

## Assisted Practice 12.1: Element Search in Sorted List

**Problem Scenario:** Write a program to illustrate the different ways to handle a list.

**Objective:** In this demonstration, we will learn how to handle a list.

**Input List:**

```
["Ryan","Adam","Anna","Robert","Zane","Mike","Ross","Samantha","Jessica","Harvey",  
"Luious","Rachel"]
```

**Expected Output:**

The sorted list of employees is given below:

```
['Adam', 'Anna', 'Harvey', 'Jessica', 'Luious', 'Mike', 'Rachel', 'Robert', 'Ross', 'Ryan',  
'Samantha', 'Zane']
```

Enter the employee's name you wish to search: Adam

Adam is present in the given list.

**Steps to Perform:**

**Step 1:** Log in to your LMS account

**Step 2:** Open the course “**Big data Hadoop and Spark developer**”

(Note: The course name reflects depending on the program purchased.)

**Step 3:** On the left side, click on the “**PRACTICE LABS**” tab and click on the “**LAUNCH LAB**” button

**Big Data Hadoop and Spark Developer**  
52% Self-Learning Videos Watched | 0/3 Projects Done

**BDH**

IMP: Dear learner,  
Please note: This lab is configured based on the curriculum covered during the live virtual classes.  
All details pertaining to the exercises in this lab are provided in the e-books available in your LMS account.

Instructions:

- When you go to the practice Lab page click on the LAUNCH LAB button and it would redirect you to the login credential page.
- To start a Cloudera manager, click on the Auth URL and enter the username and password as given.
- Similarly, for the web console, navigate to the Auth URL, and enter the credentials.
- For FTP, click on the URL and enter the credentials.

Tools:

Cluster Setup

FTP ,Cloudera Infrastructure ,Hadoop 3.x ,HDFS ,YARN ,Kafka ,Flume ,Sqoop ,Pig ,Hive ,Hbase ,Scala ,Spark ,Spark - Core ,Spark SQL ,Spark GraphX ,Spark .ML ,Cloudera Manager

[You can download the Lab Guides from HERE](#)

Your Labs are ready. **LAUNCH LAB**

**Step 4:** Again, click on the “**LAUNCH LAB**” button

**Your Lab Setup is Ready!**

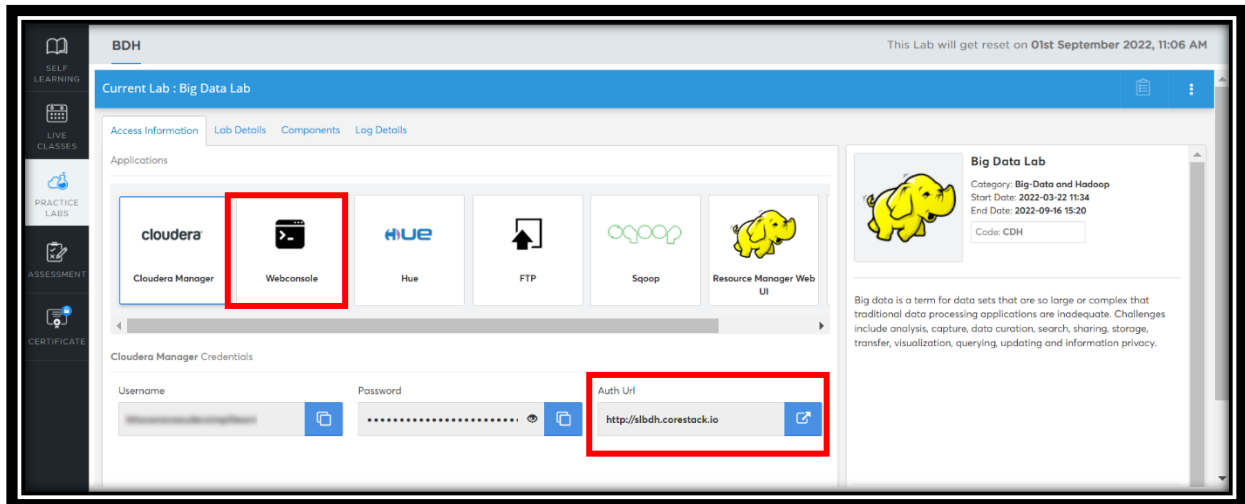
This Lab will get reset on  
**26th July 2022, 10:02 PM**

To ensure continued practice, the reset date is automatically set to **120 days from the last lab start date**. You'll lose all the lab progress on reset activity. Take a backup of your code if you are going to be away for more than 120 days.

**LAUNCH LAB**

**Note:** Practice labs are disabled on course expiry date by default.

**STEP 5:** Click on “**Webconsole**” and click on the “**Auth Url**”



**Step 6:** Copy the **"Username"** and the **"Password"** provided to log in to the Web console

**Step 7:** Paste the **"Username"** and the **"Password"** on the console and click on Enter

**Note:** The password will not be visible when pasted on the console.



**Step 8:** Create a python file

**Command:**

vi search.py

The below screen appears:



### Step 9: Perform the tasks

#### 9.1 Save the given input list in a variable and perform the sorting operation

##### Command:

```
my_list =  
["Ryan","Adam","Anna","Robert","Zane","Mike","Ross","Samantha","Jessica"  
,"Harvey","Luious","Rachel"]  
  
my_list = sorted(my_list)
```

```
my_list = ["Ryan","Adam","Anna","Robert","Zane","Mike","Ross","Samantha","Jessica","Harvey","Luious","Rachel"]  
my_list = sorted(my_list)  
~  
~  
~
```

#### 9.2 Display the sorted list

##### Command:

```
print("The sorted list of employees is given below:")  
  
print(my_list)
```

```
my_list = ["Ryan","Adam","Anna","Robert","Zane","Mike","Ross","Samantha","Jessica","Harvey","Luious","Rachel"]  
my_list = sorted(my_list)  
print("The sorted list of employees is given below:")  
print(my_list)  
~  
~
```

**Command:**

```
python3 search.py
```

```
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ python3 search.py
The sorted list of employees is given below:
['Adam', 'Anna', 'Harvey', 'Jessica', 'Luiious', 'Mike', 'Rachel', 'Robert', 'Ross', 'Ryan', 'Samantha', 'Zane']
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$
```

### 9.3 Take the input from the user to search an element

**Command:**

```
search_elem = input(str("Enter the employee name you wish to search:"))
```

### 9.4 Create a function that performs the searching operation on the list and returns a true or false value

### 9.5 Display "Element Found" if the result is true and "Element not Found" if the result is false

**Command:**

```
def search (my_list, search_elem):

    for i in range(len(my_list)):

        if my_list[i] == search_elem:

            return True

    return False

if(search(my_list, search_elem)):

    print (search_elem+ "Employee is present in the given list")

else:

    print (search_elem+ "Employee is not present in the given list")
```

```
my_list = ["Ryan", "Adam", "Anna", "Robert", "Zane", "Mike", "Ross", "Samantha", "Jessica", "Harvey", "Luious", "Rachel"]
my_list = sorted(my_list)
print("The sorted list of employees is given below:")
print(my_list)

search_elem = input(str("Enter the employee name you wish to search:"))

def search (my_list, search_elem):
    for i in range(len(my_list)):
        if my_list[i] == search_elem:
            return True
    return False

if(search(my_list, search_elem)):
    print (search_elem+ "Employee is present in the given list")
else:
    print (search_elem+ "Employee is not present in the given list")
```

## 9.6 Run the Python script to check the output

```
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ python3 search.py
The sorted list of employees is given below:
['Adam', 'Anna', 'Harvey', 'Jessica', 'Luious', 'Mike', 'Rachel', 'Robert', 'Ross', 'Ryan', 'Samantha', 'Zane']
Enter the employee name you wish to search:Adam
AdamEmployee is present in the given list
```