;
171     hashtbl->insert(user_name,password);
172     break;
173     case 2:
174         /* hash lookup password */
175         cout << "Enter user name to look up password:";
176         cin >> user_name;
177         hashtbl->hash_lookup(user_name);
178         break;
179     case 3:
180         hashtbl->print_hashtable();
181         break;
182     case -1:
183         /* hash lookup password */
184         hashtbl->print_hashtable();
185         cout << "Exiting...\n";
186         break;
187     }
188 }
189 }
190 }
191 return 0;
192 }
193 }
```

```
Output
/tmp/hxPFSB21E6.o
0
Type command: 1
Enter user name: Sadeep
Enter password to be saved: 200363R
Type command: 1
Enter user name: Sadeep
Enter password to be saved: 200363R
Type command: 3
[0]-->[]
[1]-->[]
[2]-->[Sadeep, Sadeep, ]
[3]-->[]
Type command:
```

3)

```
main.cpp
169     cout << "Enter password to be saved: ";
170     cin >> password;
171     hashtable->insert(user_name,password);
172     break;
173     case 2:
174         /* hash lookup password*/
175         cout << "Enter user name to look up password:";
176         cin >> user_name;
177         hashtable->hash_lookup(user_name);
178         break;
179     case 3:
180         hashtable->print_hashtable();
181         break;
182     case -1:
183         /* hash lookup password*/
184         hashtable->print_hashtable();
185         cout << "Exiting...\n";
186         break;
187
188     }
189 }
190
191 return 0;
192 }
193
```

```
Output
/tmp/hxPFsB21E6.o
0
Type command: 1
Enter user name: Sadeep
Enter password to be saved: 200363R
Type command: 1
Enter user name: Sadeep
Enter password to be saved: 200363R
Type command: 3
[0]-->[]
[1]-->[]
[2]-->[Sadeep, Sadeep, ]
[3]-->[]
Type command: 1
Enter user name: Tharusha
Enter password to be saved: 200449L
Type command: 1
Enter user name: Dulan
Enter password to be saved: 200356A
Type command: 1
Enter user name: Anuki
Enter password to be saved: 200445L
Linked List reached MAX CAP!
Type command:
```

Github link : <https://github.com/SadeepRathnayaka/DSA-Labs/tree/main>