**Name**: Tanveen Kaur

**Student ID**: 1014603

**Program 3**

**Postman Problem**

**Due Date**: 10/19/2016

**Table of Contents**

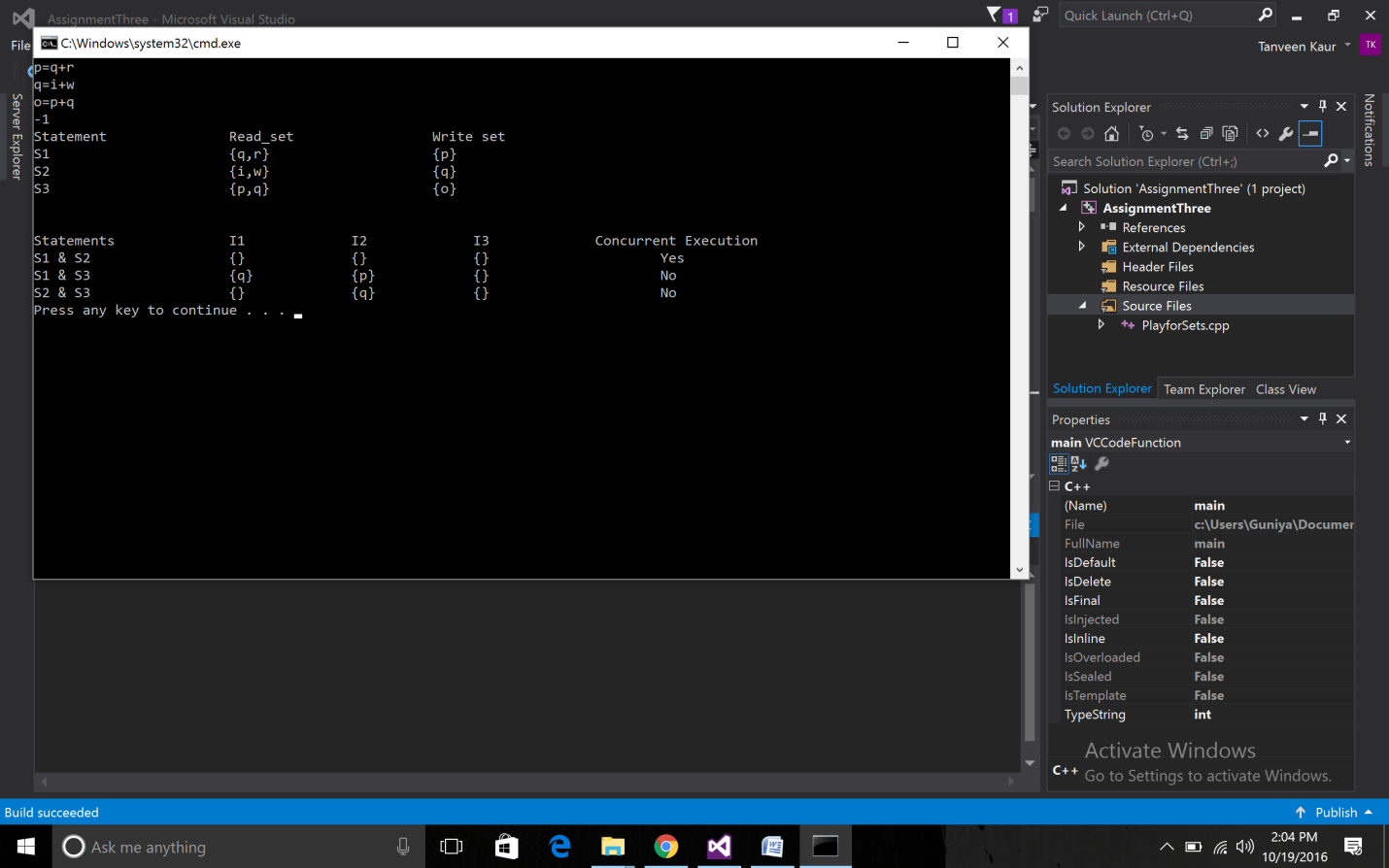
Screenshots 1

Documented Source Code .2

Things Learned…………………………………………………………………………….3

Comments………………………………………………………………………………….3

**Screenshots: Output**

****

**Source Code:**

#include <iostream>

#include <vector>

#include <string>

#include <utility>

#include <cstdio>

#include <cstdlib>

#include <cstring>

using namespace std;

vector<pair<char, vector<char> > > store;

// function to read and write sets of inputs

void print\_read\_write\_set() {

cout << "Statement\t\t";

char read\_Set[] = "Read\_set";

printf("%-25sWrite set\n", read\_Set);

for (int i = 0; i<store.size(); i++) {

cout << "S" << i + 1 << " \t\t";

char makeset[20];

memset(makeset, '\0', sizeof(makeset));

makeset[0] = '{';

int k = 1;

for (int j = 0; j<store[i].second.size(); j++) {

makeset[k++] = store[i].second[j];

makeset[k++] = ',';

}

makeset[--k] = '}';

printf("%-25s{%c}\n", makeset, store[i].first);

}

cout << endl << endl;

}

// function to print vector

void printvec(vector<char> x) {

char makeset[20];

memset(makeset, '\0', sizeof(makeset));

makeset[0] = '{';

int k = 1;

for (int j = 0; j<x.size(); j++) {

makeset[k++] = x[j];

makeset[k++] = ',';

}

if (k == 1) {

makeset[k] = '}';

}

else

makeset[--k] = '}';

printf("%-15s", makeset);

}

// function to find intersection of 2 read sets

vector<char> read\_and\_read(int x, int y) {

vector<char> ans;

for (int i = 0; i<store[x].second.size(); i++) {

char p = store[x].second[i];

for (int j = 0; j<store[y].second.size(); j++) {

if (p == store[y].second[j]) {

ans.push\_back(p);

break;

}

}

}

return ans;

}

// function to find intersection of read and write set

vector<char> write\_and\_read(int x, int y) {

vector<char> ans;

char p = store[x].first;

for (int j = 0; j<store[y].second.size(); j++) {

if (p == store[y].second[j]) {

ans.push\_back(p);

break;

}

}

return ans;

}

// function to find intersection of 2 write sets

vector<char> write\_and\_write(int x, int y) {

vector<char> ans;

char p = store[x].first;

char q = store[y].first;

if (p == q) { // comparing the 2 elements

ans.push\_back(p);

}

return ans;

}

// function to print intersection of sets

void print\_intersection() {

cout << "Statements\t\t";

printf("I1 I2 I3 Concurrent Execution\n");

for (int i = 0; i<store.size(); ++i) {

for (int j = i + 1; j<store.size(); ++j) {

cout << "S" << i + 1 << " & S" << j + 1 << " \t\t";

vector<char> i1 = read\_and\_read(i, j);

printvec(i1);

vector<char> i2 = write\_and\_read(i, j);

printvec(i2);

vector<char> i3 = write\_and\_write(i, j);

printvec(i3);

if (i1.size() == 0 && i2.size() == 0 && i3.size() == 0) {

cout << "\t Yes" << endl;

}

else {

cout << "\t No" << endl;

}

}

}

}

int main()

{

string inp = "1";

while (1)

{

getline(cin, inp);

if (inp == "-1") {

break;

}

char p = inp[0];

vector<char> q;

for (int i = 1; i<inp.size(); i++) {

if (isalpha(inp[i])) {

q.push\_back(inp[i]);

}

}

store.push\_back(make\_pair(p, q));

}

print\_read\_write\_set();

print\_intersection();

return 0;

}

**Things Learned:**

* **Use of STL components.**

**Comments:** Implementation successful with the desired output.