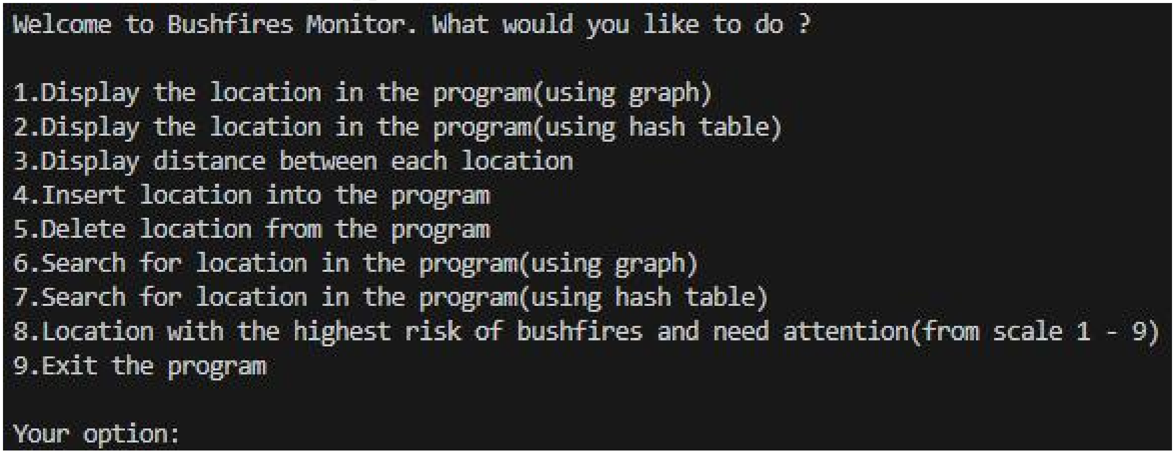
MUST READ

Files that should be present in the folder and their description are as below:

File Name	Description
coversheet	Signed and dated cover sheet
DSA_Assignment_sem1_2023_	Question and requirement of Assignment.
DoubleLinkList	Java class depend DoubleListNode class.
DoubleListNode	Java class depend on DSAGraphNode class.
DSAGraph	Java class implement graph structure, depends on DoubleListNode and DSAGraphNode class.
DSAGraphNode	Java class, consist of setter and getter for DSAGraphNode.
DSAHeap	Java class implement the heap structure and depends on DSAHeapEntry class.
DSAHeapEntry	Java class, consist of setter and getter for DSAHeap.
DSAHashTable	Java class implement the hash table structure and depends on DSAHashEntry class.
DSAHashEntry	Java class, consist of setter and getter for DSAHashTable.
Menu	Java class, have the main method which depends on DSAGraph, DSAHeap and DSAHashTable java class.
UnitTestDSAGraph	Java class for testing DSAGraph class.
UnitTestDSAHashTable	Java class for testing DSAHashTable class.
UnitTestDSAHeap	Java class for testing DSAHeap class.
AssignmentReport	Report for the assignment, consist of the program design, UML diagram, data structure and algorithm used in the program, description of classes, testing methodology and the result.
location	Txt file which has the location which is connected and the distances between the location
UAVdata	Txt file which has the location and the data of the location collected by the UAVs.

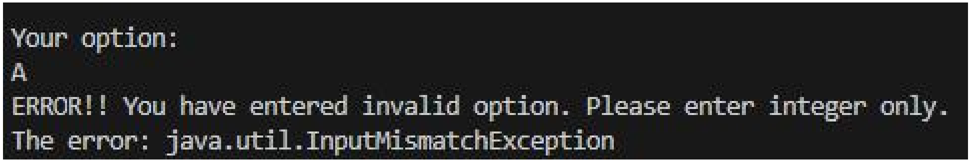
Step to run the program:

1. Before running the program, make sure the menu class which have the main method has been compiled.
2. Compile the class by using command “javac javaclass.java”, to compile Menu.java, type command “javac Menu.java”. in the terminal.
3. After compiling, run the code by writing the command “java Menu”. The menu selection as below should appear:

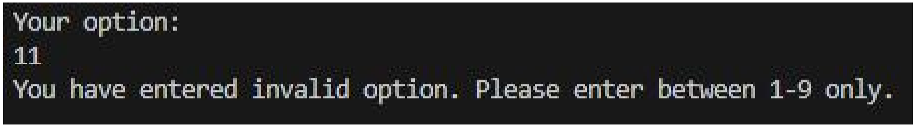


```
Welcome to Bushfires Monitor. What would you like to do ?

1.Display the location in the program(using graph)
2.Display the location in the program(using hash table)
3.Display distance between each location
4.Insert location into the program
5.Delete location from the program
6.Search for location in the program(using graph)
7.Search for location in the program(using hash table)
8.Location with the highest risk of bushfires and need attention(from scale 1 - 9)
9.Exit the program

Your option:
```

4. The user should then type in their option in integer. If the user inserts other data type, such as char, error message will be displayed. While if the integer insert is not in the list, default message will be displayed.

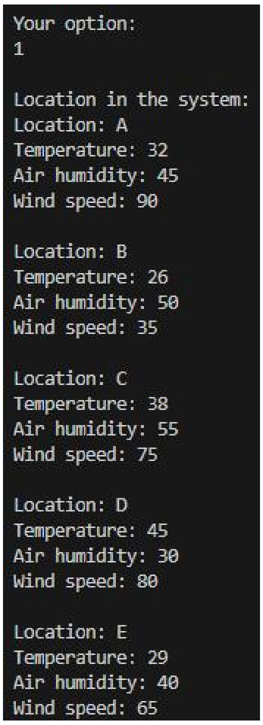


```
Your option:
A
ERROR!! You have entered invalid option. Please enter integer only.
The error: java.util.InputMismatchException
```

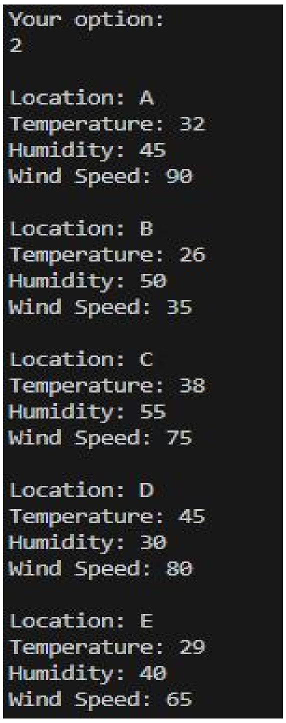


```
Your option:
11
You have entered invalid option. Please enter between 1-9 only.
```

5. For option 1 and 2, both will display the location and the data of the location in the UAVdata.txt. The difference is option 1 is implemented by using graph and option 2 is implemented by using hash table. (Disclaimer: Data shown below is not the complete data.)

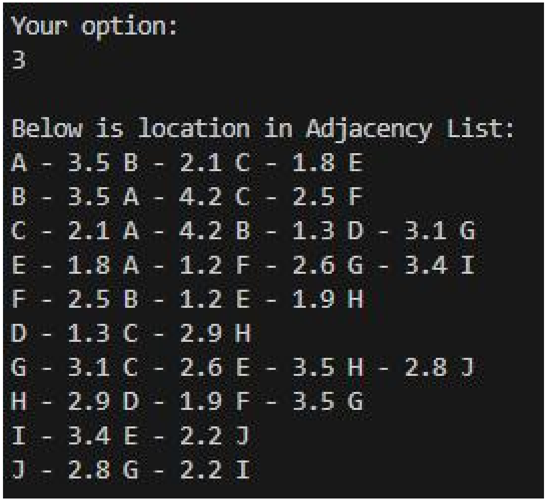


```
Your option:
1

Location in the system:
Location: A
Temperature: 32
Air humidity: 45
Wind speed: 90

Location: B
Temperature: 26
Air humidity: 50
Wind speed: 35

Location: C
Temperature: 38
Air humidity: 55
Wind speed: 75

Location: D
Temperature: 45
Air humidity: 30
Wind speed: 80

Location: E
Temperature: 29
Air humidity: 40
Wind speed: 65
```



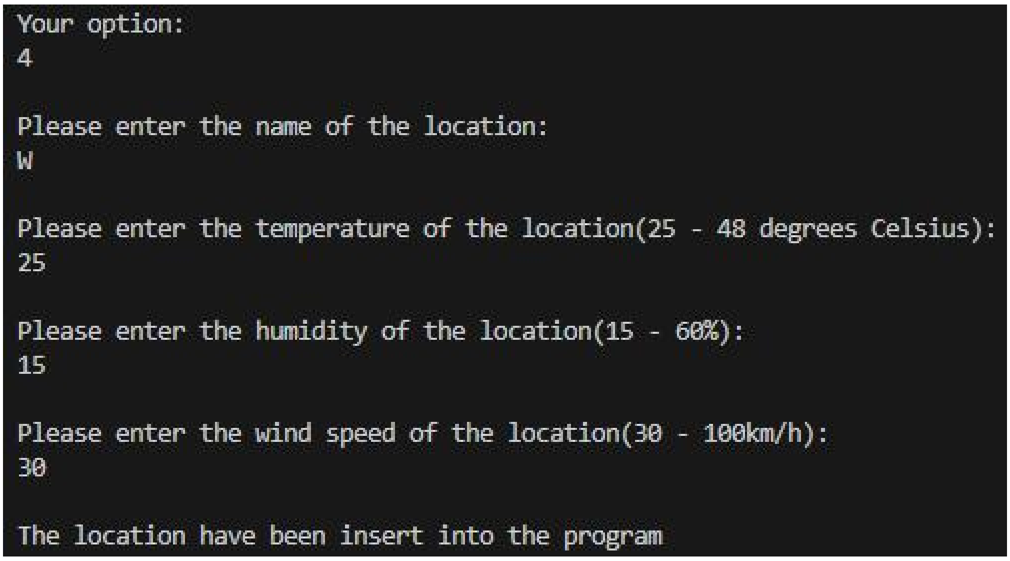
```
Your option:
2

Location: A
Temperature: 32
Humidity: 45
Wind Speed: 90

Location: B
Temperature: 26
Humidity: 50
Wind Speed: 35

Location: C
Temperature: 38
Humidity: 55
Wind Speed: 75

Location: D
Temperature: 45
Humidity: 30
Wind Speed: 80

Location: E
Temperature: 29
Humidity: 40
Wind Speed: 65
```

6. For option 3, the location adjacent to each other and their distance will be displayed.



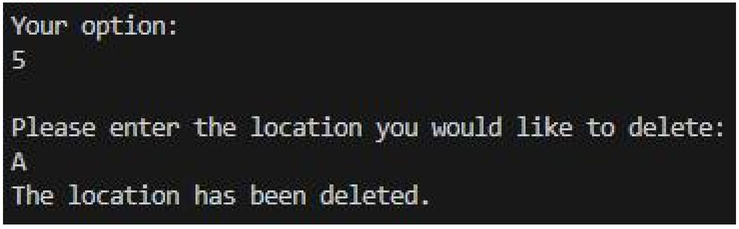
```
Your option:
3

Below is location in Adjacency List:
A - 3.5 B - 2.1 C - 1.8 E
B - 3.5 A - 4.2 C - 2.5 F
C - 2.1 A - 4.2 B - 1.3 D - 3.1 G
E - 1.8 A - 1.2 F - 2.6 G - 3.4 I
F - 2.5 B - 1.2 E - 1.9 H
D - 1.3 C - 2.9 H
G - 3.1 C - 2.6 E - 3.5 H - 2.8 J
H - 2.9 D - 1.9 F - 3.5 G
I - 3.4 E - 2.2 J
J - 2.8 G - 2.2 I
```

7. Option 4 will ask the user to insert new location, the temperature, humidity, and wind speed of that location into the graph. If the location inserts already exist, error message will be displayed. Error message will also be displayed, if the temperature, humidity and wind speed enter is out of the boundaries, and the program will keep on looping or repeating until the valid value is inserted.



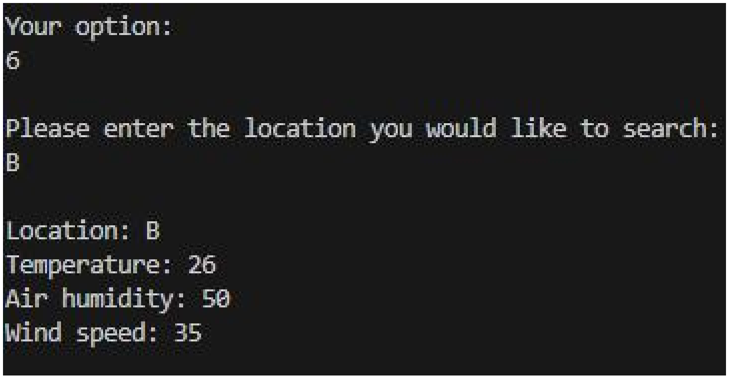
```
Your option:
4

Please enter the name of the location:
W

Please enter the temperature of the location(25 - 48 degrees Celsius):
25

Please enter the humidity of the location(15 - 60%):
15

Please enter the wind speed of the location(30 - 100km/h):
30

The location have been insert into the program
```

8. Option 5 will delete the location in the graph. If the location inserts do not exist, error message will be displayed.



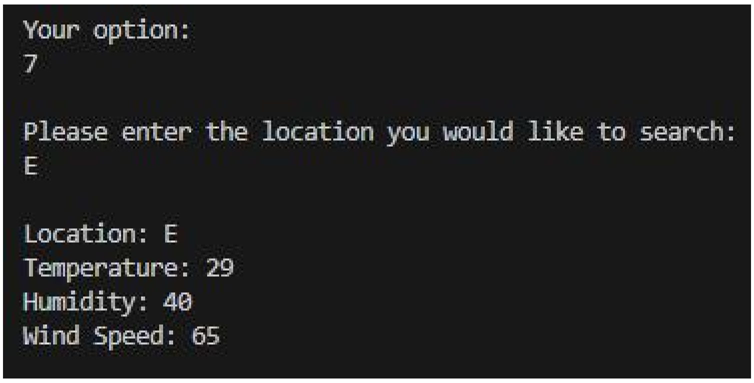
```
Your option:
5

Please enter the location you would like to delete:
A
The location has been deleted.
```

9. Option 6 will show the certain location and the data location chosen or inserted by the user in the graph. Below is the example of the outcome:



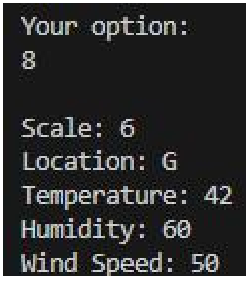
```
Your option:
6

Please enter the location you would like to search:
B

Location: B
Temperature: 26
Air humidity: 50
Wind speed: 35
```

10. Option 7 will show the certain location and the data location chosen or inserted by the user in the hash table. Below is the example of the outcome:



```
Your option:
7

Please enter the location you would like to search:
E

Location: E
Temperature: 29
Humidity: 40
Wind Speed: 65
```

11. Option 8 will display the location with the high risk of bushfires and need attention.



```
Your option:
8

Scale: 6
Location: G
Temperature: 42
Humidity: 60
Wind Speed: 50
```

12. Option 9 displayed good bye message and exit the program.