Contents

Introduction

The Program generates a random salt based on integer constants for anions and cation. These are stored in a salt class. The user then begins a session and is asked to start testing. He can go and view a list of reagents in the application. The user is prompted to use the ID's of the reagents to create a testing mixture. These reagents are pushed into a queue. If the reagents in the queue match the reagents for the test in the exact order, the test will execute. If not, an error is thrown. This testing happens through a test class which handles cases for reagent tests.

Another class called user is used to store and retrieve the progress of the user while doing the analyses. It also stores success and failures and this can be viewed on the application.

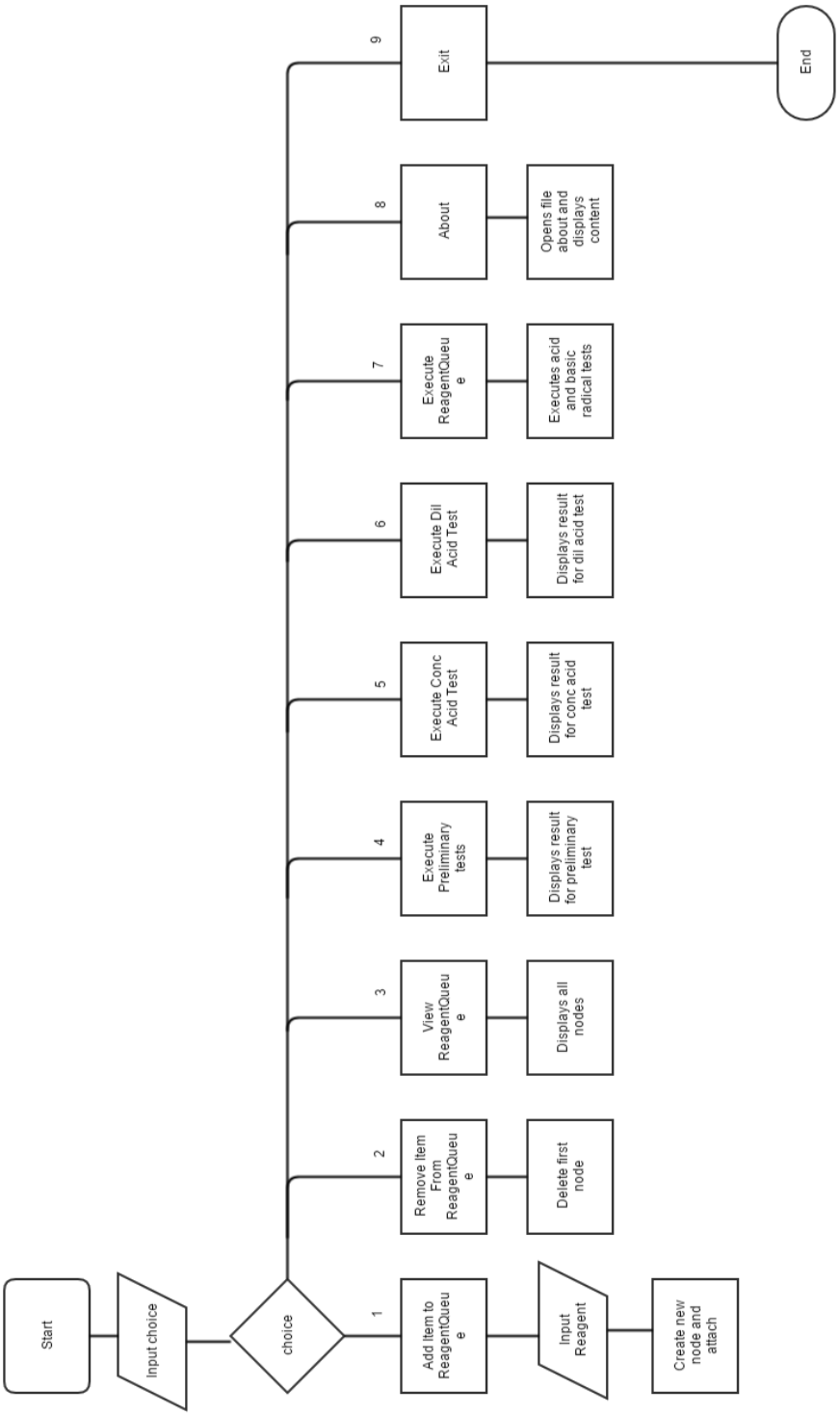
This program utilizes user-defined functions as well as C++'s built in libraries.

Algorithm

1. The user is prompted to enter an option corresponding to an item in the main menu.
2. The entered choice is compared in a switch case:
    a. Choice == 1: User is prompted to enter a chemical name to be added to the reagent queue. A new node is created and attached to the end of the queue.
    b. Choice == 2: The last item from the front is deleted from the queue after confirmation by user
    c. Choice == 3: The queue is displayed with the contents of each node
    d. Choice == 4:
        i. The salt in the user object is compared against a predefined set of conditions for preliminary testing.
        ii. If the conditions are met, the program displays the resulting observation in accordance with salt analysis rules.
    e. Choice == 5:
        i. The salt in the user object is compared against a predefined set of conditions for dilute acid test.
        ii. If the conditions are met, the program displays the resulting observation in accordance with salt analysis rules.

f. Choice == 6:
    i. The salt in the user object is compared against a predefined set of conditions for conc acid test.
    ii. If the conditions are met, the program displays the resulting observation in accordance with salt analysis rules.
g. Choice == 7:
    i. Acid radical test is executed by comparing the given queue with a standard queue using the CompareReagent method. The results of the tests are displayed if the chemicals in the queues match.
    ii. Basic radical test is executed by comparing the given queue with a standard queue using the CompareReagent method. The results of the tests are displayed if the chemicals in the queues match.
h. Choice == 8:
    i. Opens file About.txt and reads content.
    ii. Displays the read data on the screen and awaits user input to continue.
i. Choice == 7:
    i. Ends program.

Flowchart


Start
Input choice
choice
1
Add Item to ReagentQueue
Input Reagent
Create new node and attach
2
Remove Item From ReagentQueue
Delete first node
3
View ReagentQueue
Displays all nodes
4
Execute Preliminary tests
Displays result for preliminary test
5
Execute Conc Acid Test
Displays result for conc acid test
6
Execute Dil Acid Test
Displays result for dil acid test
7
Execute ReagentQueue
Executes acid and basic radical tests
8
About
Opens file about and displays content
9
Exit
End

User Defined Functions

1. **Class Salt:**
    a. **GenerateRandomAnion():** Function generates a random anion from the given list of 11 anions.
    b. **GenerateRandomCation():** This function generates a random anion from the given list of 12 cations.
    c. **getAnion(int):** Returns the generated anion.
    d. **getCation(int):** Returns the generated cation.
2. **Class ReagentQueue**
    a. **ReagentQueue() [constructor]:** Initializes the values of variables.
    b. **add():** Adds a reagent to the queue at the rear.
    c. **getFront():** Returns the value stored in the pointer pointing to the front of the queue.
    d. **add(char[][100], int):** Overloaded function, adds a reagent to the queue at rear. Specifically for adding through ConfirmBasicRadical() and ConfirmAcidRadical().
    e. **del():** Deletes the reagent at the front of the queue.
    f. **ClearQueue():** Deletes all reagents in queue.
    g. **traverse():** Traverses the queue, returning reagVal.
    h. **display():** Displays the value of reagVal.

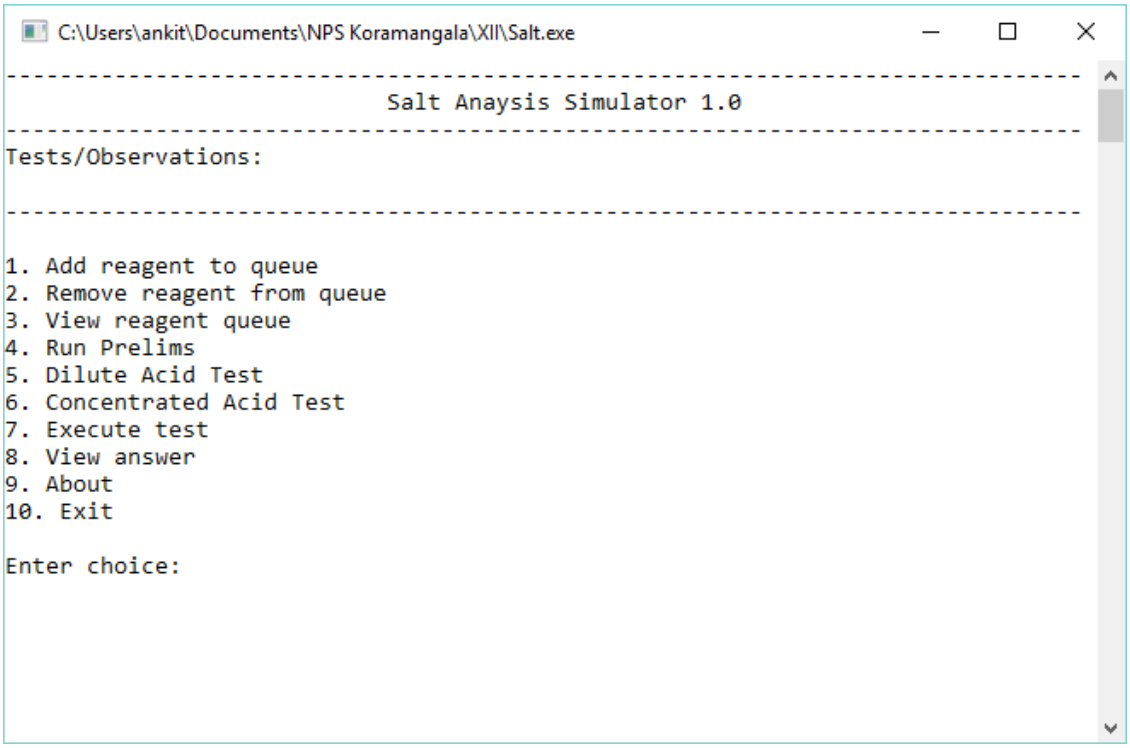
3. **CompareReagent(ReagentQ, ReagentQueue):** Compares the queues to test for equality, and if equal returns 1. Else returns 0.
4. **Class UserModel:**
    a. **UserModel() [constructor]:** Constructor to initialize value of a to 0 and calls generate();
    b. **generate():** Calls the GenerateRandomCation() and GenerateRandomAnion() to set values of cation and anion for the user.
    c. **addReagent():** Adds a reagent to the user queue.
    d. removeReagent(): Removes reagent from the user queue.
    e. **viewQueue():** Traverses and displays the user queue to keep track of the reagents in it.
    f. **displaySalt():** Displays the anion and cation present in the salt.
    g. **prelims(&UserModel):** Compares the value of anion and cation to store the correct result of preliminary tests in observations array.
    h. **DilAcid(&UserModel):** Compares the value of anion to store correct result of dilute acid test in observations array.
    i. **ConcAcid(&UserModel):** Compares the value of anion to store correct result of dilute acid test in observations array.

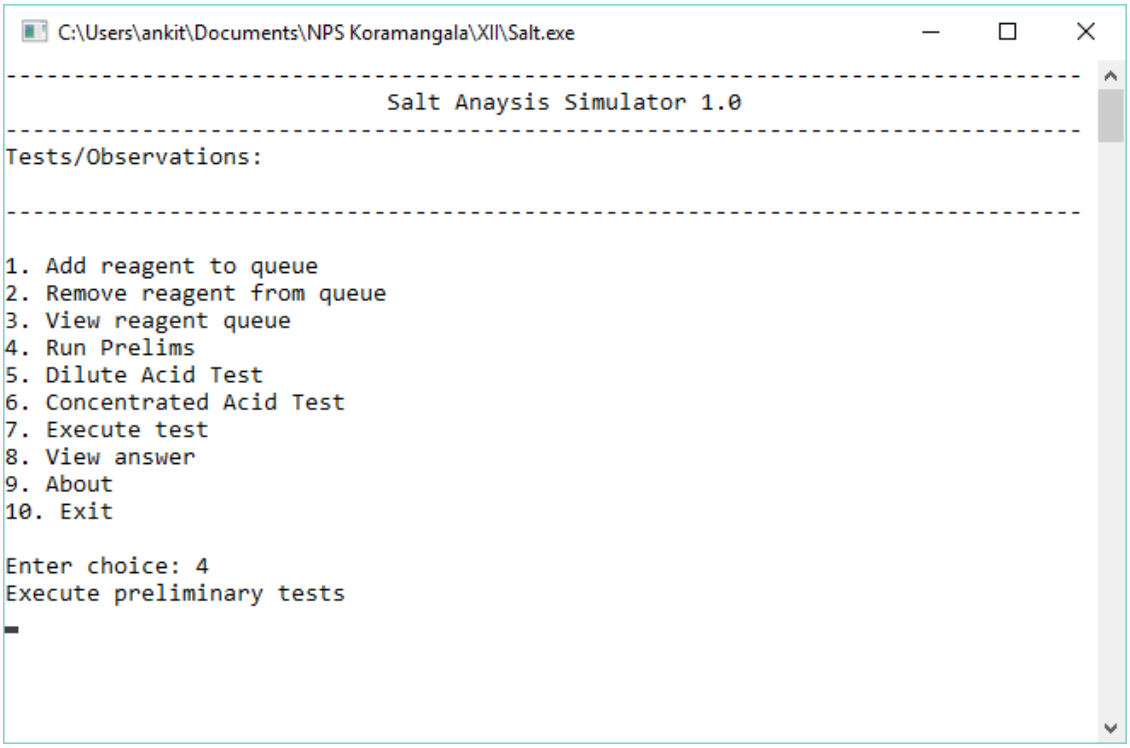
j. **ConfirmAcidRadical(&UserModel):** Compares the value of reagent in the user queue to store correct result of confirmatory anion test in observations array.
k. **ConfirmBasicRadical(&UserModel):** Compares the value of reagent in the user queue to perform inter-group separation and confirmatory cation test, and story the correct result in the observations array.
l. **execute(&UserModel):** Calls the ConfirmAcidRadical() and ConfirmBasicRadical() functions to execute the respective tests.
m. **displayAbout():** Fetches the About.txt file and displays its contents.
n. **DisplayUI(&UserModel):** Generates and displays the user interface.

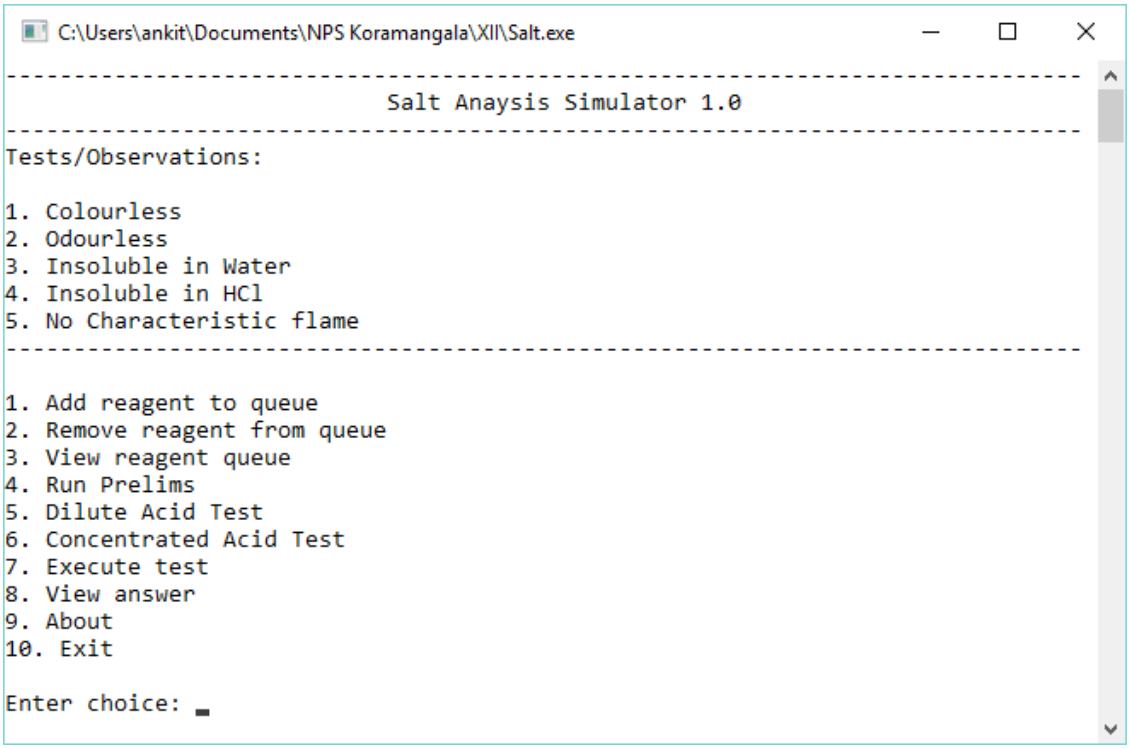
Classes and Structures

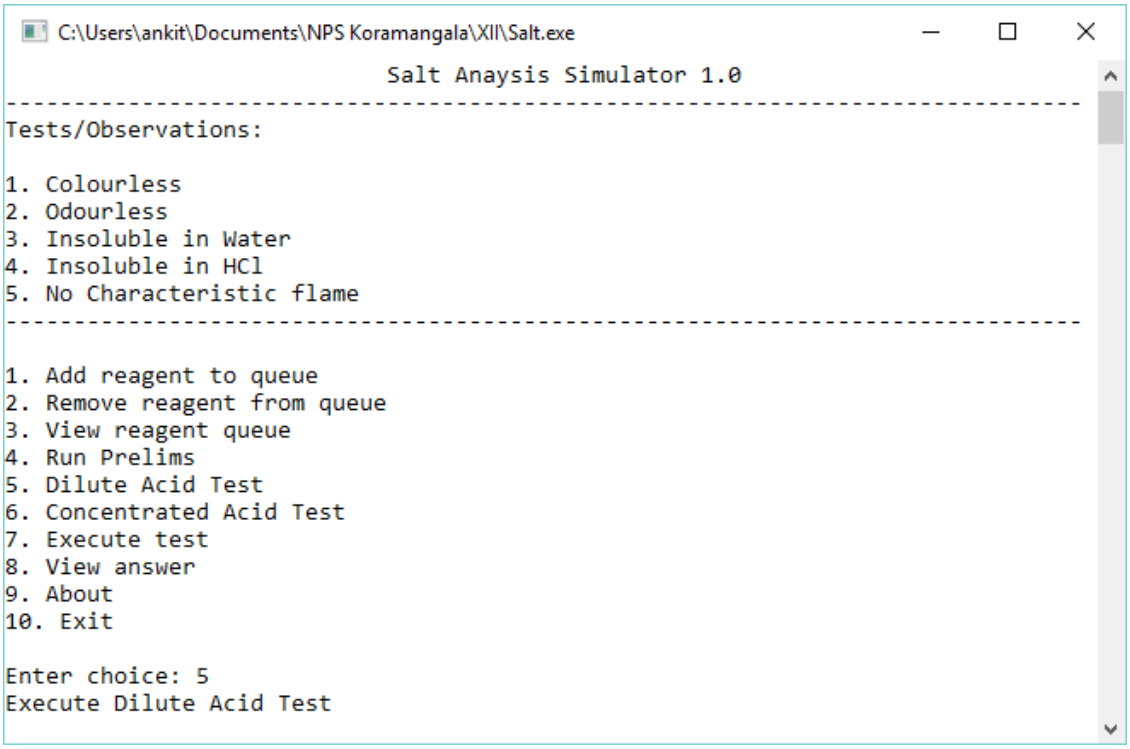
1. **Class ChemicalTests**
    a. Variables for reagents and conditions required.
    b. Function checkTest() to check if salt will react positively with the set of reagents and conditions.
2. **Class Salt**
    a. Variable containing cation and anion
3. **Class User**
    a. Variables for current salt, successes, failures.
    b. Function that writes to binary file so past tested salts can be viewed.
    c. Function that reads past salts tested.
4. **Class ReagentQueue**
    a. It consists of general implementation of a linked list stack with operations such as add, delete and traverse.
    b. It is used to hold a list of type 'Chemical' which serves as a list of reagents.
5. **Structure Chemical**
    a. It holds an integer value which correlated to a constant value held for a particular reagent.
    b. Is a self-referential structure.

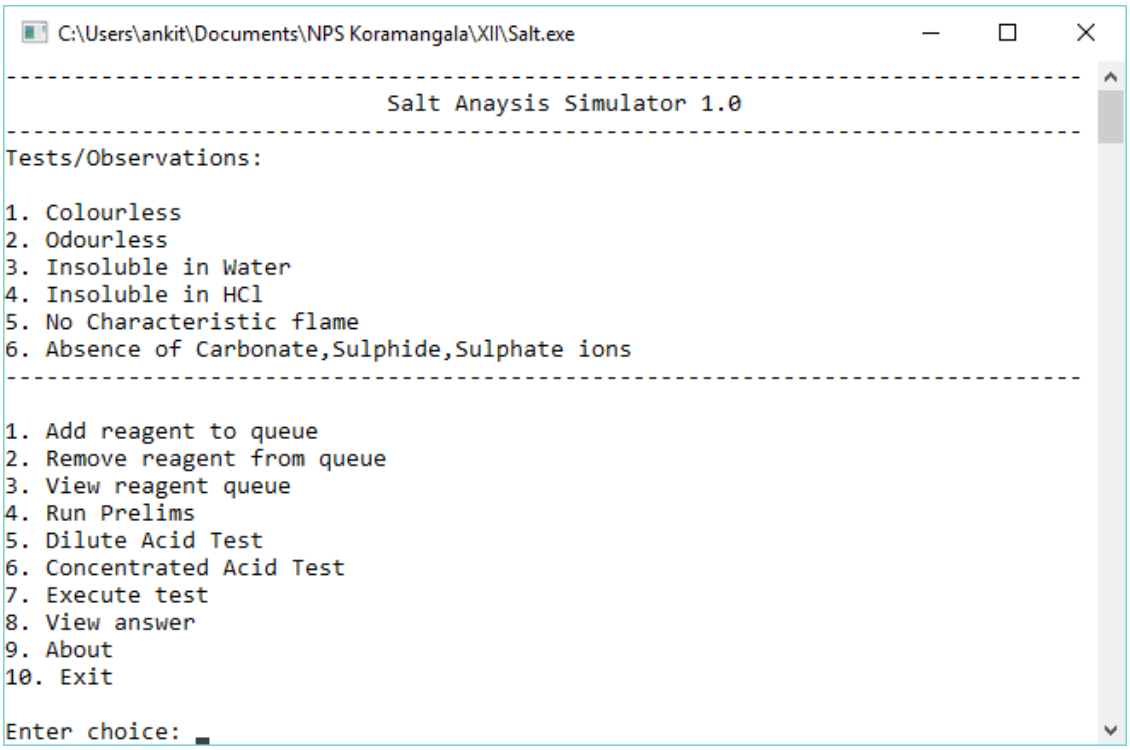
Screenshots

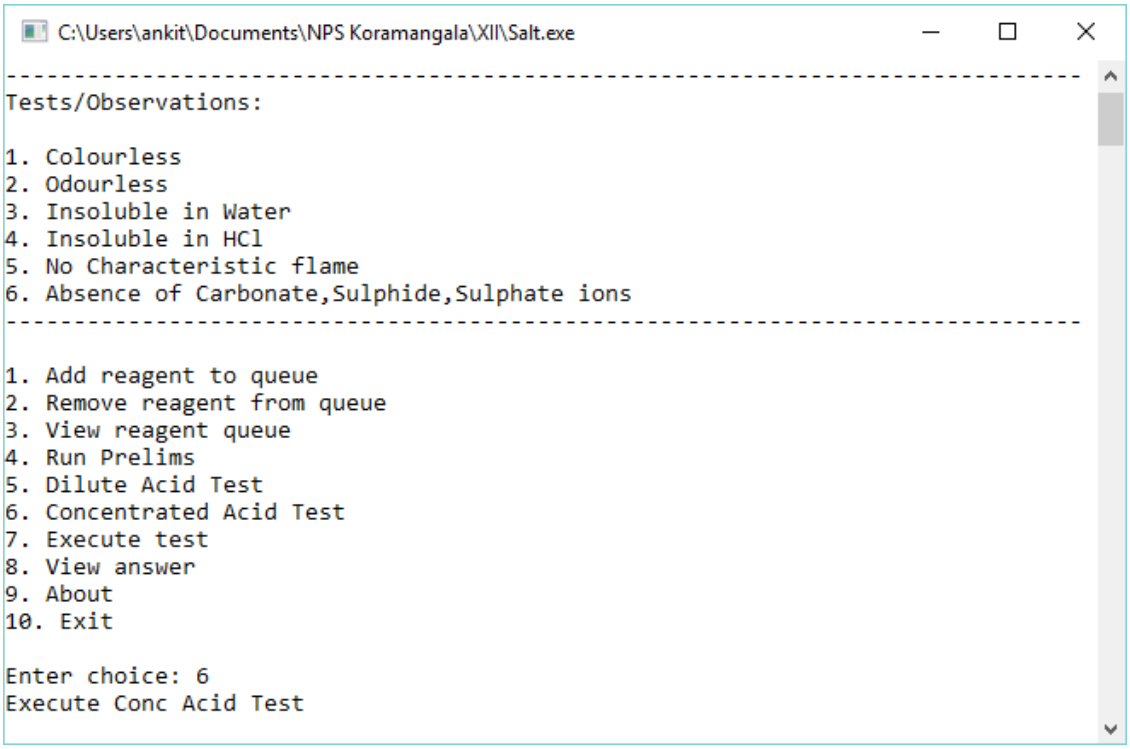


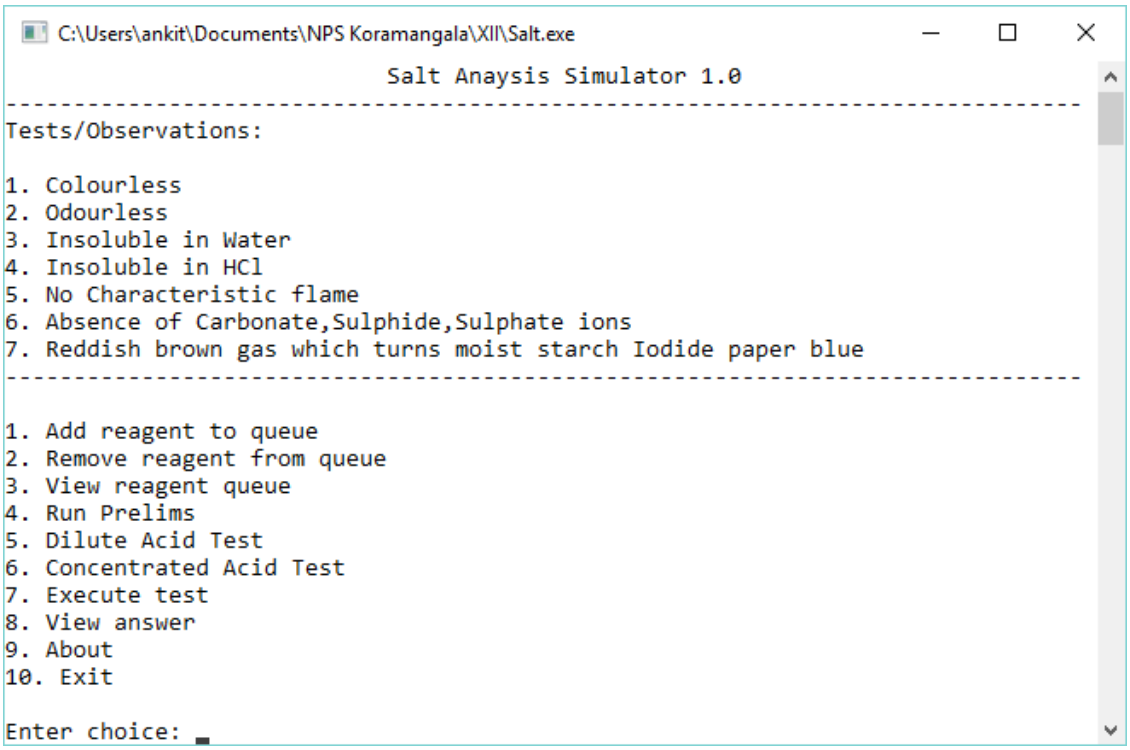


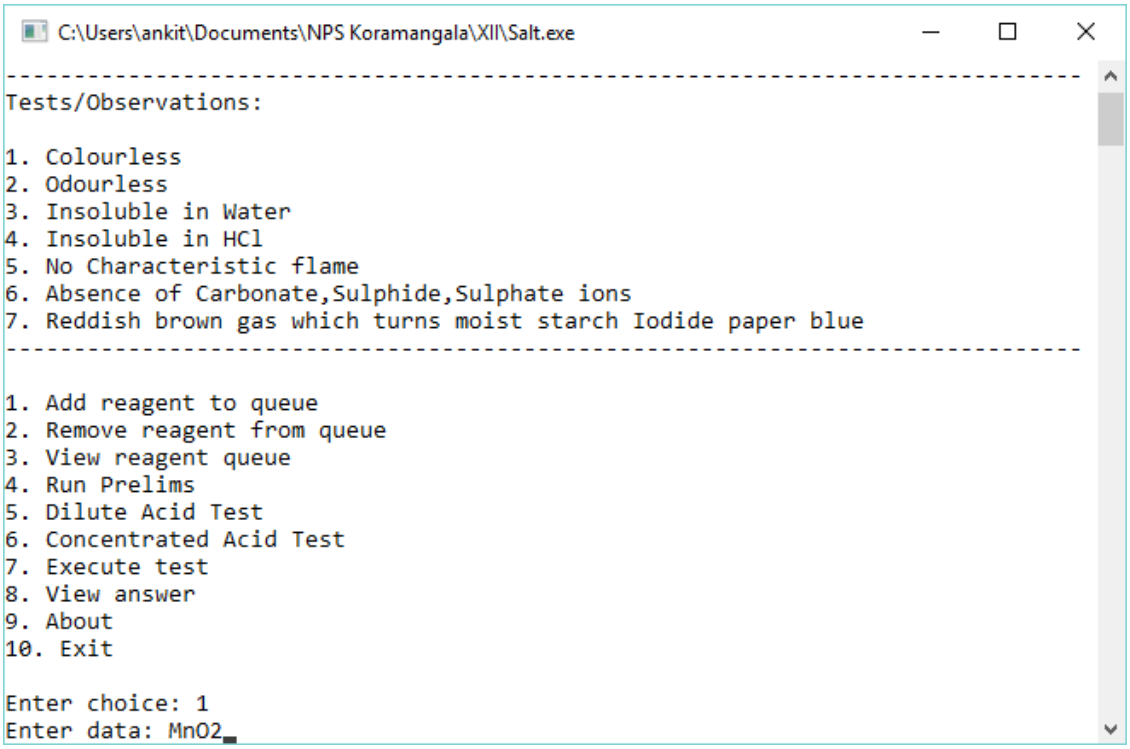

C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Salt Anaysis Simulator 1.0
Tests/Observations:
1. Colourless
2. Odourless
3. Insoluble in Water
4. Insoluble in HCl
5. No Characteristic flame
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice:

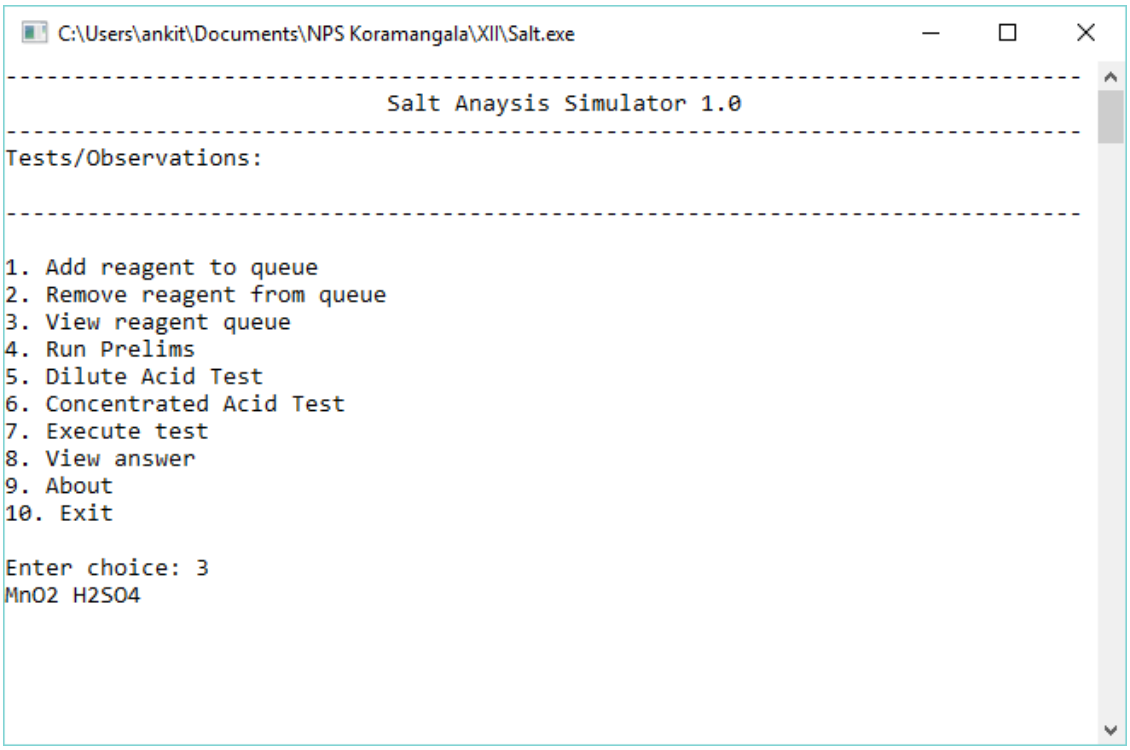

C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Salt Anaysis Simulator 1.0
Tests/Observations:
1. Colourless
2. Odourless
3. Insoluble in Water
4. Insoluble in HCl
5. No Characteristic flame
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice: 5
Execute Dilute Acid Test

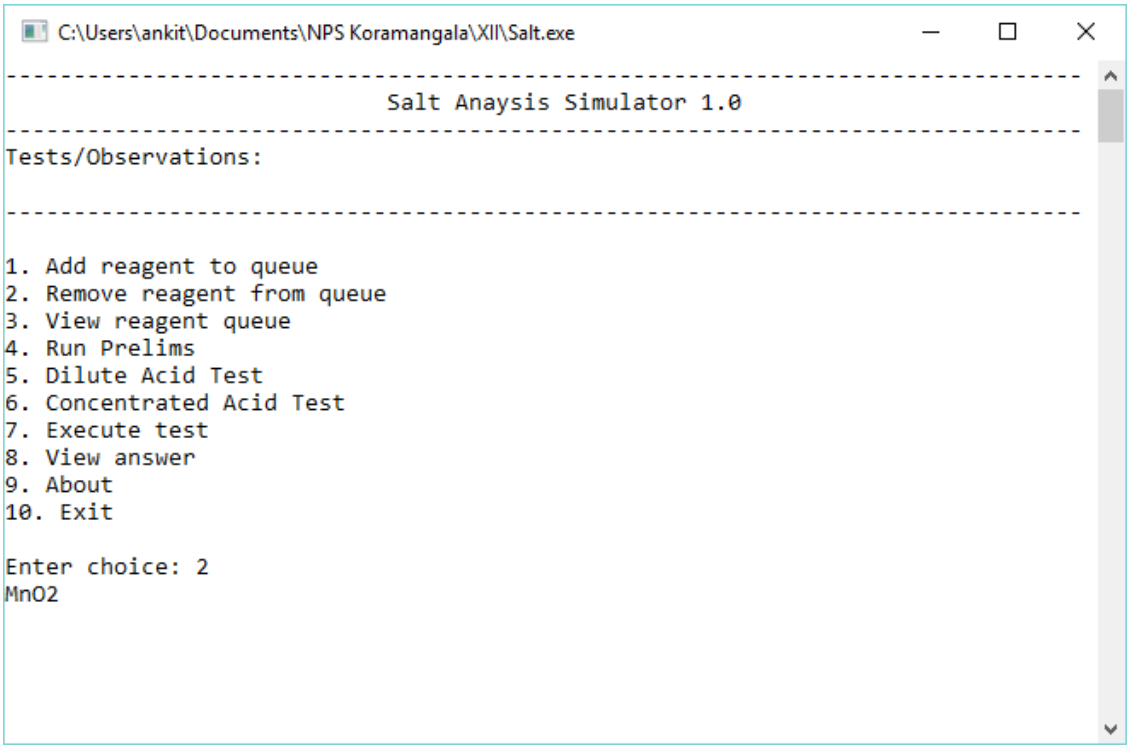

C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Salt Anaysis Simulator 1.0
Tests/Observations:
1. Colourless
2. Odourless
3. Insoluble in Water
4. Insoluble in HCl
5. No Characteristic flame
6. Absence of Carbonate,Sulphide,Sulphate ions
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice:

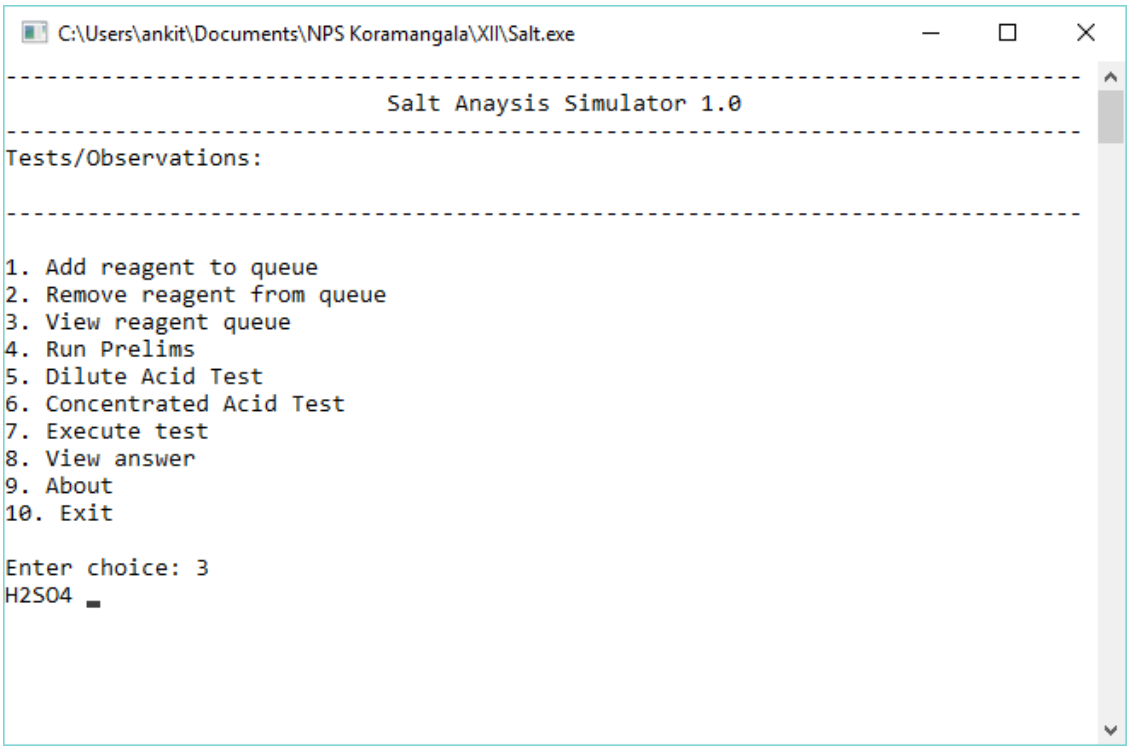

C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Tests/Observations:
1. Colourless
2. Odourless
3. Insoluble in Water
4. Insoluble in HCl
5. No Characteristic flame
6. Absence of Carbonate,Sulphide,Sulphate ions
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice: 6
Execute Conc Acid Test

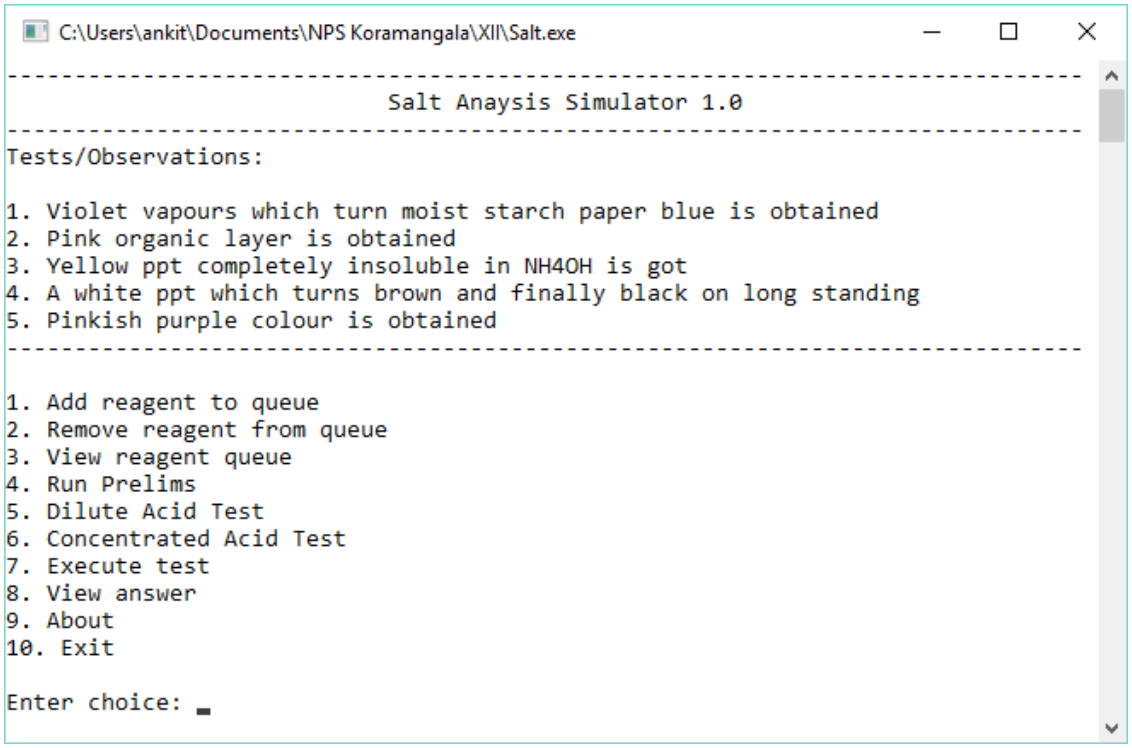

C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Salt Anaysis Simulator 1.0
Tests/Observations:
1. Colourless
2. Odourless
3. Insoluble in Water
4. Insoluble in HCl
5. No Characteristic flame
6. Absence of Carbonate,Sulphide,Sulphate ions
7. Reddish brown gas which turns moist starch Iodide paper blue
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice:


C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Tests/Observations:
1. Colourless
2. Odourless
3. Insoluble in Water
4. Insoluble in HCl
5. No Characteristic flame
6. Absence of Carbonate,Sulphide,Sulphate ions
7. Reddish brown gas which turns moist starch Iodide paper blue
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice: 1
Enter data: MnO2


C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Salt Anaysis Simulator 1.0
Tests/Observations:
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice: 3
MnO2 H2SO4


C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Salt Anaysis Simulator 1.0
Tests/Observations:
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice: 2
MnO2


C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Salt Anaysis Simulator 1.0
Tests/Observations:
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice: 3
H2SO4


C:\Users\ankit\Documents\NPS Koramangala\XII\Salt.exe
Salt Anaysis Simulator 1.0
Tests/Observations:
1. Violet vapours which turn moist starch paper blue is obtained
2. Pink organic layer is obtained
3. Yellow ppt completely insoluble in NH4OH is got
4. A white ppt which turns brown and finally black on long standing
5. Pinkish purple colour is obtained
1. Add reagent to queue
2. Remove reagent from queue
3. View reagent queue
4. Run Prelims
5. Dilute Acid Test
6. Concentrated Acid Test
7. Execute test
8. View answer
9. About
10. Exit
Enter choice:

Acknowledgments

I would like to thank the school for providing me with lab space and the required equipment for conducting my experiment. I would like to thank Chandita ma'am for her valuable inputs and constant support throughout my endeavor. This would not have been possible without her. I would also like to thank Ms Anupama who helped and provided for our needs.

Bibliography

Websites referred to -

1. www.google.com
2. www.meritnation.com
3. www.stackoverflow.com
4. www.youtube.com

Books referred to -

1. Sumitha Arora Grade 11
2. Smitha Arora Grade 12