**PERSONAL SOFTWARE PROCESS (PSP)**

**Author: Jagmeet Singh Gurm**

**Instructor: Dr. Brent Auernheimer**

**Course:CSCI 250- Advanced Software Engineering**

**Assignment1, Fall 2014**

**CALIFORNIA STATE UNIVERSITY, FRESNO**

**Table of Contents**

1. Introduction………………………………………3
2. Planning Phase………………………………….4

Requirements Statements………………..4

1. Development Phase………………………….5

Design…………………………………………….…5

Code………………………………………………...8

Compile…………………………………………….8

Test…………………………………………………..9

1. Postmortem…………………………………….15

**Introduction:**

The program is to store n real numbers in linked list and then calculate their mean and standard deviation. And this is to be implemented by following Personal Software Process(PSP0).

During all the phases of the project, the PSP0 entry and exit criteria has been followed.

**Planning Phase**

**Requirements Statement**:

The program is supposed to calculate the mean and the standard deviation of N Real numbers.

The N Real numbers can be read from any source, be it user-input from keyboard, file, or some other source.

The input of N real numbers to perform required calculations should ideally be stored in a data structure like Linked List. But resources like static arrays, variable, or other data structures can also be used to store data, if needed.

Test the program thoroughly by using at least the following two test cases from Table 1.

|  |  |
| --- | --- |
| **Test Case 1** | **Test Case 2** |
| Estimate Proxy Size | Development Hours |
| 160 | 15.0 |
| 591 | 69.9 |
| 114 | 6.5 |
| 229 | 22.4 |
| 230 | 28.4 |
| 270 | 65.9 |
| 128 | 19.4 |
| 1657 | 198.7 |
| 624 | 38.8 |
| 1503 | 138.2 |

**Table 1**

The Expected Test Results are given in Table 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tests** | **Expected Value** | | **Actual Value** | |
|  | Mean | Standard Deviation | Mean | Standard Deviation |
| Table 1: Test Case 1 | 550.6 | 572.03 | 550.6 | 572.027 |
| Table 1: Test Case 2 | 60.32 | 62.26 | 60.32 | 62.2558 |

**Table 2**

**Development Phase**

**DESIGN:**

**The Linked List** Data Structure will be used to store the N Real Numbers for calculating Mean and Standard Deviation. The data will be entered by the user from keyboard.

The program will be made in C++ language in Microsoft’s Visual Studio 2013 environment.

The program will have node and linkedlist header files. The CPP files will be linkedlist and main.

Each node will point to the next node and its previous node with first node being the head and the last one being the tail.

NODE.H

The node class in the node header file will contain the followings:

* a variable of **float type-data** to store actual value of N Real Numbers.
* **next** and prev pointers of node type which will point to the next and the previous nodes of the list respectively.

The scope of its elements will be public so that they can be accessed by linkedlist class.

LINKEDLIST.H

The linked list will contain the following functions and items:

* **constructor** which is called when a list is created.
* **destructor** which destroys the list created.
* **Create-** to create the actual list. It accepts data of float type as parameter.
* **Mean**- to calculate the mean of the N Real Numbers of linked list. Will return float value.
* **SD**-to calculate Standard Deviation of N Real Numbers. Will return float value.
* **Head pointer** of node type which will point to the head of the list.
* **Tail pointer** of node type to point to tail or last element of the list.
* **Total, avg** to store the total of the list and the mean of the list respectively.
* **Count-** to count the number of real numbers added in the list. Will act as counter.

The scope of functions will be public whereas of other items will be private.

LinkedList.cpp

This file will contain all the definitions of the functions of the linkedlist project.

* Linkedlist()-the constructor will also initialize count counter apart from initializing head and tail pointers to Null.
* ~linkedlist()-destructor to delete the linkedlist afterwards so as to release the space occupied by linkedlist.
* Create()- will create the whole linkedlist. It will receive input parameter of float type.

Create new node;

If (head==null)

Point head and tail to node;

Increment count;

Else

Tail’s next points to each new node created;

Each new node’s prev points to the tail;

New node=tail;

Increment count;

* Mean()- will return the average of the real numbers stored in linked list.

a temporary pointer will be used to traverse the whole list starting from head.

If(head==null)

Return;

else

While(t!=null)

Total=total+t->data;

Move next node;

avg=total/count;

Return avg;

* SD()-will return the standard deviation of the linked list.

A temporary pointer will be used to traverse the whole list.

The mean calculated from mean function will be used in standard deviation calculation.

If(head==null)

Return;

Else

While(t!=null)

a=a+t->data^2;

return sqrt(a/count-1);

Main.cpp

The linkedlist will be created here. All the data, that is the N real numbers will be added by the user from here. All the functions will be called from here only.

The user will be asked to enter the number of values it want to enter. This will be stored in n(int).

PS: The Design got edited a couple of times. First, a counter was added to keep track of the number of elements in the list. Second, total and avg variables were added to mean function.

CODE: See the following files:

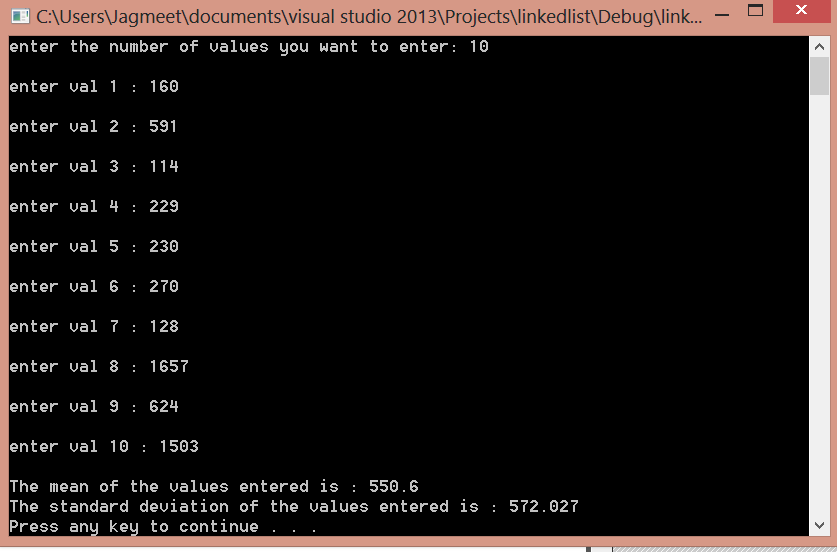
* NODE.H
* LINKEDLIST.H
* LINKEDLIST.CPP
* MAIN.CPP

**COMPILE**: The program got compiled successfully after a minor error was removed.

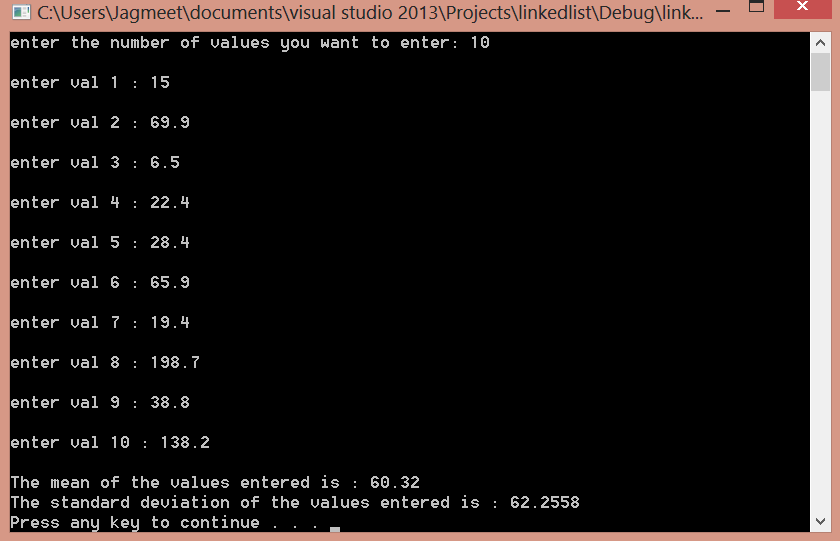
**TEST:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tests** | **Expected Value** | | **Actual Value** | |
|  | **Mean** | **Standard Deviation** | **Mean** | **Standard Deviation** |
| **Table 1: Test Case 1** | 550.6 | 572.03 | 550.6 | 572.027 |
| **Table 1: Test Case 2** | 60.32 | 62.26 | 60.32 | 62.2558 |
| **Test Case 3** | 638.9 | 625.633981 | 638.9 | 625.634 |
| **Test Case 4** | 394 | 164.7 | 394 | 164.712 |
| **Test Case 5** | 5 | 2.24 | 5 | 2.23607 |
| **Test Case 6** | 49.2 | 17 | 49.2 | 16.9954 |

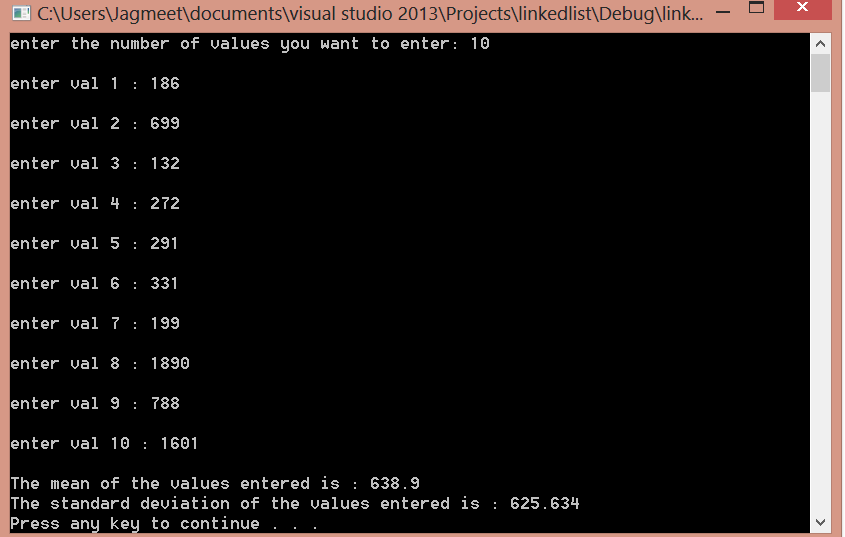
**Test Case 1:**



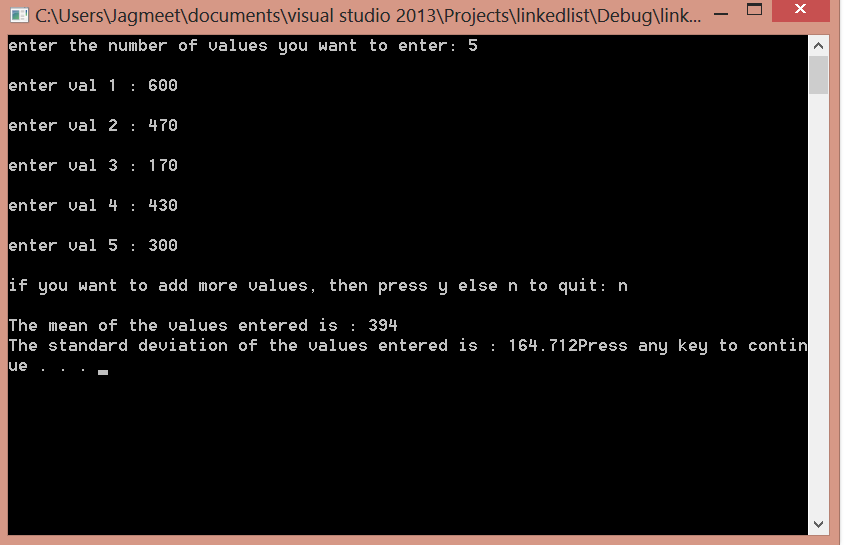
**Test Case 2:**



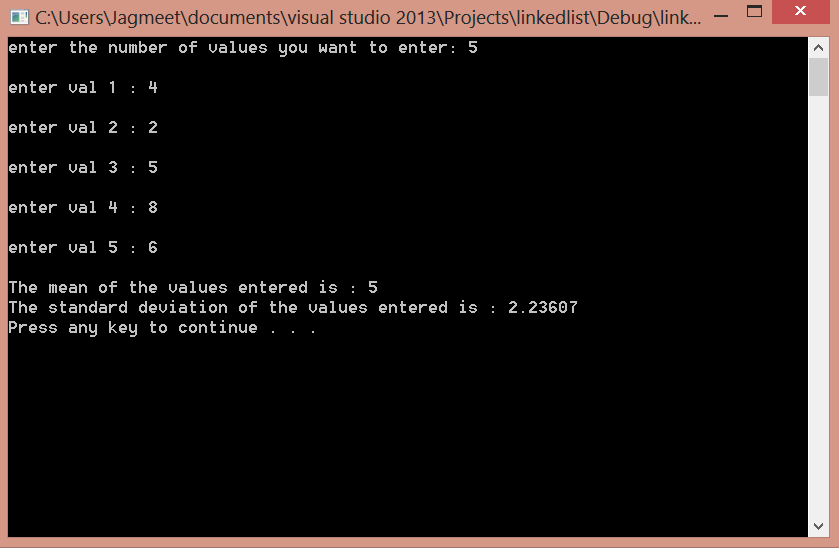
**Test Case 3:**



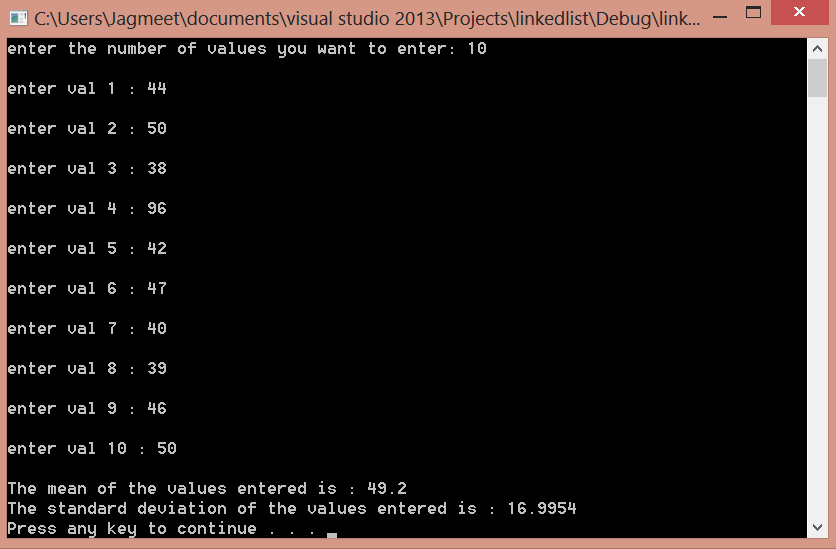
**Test Case 4:**



**Test Case 5:**



Test Case 6:



**POSTMORTEM:** All the test results showed that the program runs well and is reliable. The first 3 test cases are from the PSP given assignment kit. The 4th is from <http://www.mathsisfun.com/data/standard-deviation.html> , 5th from <http://standard-deviation.appspot.com/>, and 6th from <http://www.ltcconline.net/greenl/courses/201/descstat/mean.htm> .

In all the test results, the expected value matches the actual value.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tests** | **Expected Value** | | **Actual Value** | |
|  | **Mean** | **Standard Deviation** | **Mean** | **Standard Deviation** |
| **Table 1: Test Case 1** | 550.6 | 572.03 | 550.6 | 572.027 |
| **Table 1: Test Case 2** | 60.32 | 62.26 | 60.32 | 62.2558 |
| **Test Case 3** | 638.9 | 625.633981 | 638.9 | 625.634 |
| **Test Case 4** | 394 | 164.7 | 394 | 164.712 |
| **Test Case 5** | 5 | 2.24 | 5 | 2.23607 |
| **Test Case 6** | 49.2 | 17 | 49.2 | 16.9954 |

Further, all the defects founded have been recorded in the Defect Records Log. The data in it has been verified to the best of my knowledge. Also, the Time Recording Log is complete and verified to the best of my knowledge. Finally, the Project Plan Summary Form is verified to be complete and accurate.