Project: MP1
Course: ITM-513
Author: Brian T. Bailey

Project Description:

Application 1:
The object of application 1 is to create a Python application that models a priority queue. The queue should be a list element. Each task in the queue should be modeled as a dictionary with one key-value pair. The key should be the task name and the value should be the priority value, between 1-5. There needs to be a get() method to remove the the tasks from the list in highest priority first in first out order. There also needs to be a put() method to insert a task into the queue and make sure it is in the proper order. Hard-coded test data needs to be included to show the application working.

Application 2:
The object of application 2 is to create a Python application that processes and does some computations on three dimensional coordinate data. The application needs to randomly generate 100 three dimensional coordinate tuples. It also needs to be able to sort the tuples three different ways, by the x, y and z dimensions. These three sorted lists need to be stored in a dictionary. The application also needs to calculate and display the coordinates that are closest and farthest from the origin.

Installation, Compile and Run-Time Requirements:

This project was written in Python using version 2.7.1. The scripts were written in BBEdit version 10.1.2 on the Macintosh platform. The computer used was a 2.7 GHz dual-core Intel Core i7 13" MacBook Pro with 8GB of RAM running OS X Lion 10.7.4.

Insights and Expected Results:

Application 1:
There was discussion on a few different ways to structure this application. I chose to use an architecture similar to what was in the original project directions and described by Professor Kimont in the discussion board.

In this design, each task was represented by a dictionary element with a single key-value pair. The key was the task name as a string and the value was the numeric priority value from 1-5. A list element was used to represent the priority queue.

The put() function takes the queue list and task dictionary element as parameters. It then appends the task to the list, sorts the list based on priority value and prints a message to the display saying the task was added. The list is sorted using the sort() function with a lambda function as the key. The key function extracts the priority value from the item in the list and does a reverse sort on that value. This gives us an order with the 5 priorities at the top and 1 priorities at the bottom. This works because the sort function is guaranteed to be stable in Python 2.2 and greater. This means if multiple items have the same key value their original order is preserved. This keeps the tasks in the order they were inserted within their respective priority value and allows the list to be a FIFO highest priority queue.

The get() function is a simple function because of the chosen architecture. All that needs to be done is to retrieve the first item in the queue list. My get() function takes the list as a parameter, removes the first item from the list, prints a message to the display showing the item removed and returns the item from the function.

The test data for this application is stored in a text file called test_data.txt. This file must be located in the same directory as the application in order to run correctly.

Application 2:
I found application 2 to be the easier of the two applications. Where in application 1 I had to think about how to structure the queue and add elements, I pretty much knew what I needed to do after a first read of the directions for application 2.

When generating the random 100 coordinates I decided to limit the range of the coordinates from -250 to 250. This kept the coordinate space in a 500 point cube centered on the origin. I used a list comprehension to create a list of random values for x, y and z. I then used the zip() function on those three lists to form the list of tuples that represented the coordinates.

The project directions said to call the list of coordinate tuples 3Space. That was not possible and I had to call it threeSpace. This is because Python can not have variables that start with a number.

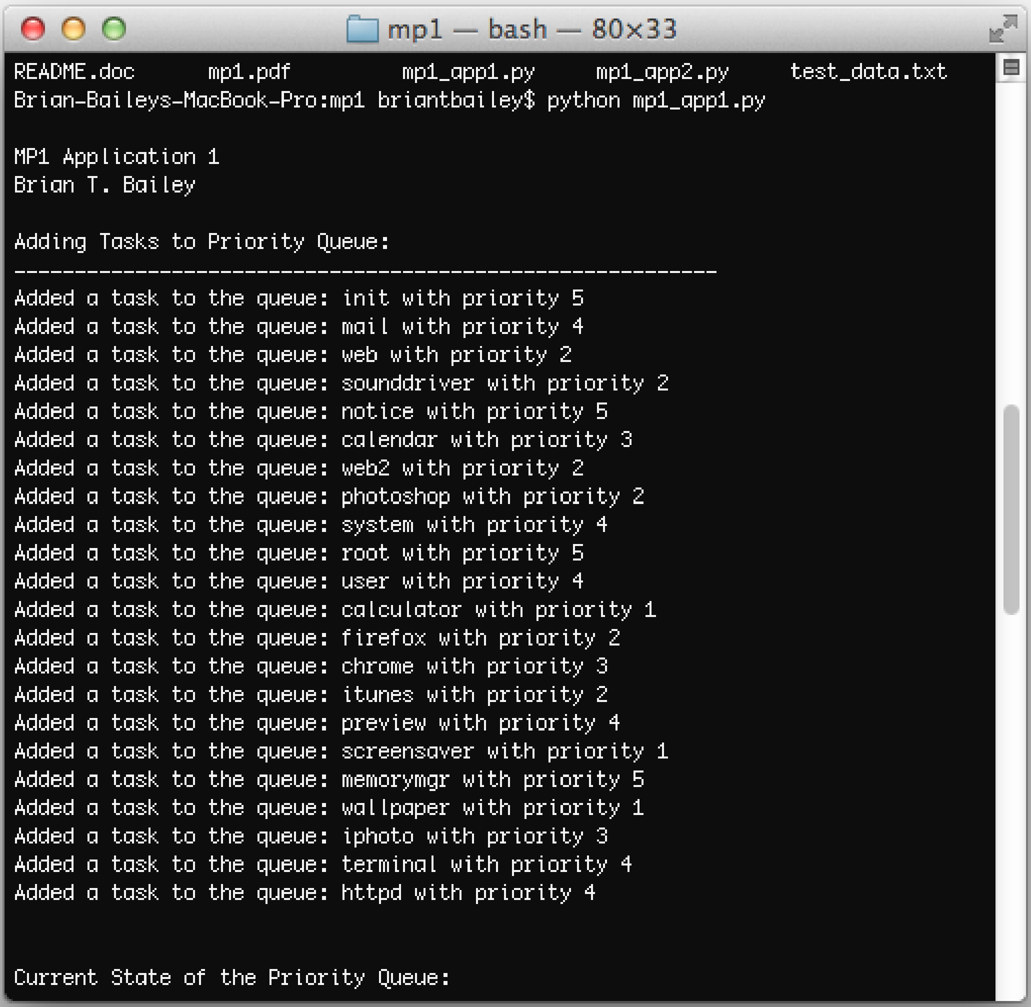
When I sorted the list by x, y and z I used the sorted() function so it would return a new copy of the sorted list and didn't change the original list. In the sorted() function I used a key to determine the sorting. The key was a lambda function that returned the value of the respective x, y or z from the coordinate tuple. Those sorted lists were then stored in a dictionary.

The distance from the origin was calculated using the Euclidian Distance which is based off the pythagorean theorem. I defined a function to take a coordinate tuple as the parameter and calculate and return the distance to the origin. I then sorted the original threeSpace list using this function as the key to the sort() function. The closest point was then at position 0 in the list and the farthest was at the last position in the list, length – 1.

<u>Screenshots Demonstrating Application:</u>

Screenshots showing the application functionality are on the following pages.

Application 1:
Application 1 starting in the terminal window. This screenshot shows the tasks being added from the data file.



mp1 — bash — 80×33

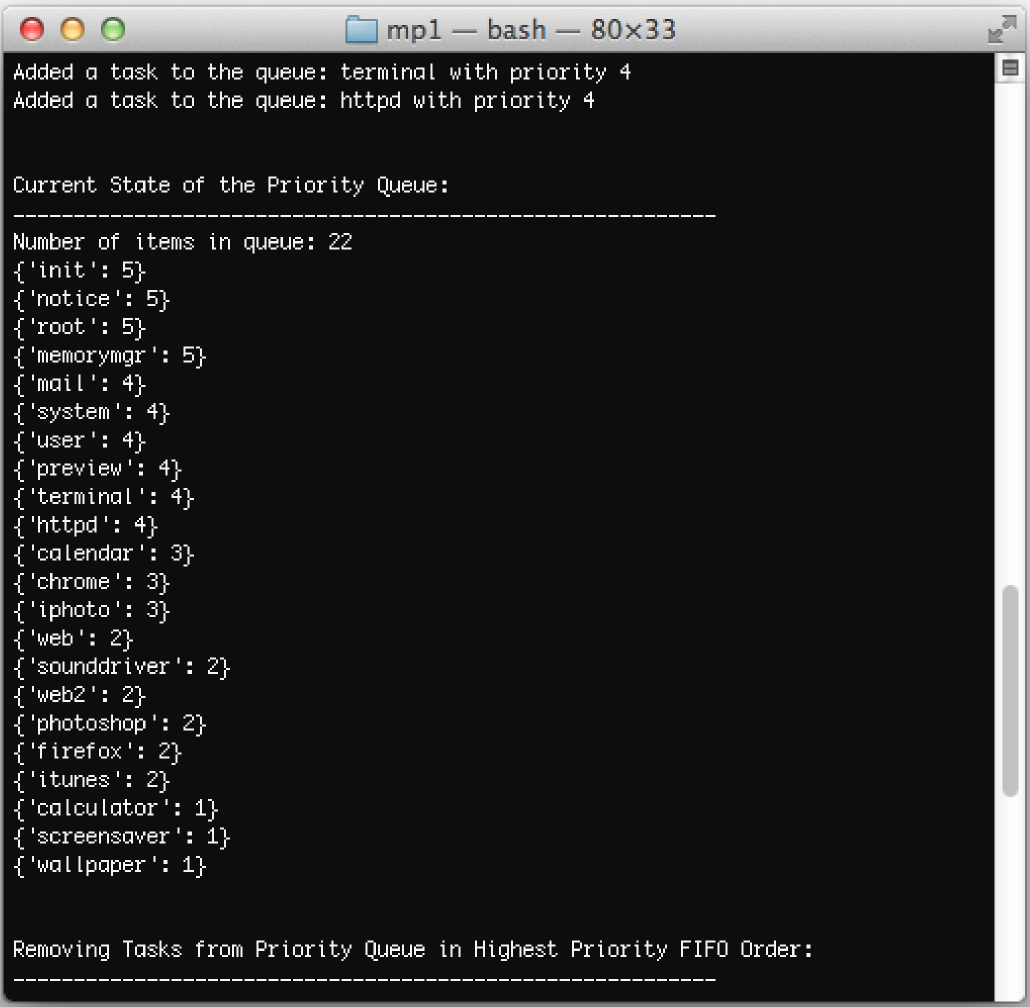
```
README.doc      mp1.pdf         mp1_app1.py     mp1_app2.py     test_data.txt
Brian-Baileys-MacBook-Pro:mp1 briantbailey$ python mp1_app1.py

MP1 Application 1
Brian T. Bailey

Adding Tasks to Priority Queue:
----------------------------------------------------------
Added a task to the queue: init with priority 5
Added a task to the queue: mail with priority 4
Added a task to the queue: web with priority 2
Added a task to the queue: sounddriver with priority 2
Added a task to the queue: notice with priority 5
Added a task to the queue: calendar with priority 3
Added a task to the queue: web2 with priority 2
Added a task to the queue: photoshop with priority 2
Added a task to the queue: system with priority 4
Added a task to the queue: root with priority 5
Added a task to the queue: user with priority 4
Added a task to the queue: calculator with priority 1
Added a task to the queue: firefox with priority 2
Added a task to the queue: chrome with priority 3
Added a task to the queue: itunes with priority 2
Added a task to the queue: preview with priority 4
Added a task to the queue: screensaver with priority 1
Added a task to the queue: memorymgr with priority 5
Added a task to the queue: wallpaper with priority 1
Added a task to the queue: iphoto with priority 3
Added a task to the queue: terminal with priority 4
Added a task to the queue: httpd with priority 4


Current State of the Priority Queue:
```

Application 1 showing the current ordered state of the priority queue and the number of items in the queue. These task items are dictionary elements.

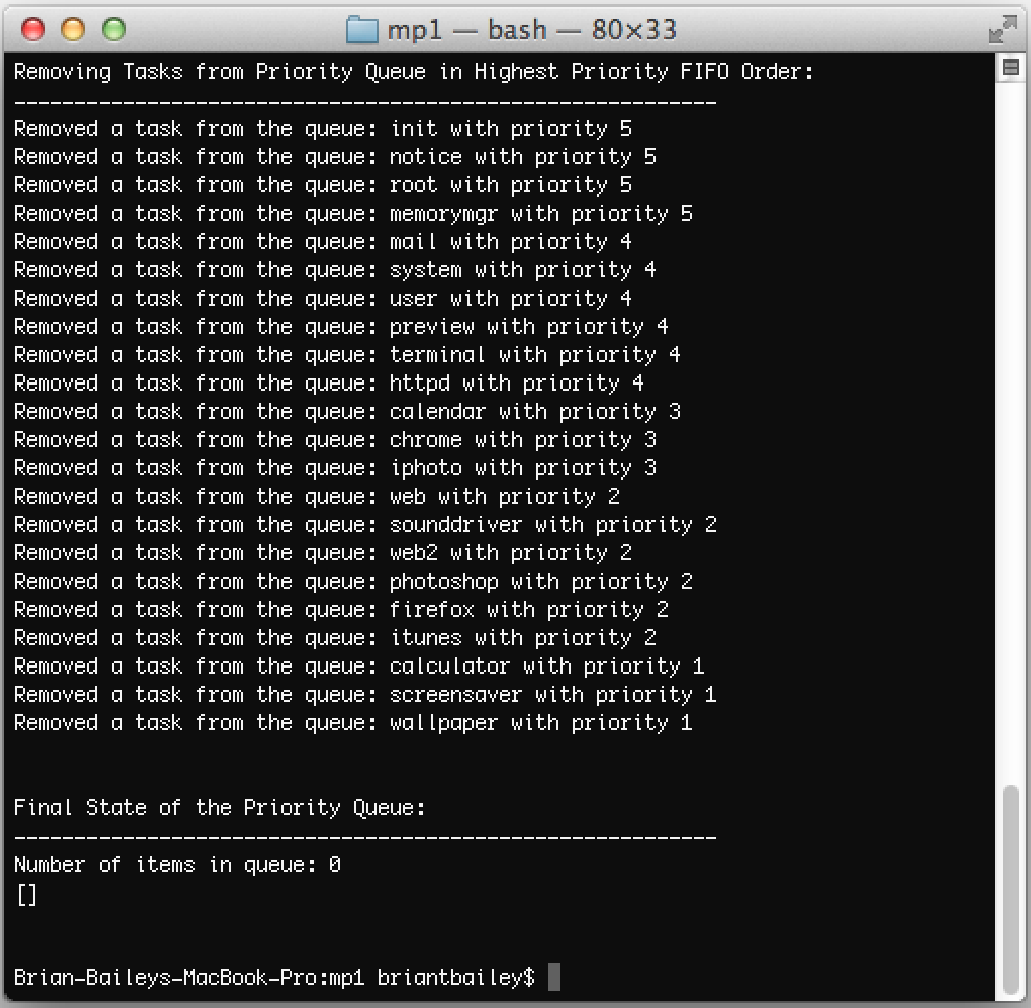


```
Added a task to the queue: terminal with priority 4
Added a task to the queue: httpd with priority 4


Current State of the Priority Queue:
----------------------------------------------------------
Number of items in queue: 22
{'init': 5}
{'notice': 5}
{'root': 5}
{'memorymgr': 5}
{'mail': 4}
{'system': 4}
{'user': 4}
{'preview': 4}
{'terminal': 4}
{'httpd': 4}
{'calendar': 3}
{'chrome': 3}
{'iphoto': 3}
{'web': 2}
{'sounddriver': 2}
{'web2': 2}
{'photoshop': 2}
{'firefox': 2}
{'itunes': 2}
{'calculator': 1}
{'screensaver': 1}
{'wallpaper': 1}


Removing Tasks from Priority Queue in Highest Priority FIFO Order:
----------------------------------------------------------
```

Application 1 finishing running. This screenshot shows the tasks being removed from the queue in FIFO highest priority order. It also shows the final state of the queue with 0 elements.

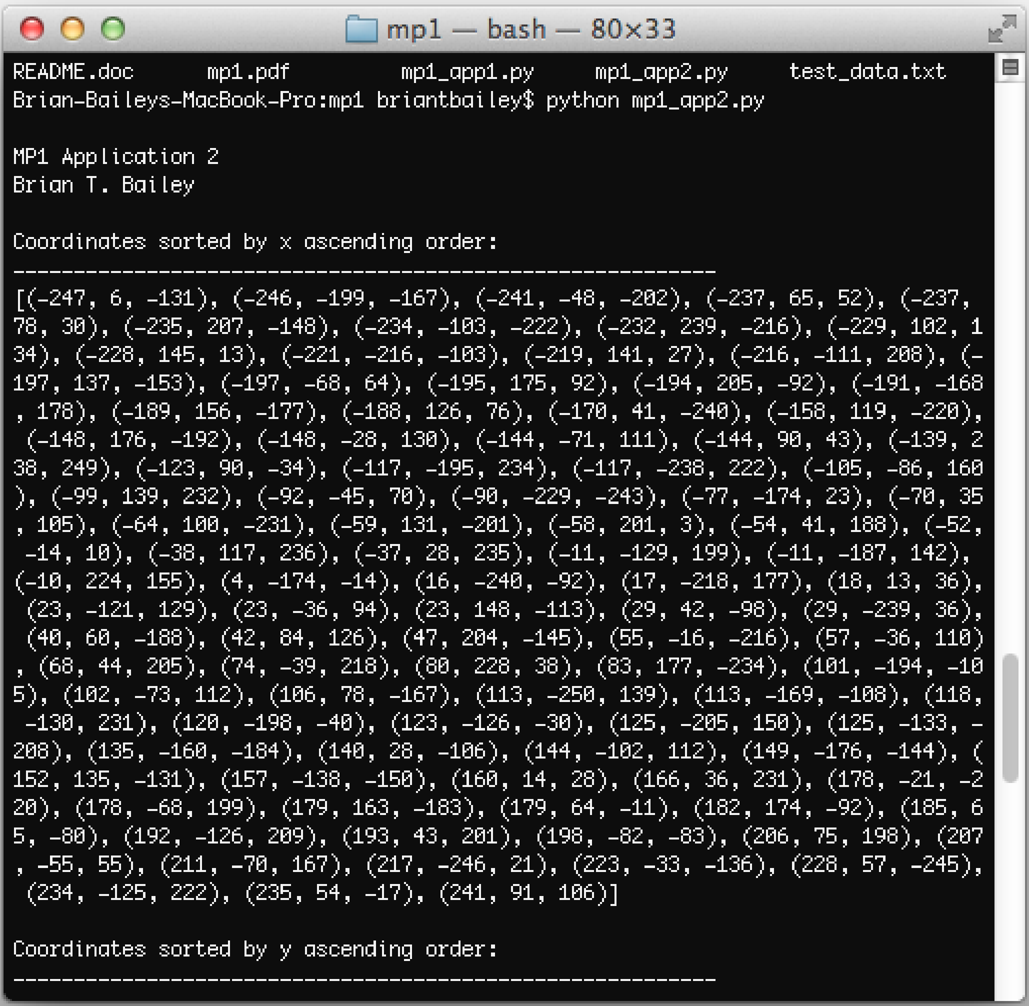


```
Removing Tasks from Priority Queue in Highest Priority FIFO Order:
-----------------------------------------------------------
Removed a task from the queue: init with priority 5
Removed a task from the queue: notice with priority 5
Removed a task from the queue: root with priority 5
Removed a task from the queue: memorymgr with priority 5
Removed a task from the queue: mail with priority 4
Removed a task from the queue: system with priority 4
Removed a task from the queue: user with priority 4
Removed a task from the queue: preview with priority 4
Removed a task from the queue: terminal with priority 4
Removed a task from the queue: httpd with priority 4
Removed a task from the queue: calendar with priority 3
Removed a task from the queue: chrome with priority 3
Removed a task from the queue: iphoto with priority 3
Removed a task from the queue: web with priority 2
Removed a task from the queue: sounddriver with priority 2
Removed a task from the queue: web2 with priority 2
Removed a task from the queue: photoshop with priority 2
Removed a task from the queue: firefox with priority 2
Removed a task from the queue: itunes with priority 2
Removed a task from the queue: calculator with priority 1
Removed a task from the queue: screensaver with priority 1
Removed a task from the queue: wallpaper with priority 1


Final State of the Priority Queue:
-----------------------------------------------------------
Number of items in queue: 0
[]


Brian-Baileys-MacBook-Pro:mp1 briantbailey$
```

Application 2:
Application 2 starting in the terminal window. This screenshot shows the application starting and the coordinate tuples sorted by the x coordinate in ascending order.



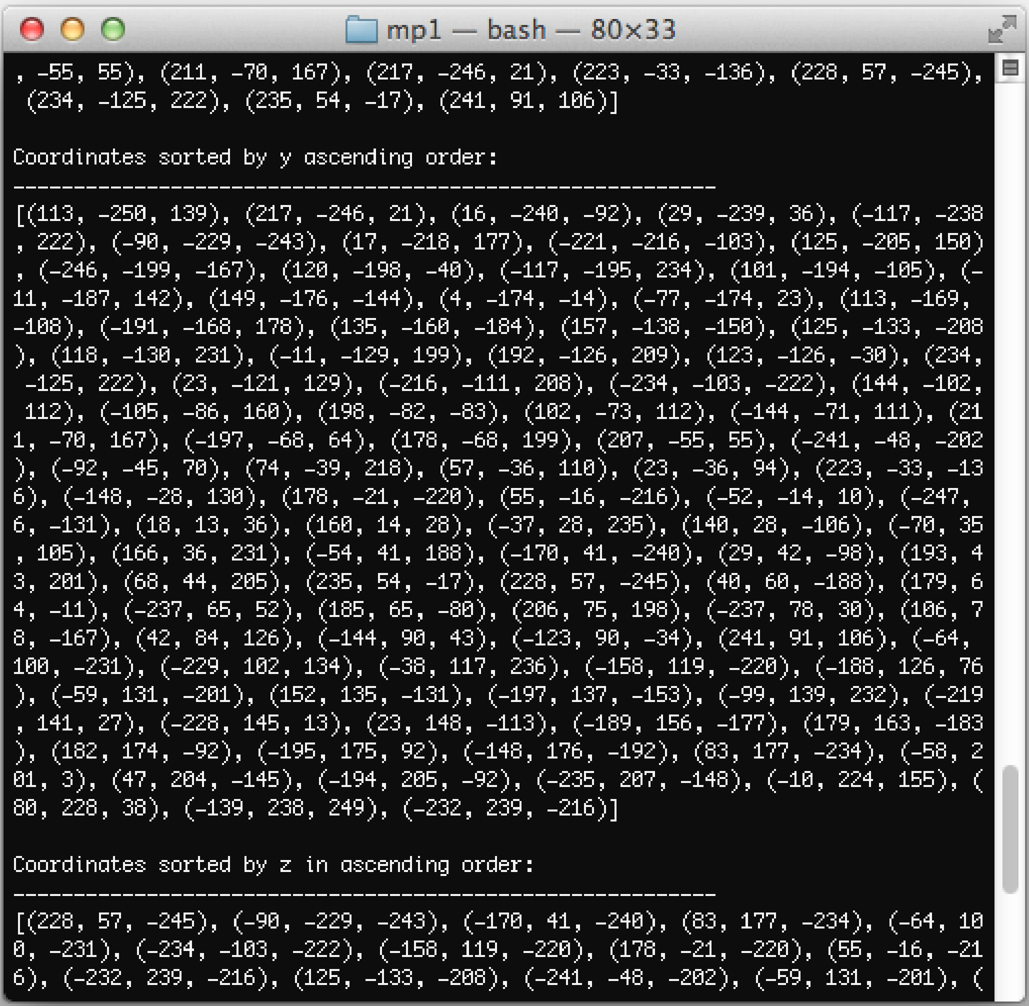
```
README.doc      mp1.pdf         mp1_app1.py     mp1_app2.py     test_data.txt
Brian-Baileys-MacBook-Pro:mp1 briantbailey$ python mp1_app2.py

MP1 Application 2
Brian T. Bailey

Coordinates sorted by x ascending order:
-----------------------------------------------------------
[(-247, 6, -131), (-246, -199, -167), (-241, -48, -202), (-237, 65, 52), (-237,
78, 30), (-235, 207, -148), (-234, -103, -222), (-232, 239, -216), (-229, 102, 1
34), (-228, 145, 13), (-221, -216, -103), (-219, 141, 27), (-216, -111, 208), (-
197, 137, -153), (-197, -68, 64), (-195, 175, 92), (-194, 205, -92), (-191, -168
, 178), (-189, 156, -177), (-188, 126, 76), (-170, 41, -240), (-158, 119, -220),
 (-148, 176, -192), (-148, -28, 130), (-144, -71, 111), (-144, 90, 43), (-139, 2
38, 249), (-123, 90, -34), (-117, -195, 234), (-117, -238, 222), (-105, -86, 160
), (-99, 139, 232), (-92, -45, 70), (-90, -229, -243), (-77, -174, 23), (-70, 35
, 105), (-64, 100, -231), (-59, 131, -201), (-58, 201, 3), (-54, 41, 188), (-52,
 -14, 10), (-38, 117, 236), (-37, 28, 235), (-11, -129, 199), (-11, -187, 142),
(-10, 224, 155), (4, -174, -14), (16, -240, -92), (17, -218, 177), (18, 13, 36),
 (23, -121, 129), (23, -36, 94), (23, 148, -113), (29, 42, -98), (29, -239, 36),
 (40, 60, -188), (42, 84, 126), (47, 204, -145), (55, -16, -216), (57, -36, 110)
, (68, 44, 205), (74, -39, 218), (80, 228, 38), (83, 177, -234), (101, -194, -10
5), (102, -73, 112), (106, 78, -167), (113, -250, 139), (113, -169, -108), (118,
 -130, 231), (120, -198, -40), (123, -126, -30), (125, -205, 150), (125, -133, -
208), (135, -160, -184), (140, 28, -106), (144, -102, 112), (149, -176, -144), (
152, 135, -131), (157, -138, -150), (160, 14, 28), (166, 36, 231), (178, -21, -2
20), (178, -68, 199), (179, 163, -183), (179, 64, -11), (182, 174, -92), (185, 6
5, -80), (192, -126, 209), (193, 43, 201), (198, -82, -83), (206, 75, 198), (207
, -55, 55), (211, -70, 167), (217, -246, 21), (223, -33, -136), (228, 57, -245),
 (234, -125, 222), (235, 54, -17), (241, 91, 106)]

Coordinates sorted by y ascending order:
-----------------------------------------------------------
```

This screenshot shows application 2 displaying the coordinate tuples sorted by the y coordinate in ascending order.

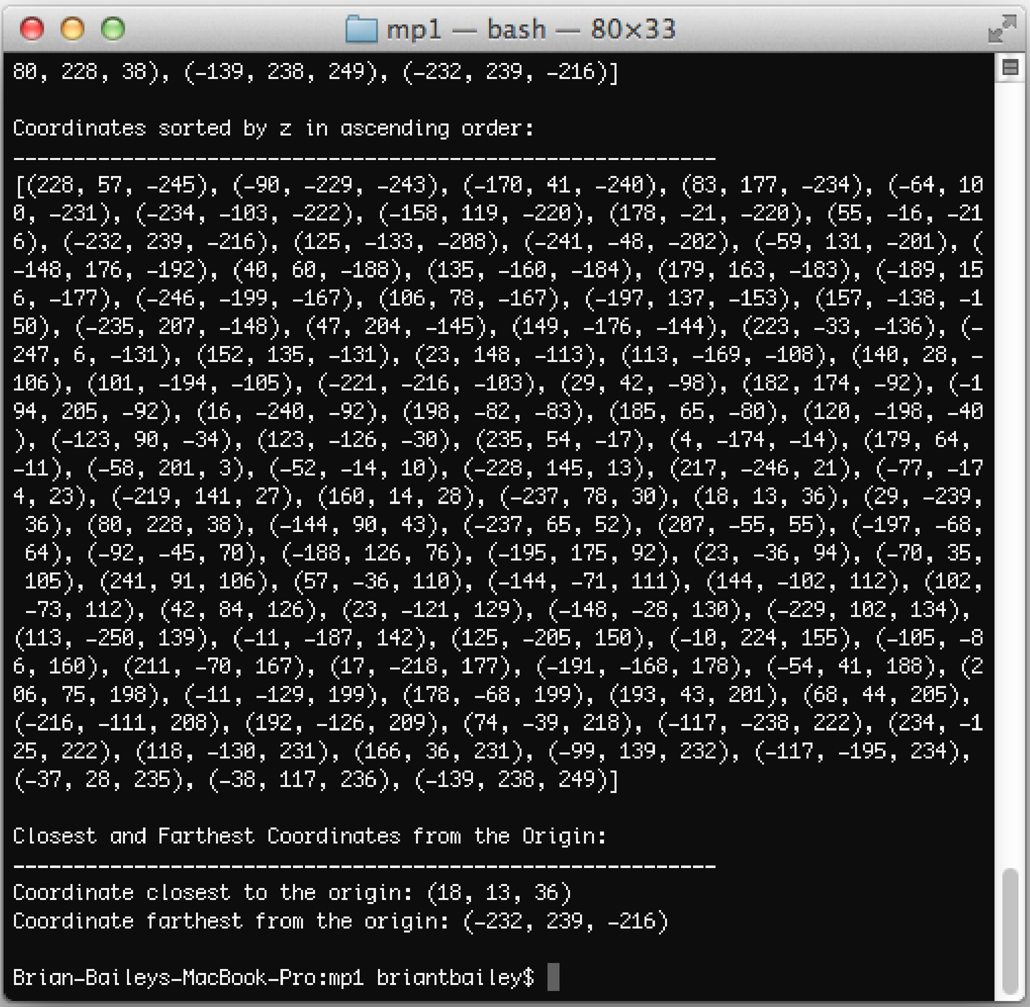


```
, -55, 55), (211, -70, 167), (217, -246, 21), (223, -33, -136), (228, 57, -245),
 (234, -125, 222), (235, 54, -17), (241, 91, 106)]

Coordinates sorted by y ascending order:
----------------------------------------------------------
[(113, -250, 139), (217, -246, 21), (16, -240, -92), (29, -239, 36), (-117, -238
, 222), (-90, -229, -243), (17, -218, 177), (-221, -216, -103), (125, -205, 150)
, (-246, -199, -167), (120, -198, -40), (-117, -195, 234), (101, -194, -105), (-
11, -187, 142), (149, -176, -144), (4, -174, -14), (-77, -174, 23), (113, -169,
-108), (-191, -168, 178), (135, -160, -184), (157, -138, -150), (125, -133, -208
), (118, -130, 231), (-11, -129, 199), (192, -126, 209), (123, -126, -30), (234,
 -125, 222), (23, -121, 129), (-216, -111, 208), (-234, -103, -222), (144, -102,
 112), (-105, -86, 160), (198, -82, -83), (102, -73, 112), (-144, -71, 111), (21
1, -70, 167), (-197, -68, 64), (178, -68, 199), (207, -55, 55), (-241, -48, -202
), (-92, -45, 70), (74, -39, 218), (57, -36, 110), (23, -36, 94), (223, -33, -13
6), (-148, -28, 130), (178, -21, -220), (55, -16, -216), (-52, -14, 10), (-247,
6, -131), (18, 13, 36), (160, 14, 28), (-37, 28, 235), (140, 28, -106), (-70, 35
, 105), (166, 36, 231), (-54, 41, 188), (-170, 41, -240), (29, 42, -98), (193, 4
3, 201), (68, 44, 205), (235, 54, -17), (228, 57, -245), (40, 60, -188), (179, 6
4, -11), (-237, 65, 52), (185, 65, -80), (206, 75, 198), (-237, 78, 30), (106, 7
8, -167), (42, 84, 126), (-144, 90, 43), (-123, 90, -34), (241, 91, 106), (-64,
100, -231), (-229, 102, 134), (-38, 117, 236), (-158, 119, -220), (-188, 126, 76
), (-59, 131, -201), (152, 135, -131), (-197, 137, -153), (-99, 139, 232), (-219
, 141, 27), (-228, 145, 13), (23, 148, -113), (-189, 156, -177), (179, 163, -183
), (182, 174, -92), (-195, 175, 92), (-148, 176, -192), (83, 177, -234), (-58, 2
01, 3), (47, 204, -145), (-194, 205, -92), (-235, 207, -148), (-10, 224, 155), (
80, 228, 38), (-139, 238, 249), (-232, 239, -216)]

Coordinates sorted by z in ascending order:
----------------------------------------------------------
[(228, 57, -245), (-90, -229, -243), (-170, 41, -240), (83, 177, -234), (-64, 10
0, -231), (-234, -103, -222), (-158, 119, -220), (178, -21, -220), (55, -16, -21
6), (-232, 239, -216), (125, -133, -208), (-241, -48, -202), (-59, 131, -201), (
```

This screenshot shows application 2 displaying the coordinate tuples sorted by the z coordinate in ascending order. It also shows the results of the closest and farthest from the origin calculation and the application finishing.



```
80, 228, 38), (-139, 238, 249), (-232, 239, -216)]

Coordinates sorted by z in ascending order:
----------------------------------------------------------------
[(228, 57, -245), (-90, -229, -243), (-170, 41, -240), (83, 177, -234), (-64, 10
0, -231), (-234, -103, -222), (-158, 119, -220), (178, -21, -220), (55, -16, -21
6), (-232, 239, -216), (125, -133, -208), (-241, -48, -202), (-59, 131, -201), (
-148, 176, -192), (40, 60, -188), (135, -160, -184), (179, 163, -183), (-189, 15
6, -177), (-246, -199, -167), (106, 78, -167), (-197, 137, -153), (157, -138, -1
50), (-235, 207, -148), (47, 204, -145), (149, -176, -144), (223, -33, -136), (-
247, 6, -131), (152, 135, -131), (23, 148, -113), (113, -169, -108), (140, 28, -
106), (101, -194, -105), (-221, -216, -103), (29, 42, -98), (182, 174, -92), (-1
94, 205, -92), (16, -240, -92), (198, -82, -83), (185, 65, -80), (120, -198, -40
), (-123, 90, -34), (123, -126, -30), (235, 54, -17), (4, -174, -14), (179, 64,
-11), (-58, 201, 3), (-52, -14, 10), (-228, 145, 13), (217, -246, 21), (-77, -17
4, 23), (-219, 141, 27), (160, 14, 28), (-237, 78, 30), (18, 13, 36), (29, -239,
 36), (80, 228, 38), (-144, 90, 43), (-237, 65, 52), (207, -55, 55), (-197, -68,
 64), (-92, -45, 70), (-188, 126, 76), (-195, 175, 92), (23, -36, 94), (-70, 35,
 105), (241, 91, 106), (57, -36, 110), (-144, -71, 111), (144, -102, 112), (102,
 -73, 112), (42, 84, 126), (23, -121, 129), (-148, -28, 130), (-229, 102, 134),
(113, -250, 139), (-11, -187, 142), (125, -205, 150), (-10, 224, 155), (-105, -8
6, 160), (211, -70, 167), (17, -218, 177), (-191, -168, 178), (-54, 41, 188), (2
06, 75, 198), (-11, -129, 199), (178, -68, 199), (193, 43, 201), (68, 44, 205),
(-216, -111, 208), (192, -126, 209), (74, -39, 218), (-117, -238, 222), (234, -1
25, 222), (118, -130, 231), (166, 36, 231), (-99, 139, 232), (-117, -195, 234),
(-37, 28, 235), (-38, 117, 236), (-139, 238, 249)]

Closest and Farthest Coordinates from the Origin:
----------------------------------------------------------------
Coordinate closest to the origin: (18, 13, 36)
Coordinate farthest from the origin: (-232, 239, -216)

Brian-Baileys-MacBook-Pro:mp1 briantbailey$
```