**Lab Description:**

This lab focuses on a review of arrays and finding the mean average of an array using simple functions. The bonus of this lab also gives a brief touch on pointers and dynamically allocating as well as dynamically resizing arrays using pointers in order to efficiently run a program without knowing what the user will input.

**Problem 1 Source Code:**

#include <iostream>

/\*

Ryan Rosiak

Lab-1 Problem 1

COSC 220

8/27/19

\*/

double mean(int\*, int); // Function that returns a double that is the mean of the array passed using bracket

// notation

double mean2(int\*, int); // Function that returns a double that is the mean of the array passed using

// pointer notation

int main() {

int size;

std::cout << "Enter the size of the array: " << std::endl; // asks the user for the size of the array

std::cin >> size;

int \*arr;

arr = new int[size]; // dynamically allocates memory for the array

for (int i = 0; i < size; i++) { // Loops through and asks the user to enter numbers to

// fill the array

std::cout << "Enter a number: " << std::endl;

std::cin >> arr[i];

}

std::cout << "Mean called: " << mean(arr, size) << std::endl;

std::cout << "Mean2 called: " << mean2(arr, size) << std::endl;

delete [] arr; // delete the dynamically allocated array

return 0;

}

double mean(int \*arr, int size) { // Passed in: array and its size

double total = 0;

for (int i = 0; i < size; i++) { // Adds up all of the array elements

total += arr[i];

}

return total / size; // returns the mean

}

double mean2(int \*arr, int size) { // Passed in: array using pointer and its size

double total = 0;

for (int i = 0; i < size; i++) { // Adds up all of the array elements

total += \*(arr + i);

}

return total / size; // returns the mean

}

**Problem 1 Sample Output:**

**Sample 1:**

Enter the size of the array:

6

Enter a number:

5

Enter a number:

4

Enter a number:

7

Enter a number:

8

Enter a number:

91

Enter a number:

54

Mean called: 28.1667

Mean2 called: 28.1667

**Sample 2:**

Enter the size of the array:

20

Enter a number:

5

Enter a number:

4

Enter a number:

1

Enter a number:

2

Enter a number:

8

Enter a number:

7

Enter a number:

-5

Enter a number:

6

Enter a number:

4

Enter a number:

5

Enter a number:

7

Enter a number:

-2

Enter a number:

4

Enter a number:

1

Enter a number:

1

Enter a number:

3

Enter a number:

6

Enter a number:

8

Enter a number:

9

Enter a number:

4

Mean called: 3.9

Mean2 called: 3.9

**Sample 3:**

Enter the size of the array:

2

Enter a number:

25

Enter a number:

41

Mean called: 33

Mean2 called: 33

**Bonus Source Code:**

#include <iostream>

/\*

Ryan Rosiak

Lab-1 Problem 1 Bonus

COSC 220

8/27/19

\*/

double mean(int\*, int); // Function that returns a double that is the mean of the array passed using bracket

// notation

double mean2(int\*, int); // Function that returns a double that is the mean of the array passed using

// pointer notation

int \*resizeArray(int \*arr, int currentSize) {

int \*buff; // Create new pointer for new memory allocation

buff = new int[currentSize + 1]; // Dynamically allocate pointer array to size of current array plus one

for (int i = 0; i < currentSize; i++) {

buff[i] = arr[i]; // Copy all of the original elements of the original array to the new buffer pointer array

}

delete [] arr; // Delete original array to avoid memory leak

return buff; // Return newly allocated array that is one larger than before

}

int main() {

int size = 1;

int \*arr;

arr = new int[size];

int input;

for (int i = 0; input != -1; i++) { // Loops through and asks the user to enter numbers to

// fill the array

std::cout << "Enter a number: (-1 to exit)" << std::endl;

std::cin >> input;

if (input == -1) { // User input is finished

break;

}

if (i > (size - 1)) { // Flag for program to allocate more space in array for more input

int \*temp = resizeArray(arr, size); // Calls resizeArray and returns a pointer to a newly allocated array that is one size larger

arr = temp; // Assign temp pointer to pointer in main

size++; // Increment counter/flag to keep track of the size

}

arr[i] = input; // Assign input to current element in array

}

std::cout << "Mean called: " << mean(arr, size) << std::endl; // Output array mean using array subscript notation

std::cout << "Mean2 called: " << mean2(arr, size) << std::endl; // Output array mean using pointer arithmetic notation

delete [] arr; // delete the dynamically allocated array

return 0;

}

double mean(int \*arr, int size) { // Passed in: array and its size

double total = 0;

for (int i = 0; i < size; i++) { // Adds up all of the array elements

total += arr[i];

}

return total / size; // returns the mean

}

double mean2(int \*arr, int size) { // Passed in: array using pointer and its size

double total = 0;

for (int i = 0; i < size; i++) { // Adds up all of the array elements

total += \*(arr + i);

}

return total / size; // returns the mean

}

**Bonus Sample Output:**

**Sample 1:**

Enter a number: (-1 to exit)

45

Enter a number: (-1 to exit)

122

Enter a number: (-1 to exit)

65

Enter a number: (-1 to exit)

89999

Enter a number: (-1 to exit)

45

Enter a number: (-1 to exit)

-1

Mean called: 18055.2

Mean2 called: 18055.2

**Sample 2:**

Enter a number: (-1 to exit)

5

Enter a number: (-1 to exit)

2

Enter a number: (-1 to exit)

4

Enter a number: (-1 to exit)

7

Enter a number: (-1 to exit)

8

Enter a number: (-1 to exit)

65

Enter a number: (-1 to exit)

40

Enter a number: (-1 to exit)

12

Enter a number: (-1 to exit)

2

Enter a number: (-1 to exit)

3

Enter a number: (-1 to exit)

5

Enter a number: (-1 to exit)

4

Enter a number: (-1 to exit)

7

Enter a number: (-1 to exit)

8

Enter a number: (-1 to exit)

9

Enter a number: (-1 to exit)

1

Enter a number: (-1 to exit)

10

Enter a number: (-1 to exit)

25

Enter a number: (-1 to exit)

25

Enter a number: (-1 to exit)

45

Enter a number: (-1 to exit)

16

Enter a number: (-1 to exit)

89

Enter a number: (-1 to exit)

78

Enter a number: (-1 to exit)

52

Enter a number: (-1 to exit)

5

Enter a number: (-1 to exit)

4

Enter a number: (-1 to exit)

1

Enter a number: (-1 to exit)

2

Enter a number: (-1 to exit)

2

Enter a number: (-1 to exit)

2

Enter a number: (-1 to exit)

2

Enter a number: (-1 to exit)

4

Enter a number: (-1 to exit)

7

Enter a number: (-1 to exit)

4

Enter a number: (-1 to exit)

5

Enter a number: (-1 to exit)

-1

Mean called: 16

Mean2 called: 16

**Sample 3:**

Enter a number: (-1 to exit)

5

Enter a number: (-1 to exit)

-1

Mean called: 5

Mean2 called: 5

**Sample 4:**

Enter a number: (-1 to exit)

5

Enter a number: (-1 to exit)

2

Enter a number: (-1 to exit)

4

Enter a number: (-1 to exit)

-1

Mean called: 3.66667

Mean2 called: 3.66667