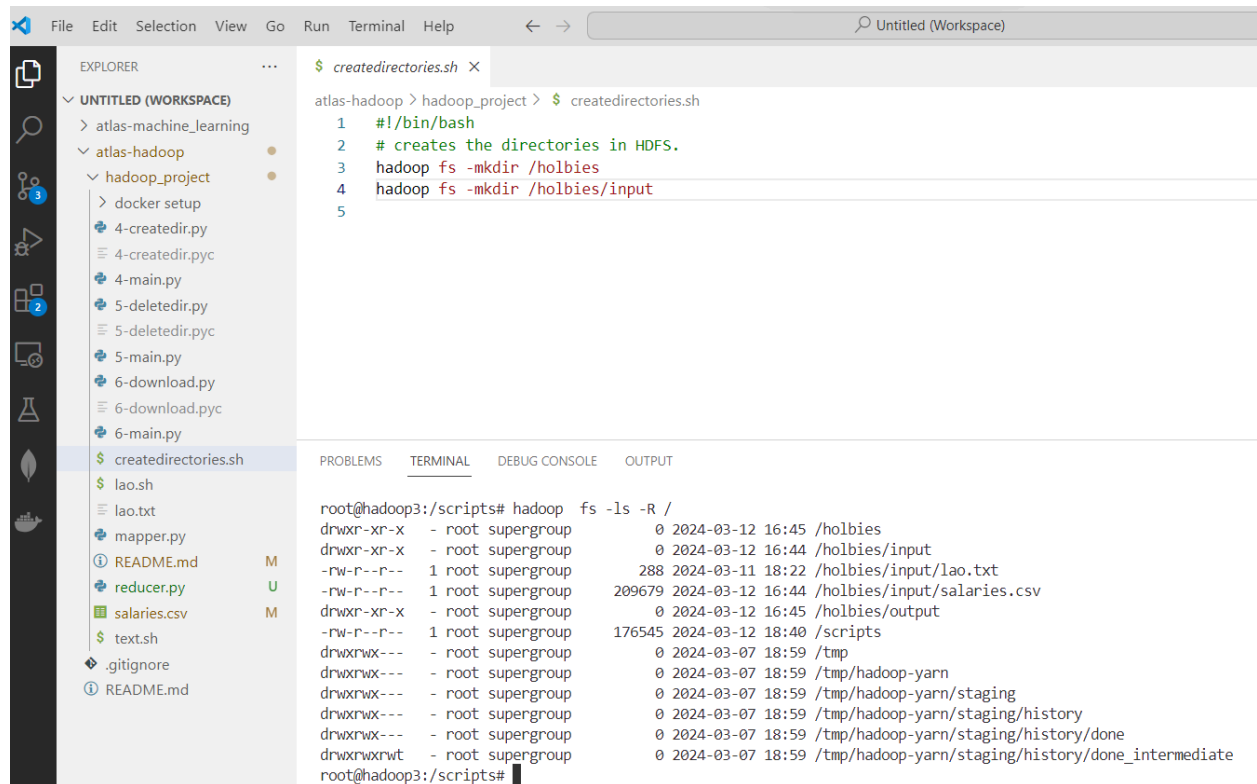


Tasks

0. HDFS with BASH (1)

Run `./createdirectories.sh` to create directories:



```
File Edit Selection View Go Run Terminal Help
Untitled (Workspace)

EXPLORER
  UNTITLED (WORKSPACE)
    atlas-machine_learning
      atlas-hadoop
        hadoop_project
          docker setup
          4-createdir.py
          4-createdir.pyc
          4-main.py
          5-deletedir.py
          5-deletedir.pyc
          5-main.py
          6-download.py
          6-download.pyc
          6-main.py
          $ createdirectories.sh
          $ lao.sh
          lao.txt
          mapper.py
          README.md
          reducer.py
          salaries.csv
          $ text.sh
          .gitignore
          README.md

$ createdirectories.sh X
atlas-hadoop > hadoop_project > $ createdirectories.sh
1 #!/bin/bash
2 # creates the directories in HDFS.
3 hadoop fs -mkdir /holbies
4 hadoop fs -mkdir /holbies/input
5

PROBLEMS TERMINAL DEBUG CONSOLE OUTPUT
root@hadoop3:/scripts# hadoop fs -ls -R /
drwxr-xr-x - root supergroup 0 2024-03-12 16:45 /holbies
drwxr-xr-x - root supergroup 0 2024-03-12 16:44 /holbies/input
-rw-r--r-- 1 root supergroup 288 2024-03-11 18:22 /holbies/input/lao.txt
-rw-r--r-- 1 root supergroup 209679 2024-03-12 16:44 /holbies/input/salaries.csv
drwxr-xr-x - root supergroup 0 2024-03-12 16:45 /holbies/output
-rw-r--r-- 1 root supergroup 176545 2024-03-12 18:40 /scripts
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn/staging
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn/staging/history
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn/staging/history/done
drwxrwxrwt - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn/staging/history/done_intermediate
root@hadoop3:/scripts#
```

1. HDFS with BASH (2)

Run `./lao.sh` to upload file

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the file structure of the 'hadoop_project' directory, including files like '4-createdir.py', '5-deletedir.py', '6-download.py', '6-main.py', 'lao.txt', 'mapper.py', 'README.md', 'reducer.py', 'salaries.csv', 'text.sh', and '.gitignore'. The 'lao.sh' file is selected. The main editor shows the content of 'lao.sh':

```
1 #!/bin/bash
2 # script to upload a file to a directory on the HDFS.
3 hadoop fs -put lao.txt /holbies/input
4
```

The TERMINAL panel at the bottom shows the command prompt 'root@hadoop3:/scripts#' and the output of the 'hadoop fs -ls /holbies/input' command:

```
root@hadoop3:/scripts# hadoop fs -ls /holbies/input
Found 2 items
-rw-r--r-- 1 root supergroup      288 2024-03-11 18:22 /holbies/input/lao.txt
-rw-r--r-- 1 root supergroup 209679 2024-03-12 16:44 /holbies/input/salaries.csv
root@hadoop3:/scripts#
```

2. HDFS with BASH (3)

Run ./text.sh to display file contents

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the file structure of the 'hadoop_project' directory, including files like '4-createdir.py', '5-deletedir.py', '6-download.py', '6-main.py', 'lao.txt', 'mapper.py', 'README.md', 'reducer.py', 'salaries.csv', 'text.sh', and '.gitignore'. The 'text.sh' file is selected. The main editor shows the content of 'text.sh':

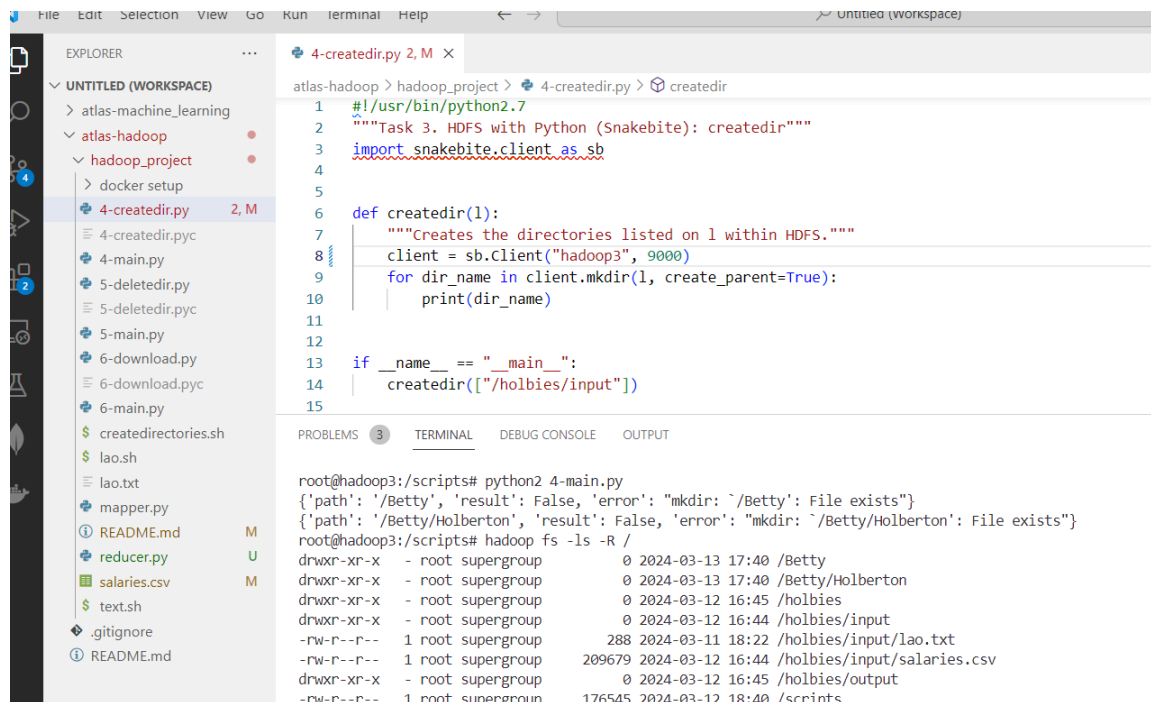
```
1 #!/bin/bash
2 # script that displays the content of a file on the HDFS.
3 hadoop fs -cat /holbies/input/lao.txt
4
```

The TERMINAL panel at the bottom shows the command prompt 'root@hadoop3:/scripts#' and the output of the './text.sh' command:

```
root@hadoop3:/scripts# ./text.sh
Simplicity, patience, compassion. These three are your greatest treasures. Simple in actions and thoughts, you return to the source of being. Patient with both friends and enemies, you accord with the way things are. Compassionate toward yourself, you reconcile all beings in the world.
root@hadoop3:/scripts#
```

3. HDFS with Python (Snakebite): createdir

Run python2 4-main.py to create directories



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'hadoop_project' folder containing '4-createdir.py' (2, M). The code editor shows the content of '4-createdir.py' with the following code:

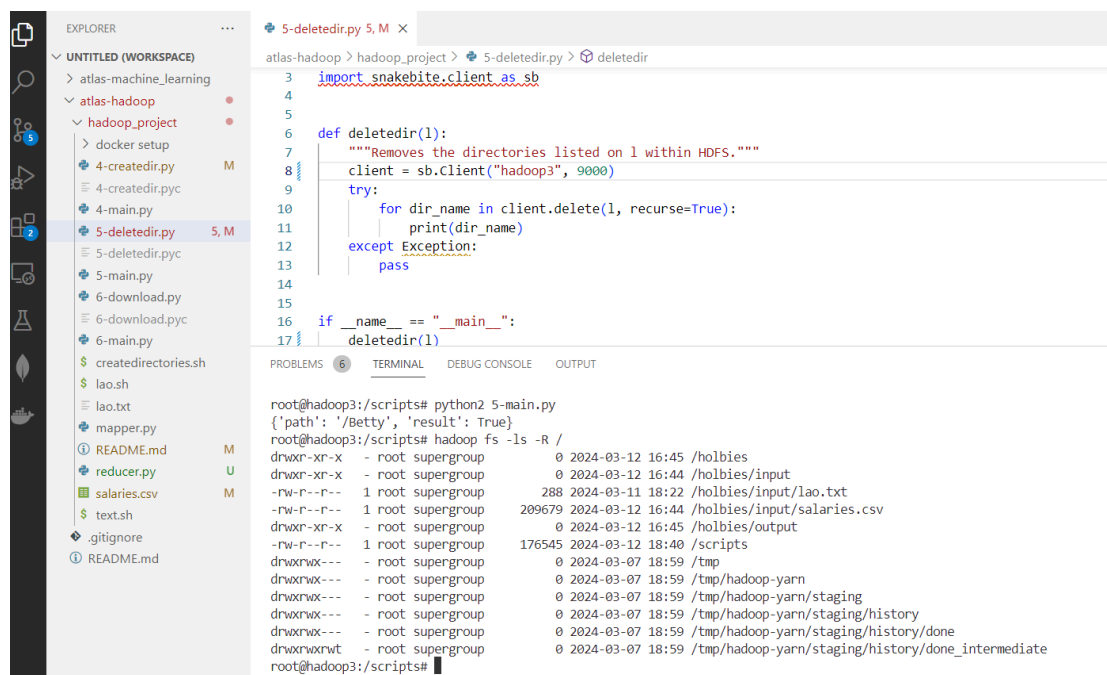
```
1 #!/usr/bin/python2.7
2 """Task 3. HDFS with Python (Snakebite): createdir"""
3 import snakebite.client as sb
4
5
6 def createdir(l):
7     """Creates the directories listed on l within HDFS."""
8     client = sb.Client("hadoop3", 9000)
9     for dir_name in client.mkdir(l, create_parent=True):
10         print(dir_name)
11
12
13 if __name__ == "__main__":
14     createdir(["/holbies/input"])
15
```

The terminal output shows the execution of 'python2 4-main.py' and the resulting HDFS directory listing:

```
root@hadoop3:/scripts# python2 4-main.py
{'path': '/Betty', 'result': False, 'error': "mkdir: '/Betty': File exists"}
{'path': '/Betty/Holberton', 'result': False, 'error': "mkdir: '/Betty/Holberton': File exists"}
root@hadoop3:/scripts# hadoop fs -ls -R /
drwxr-xr-x - root supergroup 0 2024-03-13 17:40 /Betty
drwxr-xr-x - root supergroup 0 2024-03-13 17:40 /Betty/Holberton
drwxr-xr-x - root supergroup 0 2024-03-12 16:45 /holbies
drwxr-xr-x - root supergroup 0 2024-03-12 16:44 /holbies/input
-rw-r--r-- 1 root supergroup 288 2024-03-11 18:22 /holbies/input/lao.txt
-rw-r--r-- 1 root supergroup 209679 2024-03-12 16:44 /holbies/input/salaries.csv
drwxr-xr-x - root supergroup 0 2024-03-12 16:45 /holbies/output
-rw-r--r-- 1 root supergroup 176545 2024-03-12 18:40 /scripts
```

4. HDFS with Python (Snakebite): deletedir

Run python2 5-main.py to delete directories



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'hadoop_project' folder containing '5-deletedir.py' (5, M). The code editor shows the content of '5-deletedir.py' with the following code:

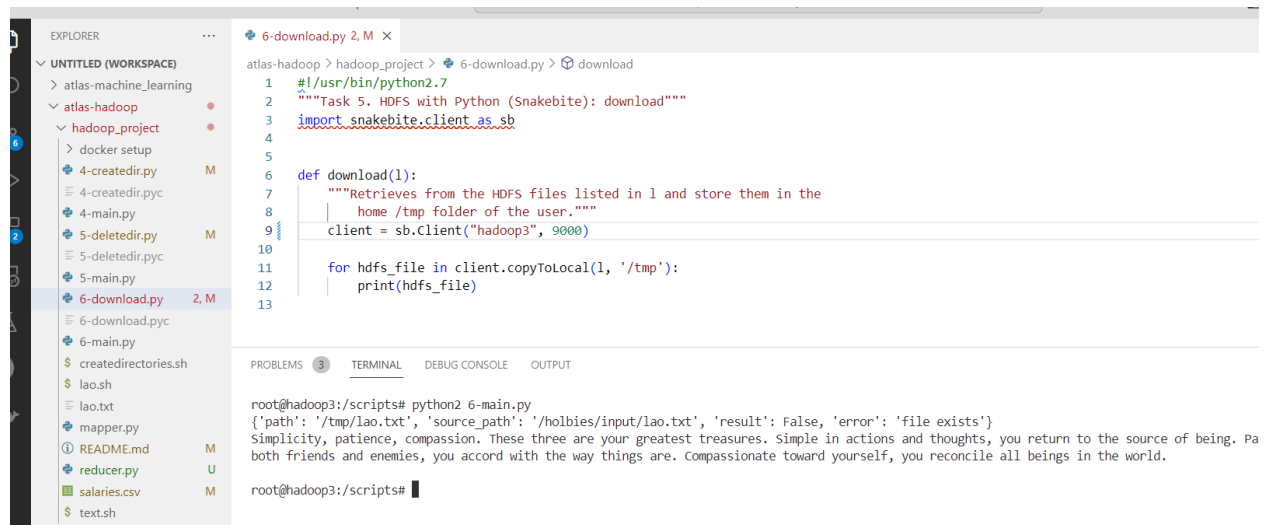
```
3 import snakebite.client as sb
4
5
6 def deletedir(l):
7     """Removes the directories listed on l within HDFS."""
8     client = sb.Client("hadoop3", 9000)
9     try:
10         for dir_name in client.delete(l, recurse=True):
11             print(dir_name)
12     except Exception:
13         pass
14
15
16 if __name__ == "__main__":
17     deletedir(l)
```

The terminal output shows the execution of 'python2 5-main.py' and the resulting HDFS directory listing:

```
root@hadoop3:/scripts# python2 5-main.py
{'path': '/Betty', 'result': True}
root@hadoop3:/scripts# hadoop fs -ls -R /
drwxr-xr-x - root supergroup 0 2024-03-12 16:45 /holbies
drwxr-xr-x - root supergroup 0 2024-03-12 16:44 /holbies/input
-rw-r--r-- 1 root supergroup 288 2024-03-11 18:22 /holbies/input/lao.txt
-rw-r--r-- 1 root supergroup 209679 2024-03-12 16:44 /holbies/input/salaries.csv
drwxr-xr-x - root supergroup 0 2024-03-12 16:45 /holbies/output
-rw-r--r-- 1 root supergroup 176545 2024-03-12 18:40 /scripts
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn/staging
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn/staging/history
drwxrwx--- - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn/staging/history/done
drwxrwxrwt - root supergroup 0 2024-03-07 18:59 /tmp/hadoop-yarn/staging/history/done_intermediate
root@hadoop3:/scripts#
```

5. HDFS with Python (Snakebite): download

Run 6-main.py to call 6-download.py which retrieves from the HDFS files listed in l and store them in the home /tmp folder of the user.



```
EXPLORER
└─ UNTITLED (WORKSPACE)
  └─ atlas-machine_learning
    └─ atlas-hadoop
      └─ hadoop_project
        └─ 6-download.py 2, M
```

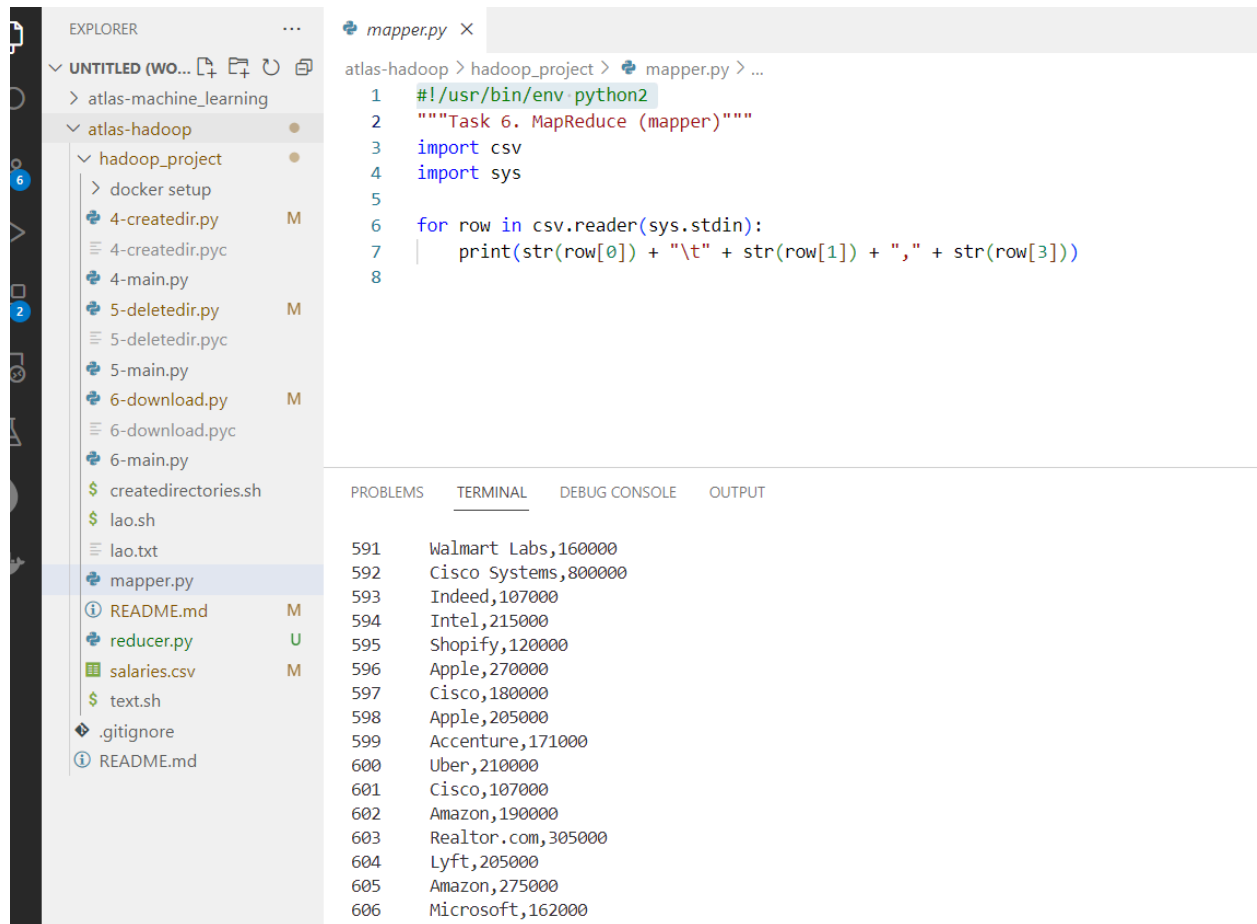
```
atlas-hadoop > hadoop_project > 6-download.py > download
1  #!/usr/bin/python2.7
2  """Task 5. HDFS with Python (Snakebite): download"""
3  import snakebite.client as sb
4
5
6  def download(l):
7      """Retrieves from the HDFS files listed in l and store them in the
8      home /tmp folder of the user."""
9      client = sb.Client("hadoop3", 9000)
10
11     for hdfs_file in client.copyToLocal(l, '/tmp'):
12         print(hdfs_file)
13
```

```
root@hadoop3:/scripts# python2 6-main.py
{'path': '/tmp/lao.txt', 'source_path': '/holbies/input/lao.txt', 'result': False, 'error': 'file exists'}
Simplicity, patience, compassion. These three are your greatest treasures. Simple in actions and thoughts, you return to the source of being. Pa
both friends and enemies, you accord with the way things are. Compassionate toward yourself, you reconcile all beings in the world.

root@hadoop3:/scripts#
```

6. MapReduce (mapper)

Run cat salaries.csv |./mapper.py to run mapper function on the file



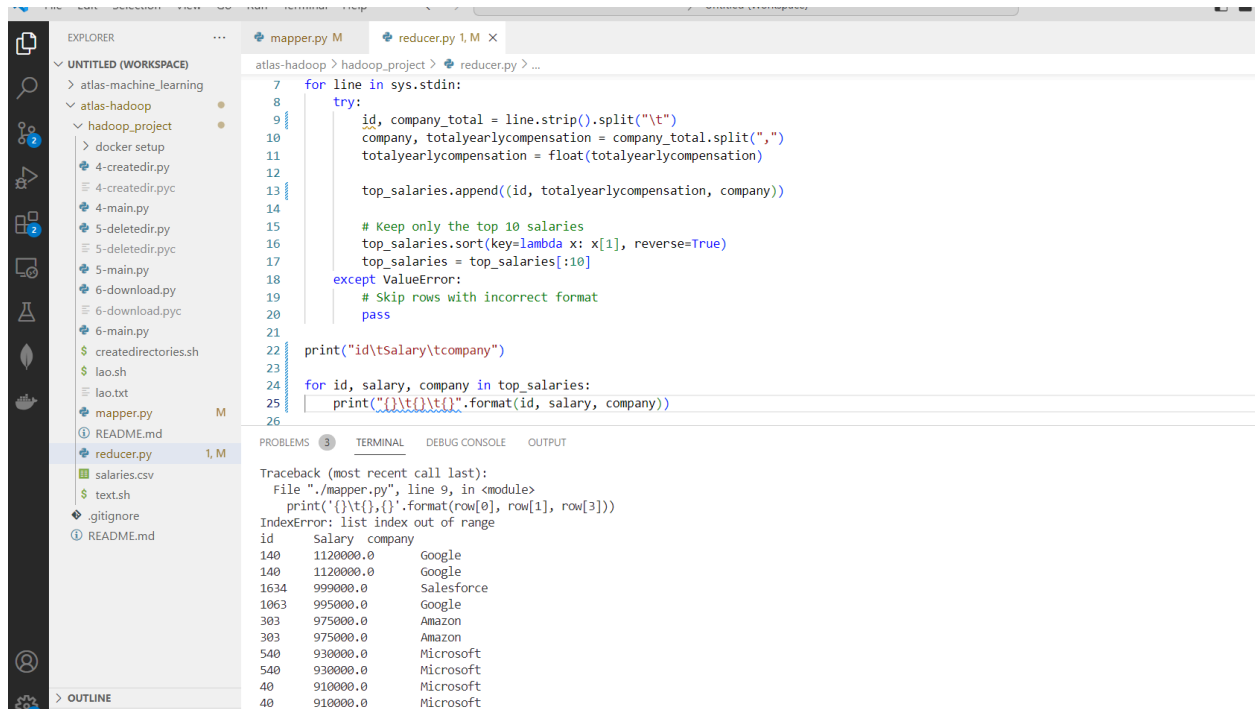
```
EXPLORER
└─ UNTITLED (WO...
  └─ atlas-machine_learning
    └─ atlas-hadoop
      └─ hadoop_project
        └─ 6-main.py
```

```
atlas-hadoop > hadoop_project > mapper.py > ...
1  #!/usr/bin/env python2
2  """Task 6. MapReduce (mapper)"""
3  import csv
4  import sys
5
6  for row in csv.reader(sys.stdin):
7      print(str(row[0]) + "\t" + str(row[1]) + "," + str(row[3]))
8
```

```
591 Walmart Labs,160000
592 Cisco Systems,800000
593 Indeed,107000
594 Intel,215000
595 Shopify,120000
596 Apple,270000
597 Cisco,180000
598 Apple,205000
599 Accenture,171000
600 Uber,210000
601 Cisco,107000
602 Amazon,190000
603 Realtor.com,305000
604 Lyft,205000
605 Amazon,275000
606 Microsoft,162000
```

7. MapReduce (reducer)

Mapred commands didn't work due to environment, used the command: `cat salaries.csv | ./mapper.py | sort | ./reducer.py` to process it



The screenshot shows a VS Code editor with a project named 'atlas-hadoop'. The Explorer sidebar on the left shows the file structure, including 'salaries.csv' and 'reducer.py'. The main editor window displays the code for 'reducer.py', which processes salary data from 'salaries.csv'. The code reads lines from 'sys.stdin', splits them into 'id', 'company_total', and 'company', and appends them to a list 'top_salaries'. It then sorts 'top_salaries' by 'company_total' in descending order and keeps the top 10 salaries. Finally, it prints the top 10 salaries in a tab-separated format.

```
7 for line in sys.stdin:
8     try:
9         id, company_total = line.strip().split("\t")
10        company, totalyearlycompensation = company_total.split(",")
11        totalyearlycompensation = float(totalyearlycompensation)
12
13        top_salaries.append((id, totalyearlycompensation, company))
14
15        # Keep only the top 10 salaries
16        top_salaries.sort(key=lambda x: x[1], reverse=True)
17        top_salaries = top_salaries[:10]
18    except ValueError:
19        # Skip rows with incorrect format
20        pass
21
22    print("id\tSalary\tcompany")
23
24    for id, salary, company in top_salaries:
25        print("{}\t{}\t{}".format(id, salary, company))
26
```

The PROBLEMS panel at the bottom shows a traceback for an 'IndexError: list index out of range' that occurred in 'mapper.py' at line 9. Below the error message, a table displays the top 10 salaries from the 'salaries.csv' file:

id	Salary	company
140	1120000.0	Google
140	1120000.0	Google
1634	999000.0	Salesforce
1063	995000.0	Google
303	975000.0	Amazon
303	975000.0	Amazon
540	930000.0	Microsoft
540	930000.0	Microsoft
40	910000.0	Microsoft
40	910000.0	Microsoft