

Script started on 2021-03-02 10:52:44-0600
m_sadafl@ares:~\$ cat set.info
Name: Madiha Sadaf
Class: CSC122 W01

Lab: "You Want Me To Do What Problems?"
Option: Encapsulate the management of your problems list.

Level: 4
Level: +2
Total Level: 6

Description:

This program takes in hyphenated assignment list and prints out a list of all the problems assigned to the user.

m_sadafl@ares:~\$ cat set.tpq
Thought Provoking Questions:

- 1) In a class.
- 2) You can test if the input == '\\' or if the input == '\"'. Then you can use getline(cin, var, '\\') for single quote or getline(cin, var, '\"') for a double quote.
- 3) By using the quotes to start the input as well as to end.
- 4) By using insertion sort.
- 5) By using cin.ignore().
- 6) You can wrap the long output line to multiple lines by tracking the length of the list with a counter variable. The counter resets after reaching its max after adding an newline.

```
m_sadafl@ares:~$ cat set.cpp
#include <iostream>
#include <vector>
#include "set.h"
#include "set2.h"
```

```
using namespace std;
```

```
void set::out() const //Outputs problems to do
```

```
{
    cout << "Do problem";
    vector<short>::size_type
        out_max_length = 70; // output length
    vector<short>::size_type
        count = 13;
    if (list.size() == 1) // one item in list
    {
        cout << " " << list[0] << " of " << data << "." <<
            endl;
    }
    else // more than one item in list
    {
        cout << "s ";
        count = count + 2;
        for (vector<short>::size_type pos = 0;
            pos < list.size() - 2; pos++)
        {
            if (count > out_max_length) // big wraps
            {
                cout << endl;
                count = 0;
            }
            cout << list[pos] << ", ";
            count = count + num_length(list[pos]) + 2; // adds to total
        }

        count = count + num_length(static_cast<double>
            (list.size() - 2));
        if (count > out_max_length) // big wraps
        {
            cout << endl;
            count = 0;
        }
        cout << list[list.size() - 2];
        // counts what's left to print and checks if needs wrap
        count = count + 10 + num_length(static_cast<double>
            (list.size() - 1)) + data.length();

        if (count > out_max_length)
        {
            cout << endl;
            count = 0;
        }
        cout << " and " << list[list.size() - 1] << " of "
            << data << "." << endl;
    }
}
```

```

}

void set::input_list(std::string const & prompt)
{
    cout << prompt;
    bool get_input = false;
    char input;

    while (!get_input) // ignores input until ", ', or a letter
        // is reached
    {
        cin >> input;
        if (input == '"' || input == '\'' || isalpha(input))
            //isalpha to check if the character is alphabet
        {
            get_input = true;
        }
    }

    string get_name; // inputs name
    if (input == '\\') // reads input until end
    {
        getline(cin, get_name, '\\');
        data = get_name;
    }
    else if (input == '\"') // ""
    {
        getline(cin, get_name, '\"');
        data = get_name;
    }
    else // Input was a char
    {
        char get_letter;
        data = input; //adds first letter to beginning of data;
        while (isalpha(cin.peek())) // while next input in
            // line is a letter
        {
            // input the letter and add it
            //to the end of the data
            cin >> get_letter;
            data = data + get_letter;
        }
    }

    // For inputting numbers

    char get_sign;

```

```

short get_number;
short get_number2;
cin >> get_number;
list.push_back(get_number);
while (cin.peek() != '\n')
{
    cin >> get_sign;
    if (get_sign == ',')
    {
        cin >> get_number;
        insert_sort(get_number);
    }
    else if (get_sign == '-')
    {
        cin >> get_number2;
        while (get_number < (get_number2))
        {
            get_number++;
            insert_sort(get_number);
        }
    }
}

void set::insert_sort(const short & input)
{
    vector<short>::size_type pos = 0;
    bool done = false;
    // just insert if nothing in list, no need for sorting
    if (list.size() == 0)
    {
        list.push_back(input);
        done = true;
    }
    if (!done) // finds position
    {
        bool match = false; // shows found a match
        while (pos < list.size() && !done)
        {
            if (list[pos] == input)
            {
                done = true;
                match = true;
            }
            else if (list[pos] > input)
            {
                done = true;

```

```

    }
    if (!done) // input > list[pos]
    {
        pos++;
    }
}
if (!match)
{
    list.push_back(0); //inc list size by 1
    vector<short>::size_type index = (list.size()
        - 1);
    // moves found data to right to make space
    while (index > pos)
    {
        list[index] = list[index - 1];
        index--;
    }
    list[pos] = input; // puts input in the right pos
    done = true;
}
}
}
}

```

m_sadafl@ares:~\$ cat set.h

```

#include <string>
#include <vector>
#ifndef set_h
#define set_h

class set
{
    std::string data;
    std::vector<short>list;
    void insert_sort(const short & input);

public:
    set() : data(), list() {}
    void input_list(std::string const & prompt);
    void out() const;
};

#endif /*set_h*/

```

m_sadafl@ares:~\$ cat set_main.cpp

```

#include <iostream>
#include <cstdlib>

```

```

#include <string>
#include "set.h"

using namespace std;

int main()
{
    set s;
    string prompt = "\t\t\tWelcome to the Assignment"
        " Clarifier!\n"
        "\nEnter your problem list: ";
    s.input_list(prompt);
    s.out();
}

```

m_sadafl@ares:~\$ CPP set set_main
set.cpp...
set_main.cpp***

m_sadafl@ares:~\$./set_main.out
Welcome to the Assignment Clarifier!

Enter your problem list: J8
Do problem 8 of J.
m_sadafl@ares:~\$
m_sadafl@ares:~\$./set_main.out
Welcome to the Assignment Clarifier!

Enter your problem list: L4-5, 1-3, 7-10, 8-12
Do problems 1, 2, 3, 4, 5, 7, 8, 9, 10, 11 and 12 of L.
m_sadafl@ares:~\$
m_sadafl@ares:~\$./set_main.out
Welcome to the Assignment Clarifier!

Enter your problem list: S 1, 2, 5
Do problems 1, 2 and 5 of S.
m_sadafl@ares:~\$
m_sadafl@ares:~\$./set_main.out
Welcome to the Assignment Clarifier!

Enter your problem list: s 1-3, 5-7
Do problems 1, 2, 3, 5, 6 and 7 of s.
m_sadafl@ares:~\$
m_sadafl@ares:~\$ exit
exit

Script done on 2021-03-02 10:55:06-0600