**Summary Report:**

**(Interpreted) Requirements of the program**

* Be able to use ubuntu terminal for printing various types graphs
* Be able to open and read file from the ubuntu terminal
* Use of dynamic arrays and its memory allocation
* Features implemented to improve user experience
* Being able to use characters to form illustrations
* De-Allocate pointers to prevent memory leak during run time

**Diagram/ Illustrations of program design**

**Option1:**

Text

Description automatically generated

**Option2:**

Text

Description automatically generated

**Option3:**

Text

Description automatically generated

**Option4:**

Text

Description automatically generated

**Option5:**

Text

Description automatically generated

**Option6:**

Text

Description automatically generated

**Option7:**

Text

Description automatically generated with low confidence

**Option8:**

Text

Description automatically generated

**Summary of implementation of each module in your program:**

* option1()

reads in config.txt file and stores data from the file

* cityMapReading ()

reads in city.txt file and removes delimiters and stores data in 2d Array(city2DArray)

* option2()

2d Array initializing and memory allocation, prints output with # along with city id in a graph like manner using double for loops

* cloudCoverageMap()

reads in cloud.txt file and removes delimiters and stores data in 2d Array(cloudArray)

* option3()

2d Array initializing and memory allocation, prints output with # along with cloudiness index from cloud cover value using a if else statement in a graph like manner using double for loops

* option4()

2d Array initializing and memory allocation, prints output with # along with cloudiness symbol from cloud cover value using a if else statement in a graph like manner using double for loops

* atmosphericPressure()

reads in pressure.txt file and removes delimiters and stores data in 2d Array(pressureArray)

* option5()

2d Array initializing and memory allocation, prints output with # along with pressure index from pressure value using a if else statement in a graph like manner using double for loops

* option6()

2d Array initializing and memory allocation, prints output with # along with pressure symbol from pressure value using a if else statement in a graph like manner using double for loops

* summaryReport()

generates output for option 7 in main menu. Generate 3 “2d Array” initializing and memory allocation, using a double for loop checks citynumber with cityID to get surrounding area according to cityID and store values in the the array to get average pressure values and average cloudcover value. With these values we can find out the rain probability.

* option7()

shows summaryReport()

**Reflections on program developments (E.g. assumptions made, difficulties faced, what could have been done better, possible enhancemenst in future, WHAT HAVE YOU LEARNT, etc)**

**Assumptions made:**

1. C++ is the same as java, but in fact C++ is very similar in concept but coding is slightly different

**Difficulties faced:**

1. Getting the terminal to print according to the guideline was challenging
2. Remembering to de-allocate memory allocation is one of the tasks that needs to be taken as a habit as it caused me a lot of problems
3. Generating the graphical terminal output was not an easy task and it required a lot of trial an error personally along with online research and it was very time consuming.

**What could have been improved:**

1. Prompt user to enter a valid input instead of just exiting

**Possible enhancements:**

1. Better usage of the 2D arrays

**What I have learnt:**

1. Code in small paragraphs to troubleshoot in case there is any problems in compilation.