Program No. : Farmer\_Garcia\_Elliot\_H6.cpp

Programmer : Elliot Farmer Garcia

Course/Section : ELET 2300-07/23493

Instructor : Dr. F. Attarzadeh

Date Assigned : 10/31/2019

Date Modified : 11/07/2019

Due Date : 11/12/2019

Compiler : Microsoft Visual Studio Enterprise 2019

Environment : Console Applications

Operating System : Windows 7

**Program Flowchart**



**.cpp source file**

/\*

Program No. : Farmer\_Garcia\_Elliot\_H6.cpp

Programmer : Elliot Farmer Garcia

Course / Section : ELET 2300 - 07 / 23493

Instructor : Dr. F. Attarzadeh

Date Assigned : 10/31/2019

Date Modified : 11/07/2019

Due Date : 11/12/2019

Compiler : Microsoft Visual Studio Enterprise 2019

Environment : Console Applications

Operating System : Windows 7

\*/

/\*

Problem Statement

This is a menu-driven program which finds the smallest number in a

user-specified range of randomly generated elements, in an array of

user-specified size, up to 12 elements.

Output is aligned to the default Windows 7 terminal size, 80x25.

\*/

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

void help(void);

void smallest(float s[], int size);

int sizeOfArray(void);

float findSmallest(float s[], int myS);

int frequency(float s[], float smallestNo, int myS);

void display(float s[], float smallestNo, int freq, int myS);

//main() is responsible for printing the menu, acquiring an operation

//selection from the user, calling the appropriate operation function,

//and error checking.

int main(void) {

system("cls"); //clears terminal screen

//variable set by user to call the desired operation

char op;

//array initialization

const int size = 12;

float s[size];

//strings for menu handling

string menu = "\n\t\t\tHelp\t\tSmallest\tQuit\n\n";

string prompt = "Please enter the initial of your desired operation: ";

string inv = "Invalid selection.\n";

//seed pseudo-random number generator

srand(static\_cast <unsigned> (time(0)));

//loop runs until 'q' or 'Q' is selected

while (true) {

//prints menu and asks user for menu selection

cout << menu;

cout << prompt;

cin >> op;

cout << "\n";

//menu selection is evaluated

switch (op) {

//help has been selected

case 'h':

case 'H':

help();

break;

//smallest() has been selected

case 's':

case 'S':

smallest(s, size);

break;

//quit program has been selected

case 'q':

case 'Q':

system("cls");

return 0; //exits program

default:

cout << inv;

}

}

}

//prints a brief explanation of the program

void help(void) {

string help = "This program will find the smallest number in a randomly "

"generated array of no\ngreater than 12 floats. Entering 'S' or 's' "

"begins this process."

"\n\nThe program will ask the user for the size of the array, then "

"the minimum and\nmaximum values of the elements in the array. It "

"will then find the smallest\nvalue in array, and how frequently "

"that value occurs.\n\nTo continue, despite what the program tells "

"the user, the user must enter a\ncharacter into the program."

"\n\nEntering 'Q' or 'q' from the main menu will exit the program."

"\n\nThe user would do well to memorize this help menu, because "

"it will be cleared\nwhen the user presses a key.\n\n";

cout << help;

system("pause"); //waits for user to press a key before continuing

system("cls");

return;

}

//smallest() is the principal function of the program. It acquires an array

//size from the user and error-checks it, acquires minimum and maximum element

//values for the array, and finds the smallest array element and its

//frequency, mostly by calling the relevant helper function.

void smallest(float s[], int size) {

//get array size

int myS = sizeOfArray();

//error-checking

while ((myS <= 0) || (myS > size)) {

cout << "Invalid selection - array size must be > 0 and <= " << size

<< ".\n";

myS = sizeOfArray();

}

//element value minimum and maximum

float min, max;

cout << "Please enter minimum value for elements in array s[]: ";

cin >> min;

cout << "Please enter maximum value for elements in array s[]: ";

cin >> max;

//error-checking

while (max < min) {

cout << "Invalid selection - maximum cannot be smaller than minimum."

<< "\nPlease enter maximum value for elements in array s[]: ";

cin >> max;

}

//randomly generate values for elements up to s[myS]

for (int i = 0; i < myS; i++)

s[i] = min + (rand() / (RAND\_MAX / (max - min)));

//helper functions to analyze and print results

float smallestNo = findSmallest(s, myS);

int freq = frequency(s, smallestNo, myS);

display(s, smallestNo, freq, myS);

return;

}

//Acquires the size of the array from the user. Its value is error-checked

//by smallest() to ensure it is greater than 0 and no greater than size

//(const int defined in main()).

int sizeOfArray(void) {

int i;

cout << "Please enter integer value for array size: ";

cin >> i;

return i;

}

//finds the smallest value in s[] up to s[myS]

float findSmallest(float s[], int myS) {

//assume first element is smallestNo

float smallestNo = s[0];

//if any other element is smaller, it becomes smallestNo

for (int i = 1; i < myS; i++)

if (s[i] < smallestNo)

smallestNo = s[i];

return smallestNo;

}

//determines how frequently smallestNo appears up to s[myS]

int frequency(float s[], float smallestNo, int myS) {

//assumes smallestNo appears 0 times

int freq = 0;

//increment freq each time smallestNo == s[i]

for (int i = 0; i < myS; i++)

if (smallestNo == s[i])

freq++;

return freq;

}

//Prints array s[] up to s[myS], and the results of findSmallest() and

//frequency()

void display(float s[], float smallestNo, int freq, int myS) {

//prints specified array

cout << "\n";

for (int i = 0; i < myS; i++)

cout << fixed << setprecision(2) << "s[" << i << "] = " << s[i] << "\n";

//prints smallestNo and freq

cout << fixed << setprecision(2) << "Smallest no. = " << smallestNo <<

"\t\tFrequency: " << freq

<< "\n\n\t\t\t\t\t\tStrike any key to continue...";

//char used solely to pseudo-pause program

char c;

cin >> c;

system("cls");

return;

}

**Sample Runs**

Help Smallest Quit

Please enter the initial of your desired operation: s

Please enter integer value for array size: 4

Please enter minimum value for elements in array s[]: 0

Please enter maximum value for elements in array s[]: 100

s[0] = 41.45

s[1] = 9.98

s[2] = 93.93

s[3] = 96.04

Smallest no. = 9.98 Frequency: 1

Strike any key to continue...

Help Smallest Quit

Please enter the initial of your desired operation: s

Please enter integer value for array size: 12

Please enter minimum value for elements in array s[]: -1000

Please enter maximum value for elements in array s[]: 1000

s[0] = -798.27

s[1] = -794.79

s[2] = -552.72

s[3] = 587.57

s[4] = 695.18

s[5] = -596.97

s[6] = 338.72

s[7] = 899.72

s[8] = 475.20

s[9] = 886.23

s[10] = 501.63

s[11] = -671.86

Smallest no. = -798.27 Frequency: 1

Strike any key to continue...