Project Overview:

The project is essentially a terminal-based air-traffic manager that maintains a set of flights and runways at an airport. The system handles many functions such as initialize() ->(is called initialize_runways in my code), submitflight(), cancelflight(), reprioritize(), addrunways(), groundhold(), printactive(), printschedule(), tick(), and lastly quit(). Additionally, the system is handled as a 2-phased system being phase1() and phase2().

Project Structure:

[Gafarov_Shahaddin.zip]:

|----> gatorAirTrafficScheduler.py -> primary file, processes inputs, outputs, and the terminal/console related matters

|----> data_structures.py -> holds the necessary data structures for the code to function properly

|----> logic.py -> all the logical decision making is happening here, and where the functions are being handled

|----> Makefile -> Makefile, running “make” will create an executable called gatorAirTrafficScheduler

Class and Function prototypes:

Inside gatorAirTrafficScheduler.py:

def main() -> surface-level console operation handling the inputs and outputs

-a handler that reads the input file name and creates an output file

- with open(input_file, "r") -> read the input from file, and categorize each line by the function that is being called and the contents within its parentheses

quit()-> handled here instead of inside logic.py, terminating the read and t riggering the write to file

- with open(output_file, "w") -> write the lines from list into the output_file

Inside data_structures.py:

class binaryminheap: -> the binary min heap implementation

def __init__() ->initialize an empty heap

def insert() -> insert a new number into the binary min heap

def getmin() -> get the minimum element(root in other words)

def deletemin() -> delete the minimum element from the heap

class maxpairingheap(): -> the max pairing heap implementation

class pairingheapnode: ->properties of the node(s) are defined here

def __init__() -> assign a number, key, child, left, and right sibling

def __init__() -> inititalize an empty max pairing heap

def insert() -> insert an element into the max pairing heap

def getmax() -> fetch the max element node in the heap(root)

def deletemax() -> delete the max element node from the heap

def increasekey() -> increase the key of the node

def meld() -> melding 2 max pairing heaps, taking them as inputs

def two_pass() -> two-pass merging scheme for the children after their parent node gets deleted

def severnode() -> severs the connection of the node from the rest of the max pairing heap as an independent heap

class flight_class: -> handles the necessary parameters of a flight

flight_id -> must have an integer value for the flight

airline_id -> must have an integer value for the flight

submit_time -> must have an integer value for the flight

priority -> must have an integer value for the flight

duration -> must have an integer value for the flight

status -> data that can be defined later

runway_id -> data that can be defined later

start_time -> data that can be defined later

end_time -> data that can be defined later

Inside logic.py:

__init__() -> start the logic, with time and runways at 0, and call flight_class

phase1() -> for settling the completions, takes the new time as input

phase2() -> rescheduling all the flights that are still unsatisfied

initialize_runways() -> start the system with x amount of runways

submitflight() -> submit a new flight into the system

cancelflight() -> cancl the given flight(by looking flight id)

reprioritize() -> change a flight's priority

addrunways() -> the number of new runways, which will be available at time point "time"

groundhold() -> hold the flights whose airline_id within the given range

printactive() -> prints all flights(active) in system

printschedule() -> print all flights with ETAs within the given range

tick() -> advance time, do settle(phase1) and the reschedule(phase2)

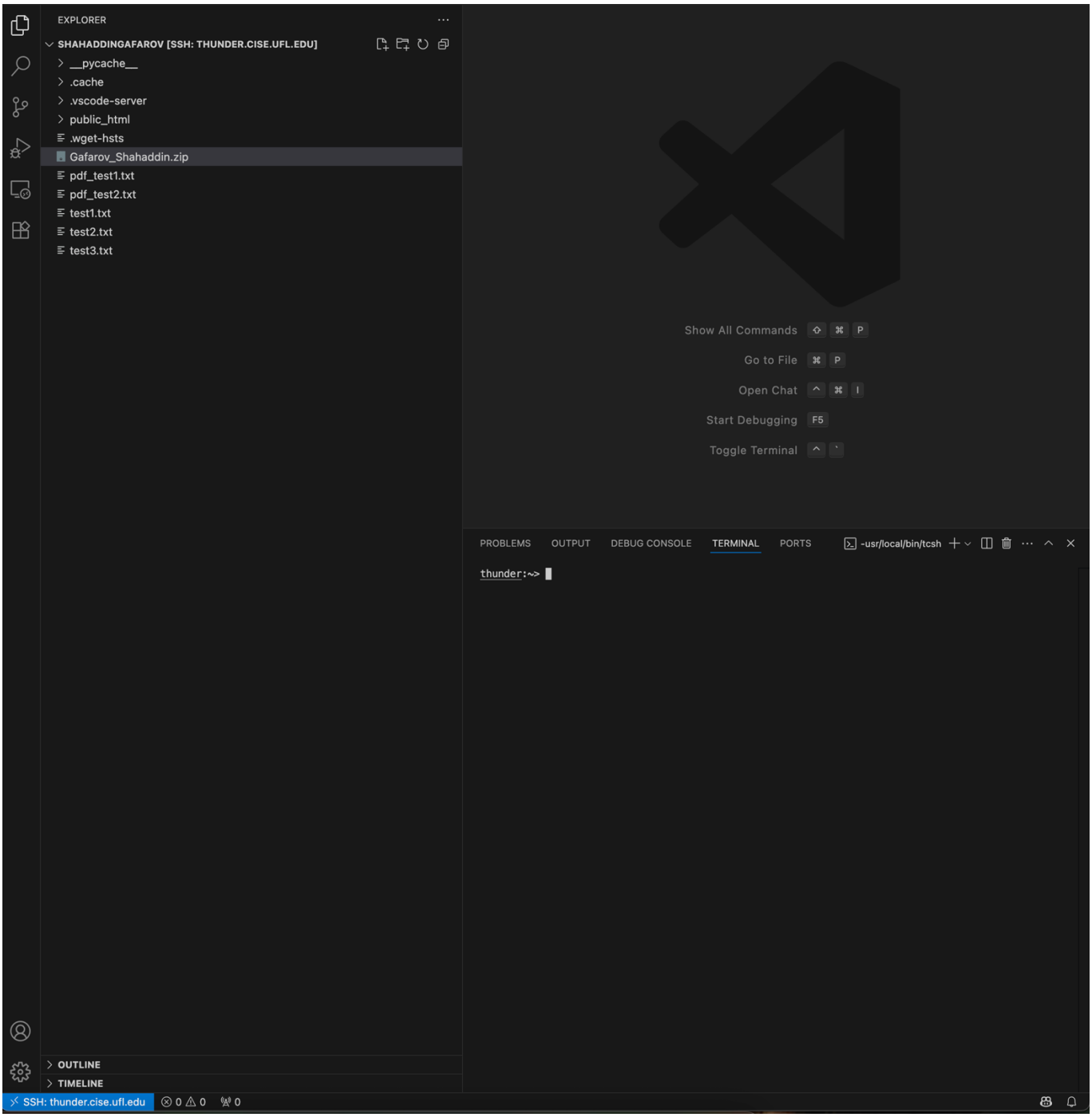
Relevant Notes:

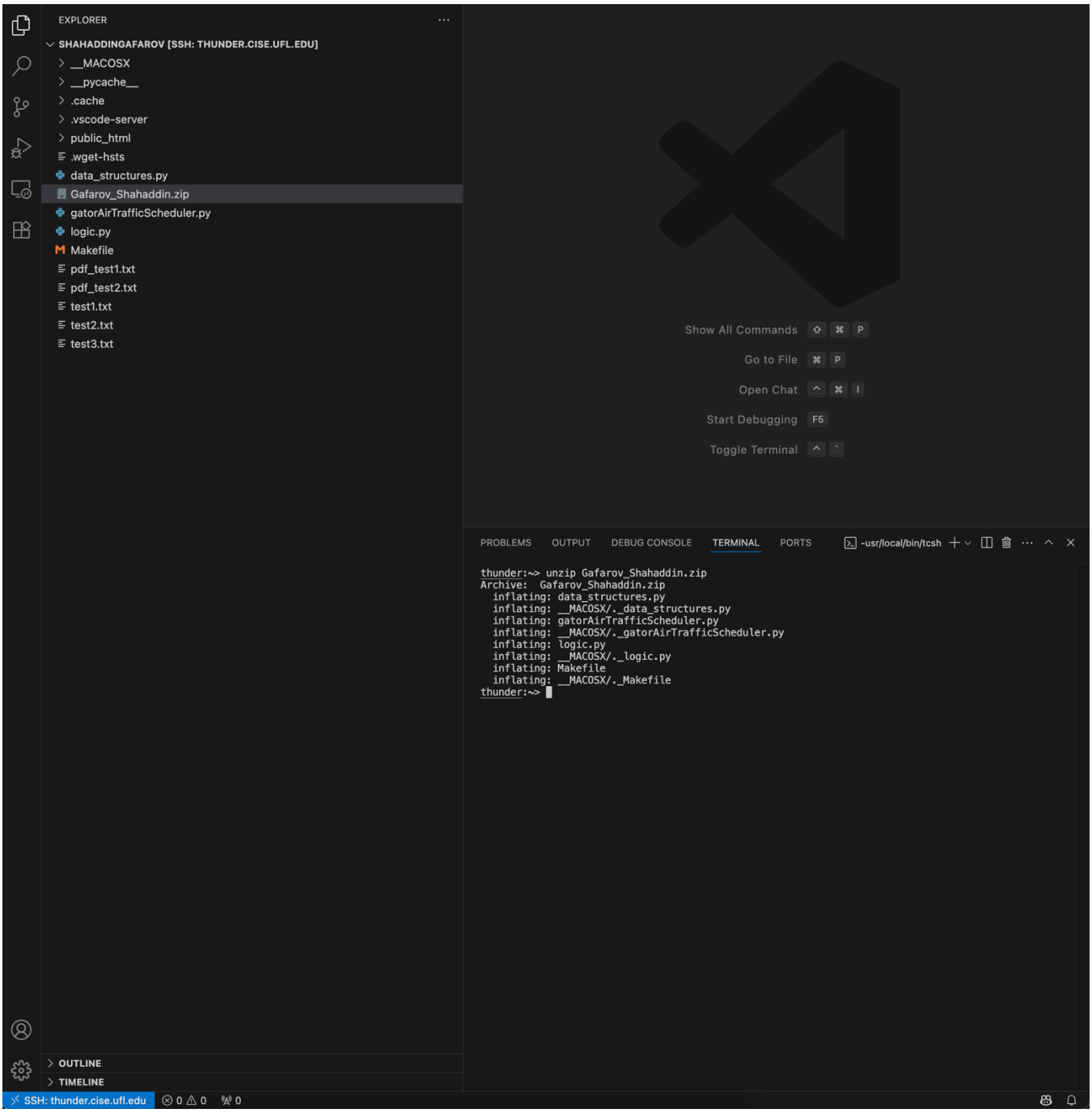
1) You may not see “Advance time to currentTime → settle completions → reschedule unsatisfied” itself directly on some functions of the code. What you will see instead is that such functions start with tick(),which already handles advancing time, settling, and rescheduling operations all at once.

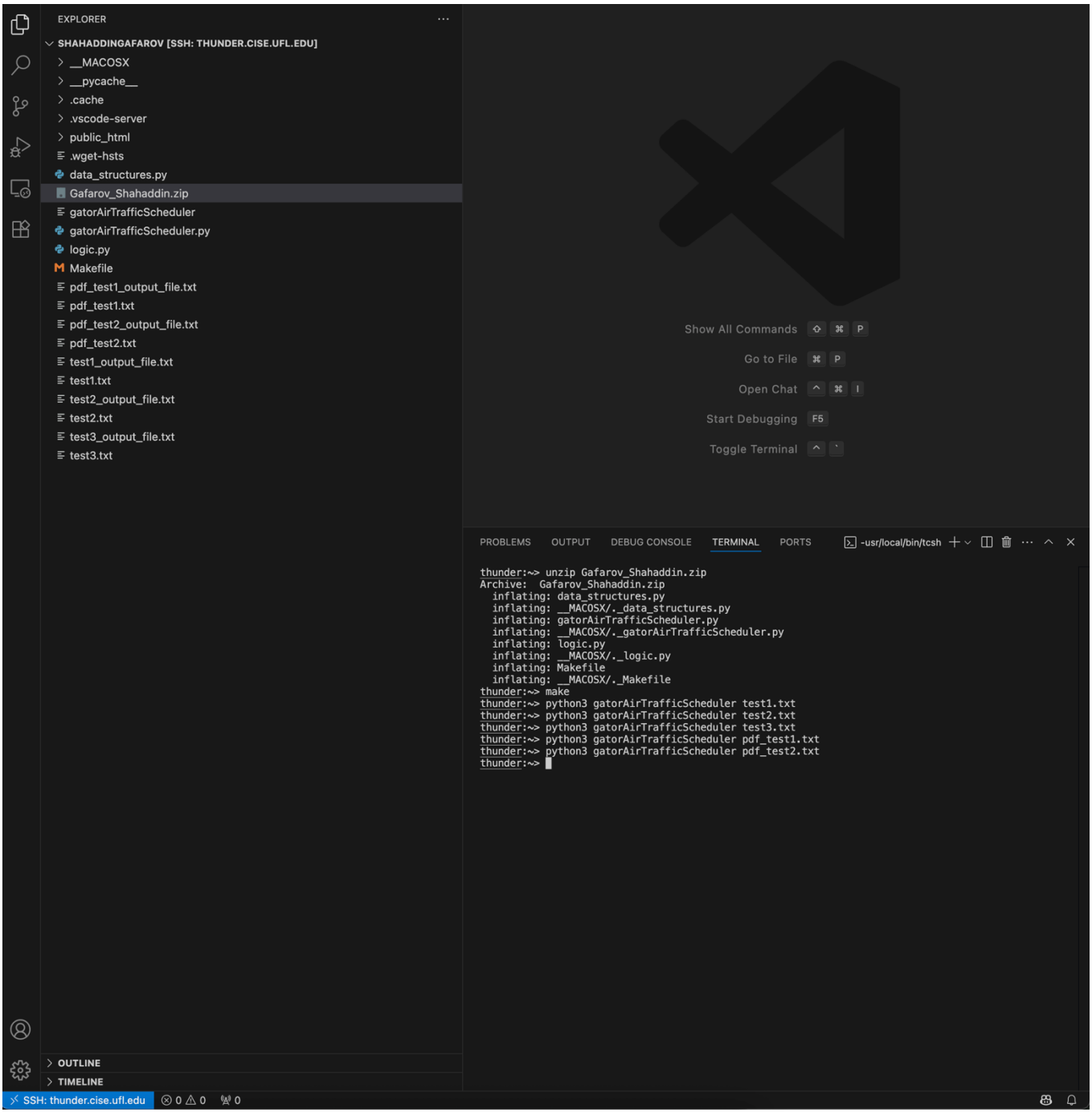
2) For a more detailed explanation that explains what is going on every step, please look at my comments on logic.py, data_structures.py, gatorAirTrafficScheduler.py, and Makefile.

3) Although it is in proper format, as a helper for the TA(or reader), please:

1. Unzip the folder, 2. run “make” within the folder, which will create the executable gatorAirTrafficScheduler, and 3. run python3 gatorAirTrafficScheduler file_name. Assuming there will be a file_name that is a .txt file, where the output will also become a .txt file. test111.txt -> test111_output_file.txt.


EXPLORER
SHAHADDINGAFAROV [SSH: THUNDER.CISE.UFL.EDU]
__pycache__
.cache
.vscode-server
public_html
.wget-hsts
Gafarov_Shahaddin.zip
pdf_test1.txt
pdf_test2.txt
test1.txt
test2.txt
test3.txt
Show All Commands
Go to File
Open Chat
Start Debugging
F5
Toggle Terminal
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
-usr/local/bin/tcsh
thunder:~>
OUTLINE
TIMELINE
SSH: thunder.cise.ufl.edu


EXPLORER
SHAHADDINGAFAROV [SSH: THUNDER.CISE.UFL.EDU]
__MACOSX
__pycache__
.cache
.vscode-server
public_html
.wget-hsts
data_structures.py
Gafarov_Shahaddin.zip
gatorAirTrafficScheduler.py
logic.py
Makefile
pdf_test1.txt
pdf_test2.txt
test1.txt
test2.txt
test3.txt
OUTLINE
TIMELINE
Show All Commands ⇧ ⌘ P
Go to File ⌘ P
Open Chat ^ ⌘ I
Start Debugging F5
Toggle Terminal ^ `
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
-usr/local/bin/tcsh
thunder:~> unzip Gafarov_Shahaddin.zip
Archive: Gafarov_Shahaddin.zip
inflating: data_structures.py
inflating: __MACOSX/._data_structures.py
inflating: gatorAirTrafficScheduler.py
inflating: __MACOSX/._gatorAirTrafficScheduler.py
inflating: logic.py
inflating: __MACOSX/._logic.py
inflating: Makefile
inflating: __MACOSX/._Makefile
thunder:~>
SSH: thunder.cise.ufl.edu
0 0
0


EXPLORER
SHAHADDINGAFAROV [SSH: THUNDER.CISE.UFL.EDU]
__MACOSX
__pycache__
.cache
.vscode-server
public_html
.wget-hsts
data_structures.py
Gafarov_Shahaddin.zip
gatorAirTrafficScheduler
gatorAirTrafficScheduler.py
logic.py
Makefile
pdf_test1_output_file.txt
pdf_test1.txt
pdf_test2_output_file.txt
pdf_test2.txt
test1_output_file.txt
test1.txt
test2_output_file.txt
test2.txt
test3_output_file.txt
test3.txt
Show All Commands ⇧ ⌘ P
Go to File ⌘ P
Open Chat ⌃ ⌘ I
Start Debugging F5
Toggle Terminal ⌃ `
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
-usr/local/bin/tcsh
thunder:~> unzip Gafarov_Shahaddin.zip
Archive: Gafarov_Shahaddin.zip
inflating: data_structures.py
inflating: __MACOSX/._data_structures.py
inflating: gatorAirTrafficScheduler.py
inflating: __MACOSX/._gatorAirTrafficScheduler.py
inflating: logic.py
inflating: __MACOSX/._logic.py
inflating: Makefile
inflating: __MACOSX/._Makefile
thunder:~> make
thunder:~> python3 gatorAirTrafficScheduler test1.txt
thunder:~> python3 gatorAirTrafficScheduler test2.txt
thunder:~> python3 gatorAirTrafficScheduler test3.txt
thunder:~> python3 gatorAirTrafficScheduler pdf_test1.txt
thunder:~> python3 gatorAirTrafficScheduler pdf_test2.txt
thunder:~>
OUTLINE
TIMELINE
SSH: thunder.cise.ufl.edu
0 0 0

pdf_test1.txt

```
Initialize(2)
SubmitFlight(201, 1, 0, 5, 4)
SubmitFlight(202, 2, 0, 6, 4)
SubmitFlight(203, 1, 0, 4, 5)
PrintSchedule(1, 10)
SubmitFlight(205, 4, 2, 7, 2)
PrintSchedule(4, 6)
Reprioritize(203, 3, 9)
GroundHold(5, 4, 3)
GroundHold(4, 4, 3)
AddRunways(1, 3)
SubmitFlight(206, 6, 3, 6, 4)
CancelFlight(206, 3)
Tick(4)
SubmitFlight(204, 3, 4, 8, 2)
AddRunways(0, 5)
PrintActive()
PrintSchedule(5, 10)
Quit()
```

pdf_test1_output_file.txt

```
2 Runways are now available
Flight 201 scheduled — ETA: 4
Flight 202 scheduled — ETA: 4
Flight 203 scheduled — ETA: 9
[203]
Flight 205 scheduled — ETA: 6
[205]
Priority of Flight 203 has been updated to 9
Invalid input. Please provide a valid airline range.
Flights of the airlines in the range [4, 4] have been grounded
Additional 1 Runways are now available
Updated ETAs: [203: 8]
Flight 206 scheduled — ETA: 8
Flight 206 has been canceled
Flight 201 has landed at time 4
Flight 202 has landed at time 4
Flight 204 scheduled — ETA: 6
Invalid input. Please provide a valid number of runways.
[flight203, airline1, runway3, start3, ETA8]
[flight204, airline3, runway1, start4, ETA6]
There are no flights in that time period
Program Terminated!!
```

pdf_test2.txt

```
Initialize(2)
SubmitFlight(301, 5, 0, 7, 4)
SubmitFlight(302, 6, 0, 6, 5)
SubmitFlight(303, 7, 0, 5, 3)
PrintSchedule(5, 8)
GroundHold(6, 6, 0)
SubmitFlight(301, 5, 0, 7, 4)
SubmitFlight(304, 8, 1, 9, 4)
Reprioritize(304, 1, 10)
PrintSchedule(4, 8)
AddRunways(1, 1)
PrintSchedule(4, 8)
SubmitFlight(305, 9, 2, 6, 3)
CancelFlight(305, 2)
AddRunways(0, 2)
Tick(3)
PrintActive()
PrintSchedule(3, 7)
Tick(4)
Quit()
```

pdf_test2_output_file.txt

```
2 Runways are now available
Flight 301 scheduled — ETA: 4
Flight 302 scheduled — ETA: 5
Flight 303 scheduled — ETA: 7
[303]
Flights of the airlines in the range [6, 6] have been grounded
Duplicate FlightID
Flight 304 scheduled — ETA: 8
Updated ETAs: [303: 8]
Priority of Flight 304 has been updated to 10
[303]
[304]
Additional 1 Runways are now available
Updated ETAs: [303: 7, 304: 5]
[303]
Flight 305 scheduled — ETA: 7
Updated ETAs: [303: 8]
Flight 305 has been canceled
Updated ETAs: [303: 7]
Invalid input. Please provide a valid number of runways.
[flight301, airline5, runway1, start0, ETA4]
[flight302, airline6, runway2, start0, ETA5]
[flight303, airline7, runway1, start4, ETA7]
[flight304, airline8, runway3, start1, ETA5]
[303]
Flight 301 has landed at time 4
Program Terminated!!
```

test1.txt

```
Initialize(2)
SubmitFlight(401, 11, 0, 8, 4)
SubmitFlight(402, 12, 0, 7, 5)
SubmitFlight(403, 13, 0, 6, 3)
SubmitFlight(404, 14, 0, 5, 4)
PrintSchedule(3, 9)
Reprioritize(404, 1, 10)
AddRunways(1, 1)
SubmitFlight(405, 15, 2, 6, 2)
SubmitFlight(406, 16, 3, 7, 5)
GroundHold(16, 16, 3)
CancelFlight(405, 3)
Tick(4)
SubmitFlight(407, 17, 4, 6, 3)
Reprioritize(407, 4, 9)
SubmitFlight(408, 18, 4, 8, 2)
AddRunways(1, 5)
PrintActive()
Tick(8)
PrintSchedule(6, 12)
Quit()
```

test1_output_file.txt

```
2 Runways are now available
Flight 401 scheduled — ETA: 4
Flight 402 scheduled — ETA: 5
Flight 403 scheduled — ETA: 7
Flight 404 scheduled — ETA: 9
[403]
[404]
Priority of Flight 404 has been updated to 10
Updated ETAs: [403: 8, 404: 8]
Additional 1 Runways are now available
Updated ETAs: [403: 7, 404: 5]
Flight 405 scheduled — ETA: 7
Flight 406 scheduled — ETA: 9
Updated ETAs: [403: 8]
Flights of the airlines in the range [16, 16] have been grounded
Updated ETAs: [403: 7]
Flight 405 has been canceled
Flight 401 has landed at time 4
Flight 407 scheduled — ETA: 8
Priority of Flight 407 has been updated to 9
Flight 408 scheduled — ETA: 7
Flight 402 has landed at time 5
Flight 404 has landed at time 5
Additional 1 Runways are now available
[flight403, airline13, runway1, start4, ETA7]
[flight407, airline17, runway2, start5, ETA8]
[flight408, airline18, runway3, start5, ETA7]
Flight 403 has landed at time 7
Flight 408 has landed at time 7
Flight 407 has landed at time 8
There are no flights in that time period
Program Terminated!!
```

test2.txt

```
Initialize(3)
SubmitFlight(501, 20, 0, 8, 4)
SubmitFlight(502, 21, 0, 7, 6)
SubmitFlight(503, 22, 0, 7, 5)
SubmitFlight(510, 23, 0, 9, 3)
SubmitFlight(511, 23, 0, 9, 3)
SubmitFlight(504, 24, 0, 6, 4)
PrintSchedule(3, 10)
Reprioritize(504, 1, 10)
AddRunways(1, 1)
SubmitFlight(505, 25, 2, 6, 4)
SubmitFlight(506, 26, 3, 8, 5)
GroundHold(26, 26, 3)
CancelFlight(505, 3)
Tick(4)
SubmitFlight(507, 27, 4, 6, 4)
SubmitFlight(508, 28, 4, 7, 6)
Reprioritize(509, 4, 9)
SubmitFlight(509, 29, 4, 6, 6)
AddRunways(1, 6)
PrintActive()
Tick(9)
PrintSchedule(6, 14)
Quit()
```

test2_output_file.txt

```
3 Runways are now available
Flight 501 scheduled — ETA: 4
Flight 502 scheduled — ETA: 6
Flight 503 scheduled — ETA: 5
Flight 510 scheduled — ETA: 7
Flight 511 scheduled — ETA: 8
Flight 504 scheduled — ETA: 10
[510]
[511]
[504]
Priority of Flight 504 has been updated to 10
Updated ETAs: [504: 8, 510: 8, 511: 9]
Additional 1 Runways are now available
Updated ETAs: [504: 5, 510: 7, 511: 8]
Flight 505 scheduled — ETA: 9
Flight 506 scheduled — ETA: 10
Updated ETAs: [505: 10]
Flights of the airlines in the range [26, 26] have been grounded
Updated ETAs: [505: 9]
Flight 505 has been canceled
Flight 501 has landed at time 4
Flight 507 scheduled — ETA: 9
Flight 508 scheduled — ETA: 11
Updated ETAs: [507: 10]
Flight 509 not found
Flight 509 scheduled — ETA: 13
Flight 503 has landed at time 5
Flight 504 has landed at time 5
Flight 502 has landed at time 6
Additional 1 Runways are now available
Updated ETAs: [509: 12]
[flight507, airline27, runway2, start6, ETA10]
[flight508, airline28, runway4, start5, ETA11]
[flight509, airline29, runway5, start6, ETA12]
[flight510, airline23, runway1, start4, ETA7]
[flight511, airline23, runway3, start5, ETA8]
Flight 510 has landed at time 7
Flight 511 has landed at time 8
There are no flights in that time period
Program Terminated!!
```

test3.txt

```
Initialize(2)
SubmitFlight(601, 30, 0, 7, 5)
SubmitFlight(602, 31, 0, 6, 7)
SubmitFlight(603, 32, 0, 9, 3)
SubmitFlight(604, 33, 0, 5, 4)
PrintSchedule(2, 12)
Reprioritize(602, 1, 4)
Reprioritize(602, 1, 10)
AddRunways(0, 1)
AddRunways(1, 1)
SubmitFlight(605, 34, 2, 6, 6)
SubmitFlight(606, 35, 3, 8, 2)
SubmitFlight(607, 36, 3, 5, 9)
GroundHold(33, 36, 3)
CancelFlight(605, 3)
Tick(4)
SubmitFlight(610, 37, 6, 6, 12)
SubmitFlight(611, 38, 7, 8, 1)
SubmitFlight(612, 38, 7, 8, 1)
SubmitFlight(613, 39, 7, 7, 1)
PrintSchedule(7, 10)
Tick(8)
SubmitFlight(608, 40, 8, 7, 5)
SubmitFlight(609, 41, 8, 7, 5)
Reprioritize(608, 8, 11)
AddRunways(1, 8)
PrintActive()
Tick(18)
PrintSchedule(10, 25)
Quit()
```

test3_output_file.txt

```
2 Runways are now available
Flight 601 scheduled — ETA: 5
Flight 602 scheduled — ETA: 7
Flight 603 scheduled — ETA: 8
Flight 604 scheduled — ETA: 11
[603]
[604]
Cannot reprioritize. Flight 602 has already departed
Cannot reprioritize. Flight 602 has already departed
Invalid input. Please provide a valid number of runways.
Additional 1 Runways are now available
Updated ETAs: [603: 4, 604: 8]
Flight 605 scheduled — ETA: 10
Updated ETAs: [604: 9]
Flight 606 scheduled — ETA: 6
Updated ETAs: [604: 10, 605: 11]
Flight 607 scheduled — ETA: 16
Flights of the airlines in the range [33, 36] have been grounded
Flight 605 does not exist
Flight 603 has landed at time 4
Flight 601 has landed at time 5
Flight 610 scheduled — ETA: 18
Flight 602 has landed at time 7
Flight 611 scheduled — ETA: 8
Flight 612 scheduled — ETA: 8
Flight 613 scheduled — ETA: 9
[613]
Flight 611 has landed at time 8
Flight 612 has landed at time 8
Flight 608 scheduled — ETA: 13
Flight 609 scheduled — ETA: 14
Cannot reprioritize. Flight 608 has already departed
Additional 1 Runways are now available
Updated ETAs: [609: 13]
[flight608, airline40, runway3, start8, ETA13]
[flight609, airline41, runway4, start8, ETA13]
[flight610, airline37, runway1, start6, ETA18]
[flight613, airline39, runway2, start8, ETA9]
Flight 613 has landed at time 9
Flight 608 has landed at time 13
Flight 609 has landed at time 13
Flight 610 has landed at time 18
There are no flights in that time period
Program Terminated!!
```