```
# import the libraries

from datetime import timedelta
# The DAG object; we'll need this to instantiate a DAG
from airflow.models import DAG
# Operators; you need this to write tasks!
from airflow.operators.bash_operator import BashOperator
# This makes scheduling easy
from airflow.utils.dates import days_ago

#defining DAG arguments

# You can override them on a per-task basis during operator
initialization
default_args = {
    'owner': 'dummy_name',
    'start_date': days_ago(0),
    'email': ['pravalithota@gmail.com'],
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
}

# defining the DAG

# define the DAG
dag = DAG(
    'ETL_toll_data',
    default_args=default_args,
    description='Apache Airflow Final Assignment',
    schedule_interval=timedelta(days=1),
)

# define the tasks

# download = BashOperator(
#     task_id='download',
#     bash_command='curl https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBM-DB0250EN-
SkillsNetwork/labs/Final%20Assignment/tolldata.tgz -o
/home/project/airflow/dags/finalassignment/tolldata.tgz',
#     dag=dag,
# )

# define the first task
unzip = BashOperator(
    task_id='unzip',
    bash_command='tar -xzf
/home/project/airflow/dags/finalassignment/tolldata.tgz -C
/home/project/airflow/dags/finalassignment/tolldata',
    dag=dag,
)

# define the second task
extract_data_from_csv = BashOperator(
```

```
    task_id='extract_data_from_csv',
    bash_command='cut -d "," -f1-4
/home/project/airflow/dags/finalassignment/vehicle-data.csv | tr "," " "
> /home/project/airflow/dags/finalassignment/csv_data.csv',
    dag=dag,
)

# define the third task
extract_data_from_tsv = BashOperator(
    task_id='extract_data_from_tsv',
    bash_command='cut -d " " -f5
/home/project/airflow/dags/finalassignment/tollplaza-data.tsv | cut -c
17- > /home/project/airflow/dags/finalassignment/tsv_data.csv',
    dag=dag,
)

# define the fourth task
extract_data_from_fixed_width = BashOperator(
    task_id='extract_data_from_fixed_width',
    bash_command='cut -c 59-
/home/project/airflow/dags/finalassignment/payment-data.txt >
/home/project/airflow/dags/finalassignment/fixed_width_data.csv',
    dag=dag,
)

# define the fifth task
consolidate_data = BashOperator(
    task_id='consolidate_data',
    bash_command='paste
/home/project/airflow/dags/finalassignment/csv_data.csv
/home/project/airflow/dags/finalassignment/tsv_data.csv
/home/project/airflow/dags/finalassignment/fixed_width_data.csv  >
/home/project/airflow/dags/finalassignment/extracted_data.csv',
    dag=dag,
)

# define the sixth task
transform_data = BashOperator(
    task_id='transform_data',
    bash_command='tr -f4 "[a-z]" "[A-Z]" <
/home/project/airflow/dags/finalassignment/ | tr "," " " >
/home/project/airflow/dags/finalassignment/extracted_data.csv >
/home/project/airflow/dags/finalassignment/transformed_data.csv',
    dag=dag,
)


# task pipeline
unzip_data >> extract_data_from_csv >> extract_data_from_tsv >>
extract_data_from_fixed_width >> consolidate_data >> transform_data

#############
#Instructions to run
```

```
# 1. open apache webserver
# 2. Create a python file and add code
# 3. Open terminal and add the following commands:
### a. to set the AIRFLOW_HOME.
### export AIRFLOW_HOME=/home/project/airflow
### echo $AIRFLOW_HOME
### b.  to submit the DAG that was created
### export AIRFLOW_HOME=/home/project/airflow
### cp ETL_toll_data.py $AIRFLOW_HOME/dags
### d. list out all dags
### airflow dags list
### e. monitor by going to the airflow
```