Consolidated Assignment 5 Report

This report contains the graded results for the newest of each exercise submitted to the assignment checker prior to 2/16/2022 12:05:59 AM PST.

Student Name: Phillip Ward
Student ID: U09339367
Contact email: phillip.ward@seagate.com
C/C++ Programming I (Section 162461)

Submitted:
- Exercise 0: 2/12/2022 10:10:41 AM PST
- Exercise 1: 2/12/2022 10:42:36 AM PST
- Exercise 2: 2/12/2022 10:58:45 AM PST
- Exercise 3: 2/12/2022 11:25:39 AM PST

Credit to be deducted for uncorrected assignment checker issue(s):
- Exercise 1: 0.9 points (~15%) minimum
- Exercise 2: 0.6 points (~15%) minimum

Score (out of 20 possible): 14.9

THIS WAS SENT FROM A NOTIFICATION-ONLY ADDRESS THAT CANNOT ACCEPT INCOMING MAIL.
For help please contact the instructor at the email address provided on the "Home" page of the course's Canvas website. The assignment checker DOES NOT GRADE your submissions but merely reports on issues so you can avoid credit loss by making corrections and resubmitting. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise that was submitted BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

From: Phillip Ward <mailto:phillip.ward@seagate.com>
Subject: C1A5E0_162461_U09339367
Submitted: 2/12/2022 10:10:41 AM PST
Course: C/C++ Programming I (Section 162461)
Student's name: Phillip Ward
Contact email: phillip.ward@seagate.com
Student ID: U09339367
Assignment 5, Exercise 0 (001474588M01005X28474)
Exercise point value: 6
File submitted:
C1A5E0_Quiz.txt

NOTE: The assignment checker does not check the correctness of answers for this exercise.

Your submission has been accepted and will be graded manually by the instructor. You may resubmit it as many times as you wish BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

-4

Phillip Ward U09339367
Phillip.Ward@seagate.com
C/C++ Programming I
162461 Ray Mitchell
02/12/2022
C1A5E0_Quiz.txt
Quiz Answers

1. E <---C
2. B
3. D <---B
4. B <---E
5. D
6. E <---C

THIS WAS SENT FROM A NOTIFICATION-ONLY ADDRESS THAT CANNOT ACCEPT INCOMING MAIL.
For help please contact the instructor at the email address provided on the "Home" page of the course's Canvas website. The assignment checker DOES NOT GRADE your submissions but merely reports on issues so you can avoid credit loss by making corrections and resubmitting. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise that was submitted BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

From: Phillip Ward <mailto:phillip.ward@seagate.com>
Subject: C1A5E1_162461_U09339367
Submitted: 2/12/2022 10:42:36 AM PST
Course: C/C++ Programming I (Section 162461)
Student's name: Phillip Ward
Contact email: phillip.ward@seagate.com
Student ID: U09339367
Assignment 5, Exercise 1 (001280918M01005X34280)
Exercise point value: 6
File submitted:
C1A5E1_main.c

"Static analysis" results:
1 advisory as follows:
1 miscellaneous advisory (custom validator);
"Runtime" results:
Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

STANDARD GRADING POLICY:
The MINIMUM deduction is the greater of the following for static analysis issues plus a possible additional deduction for runtime issues, if any:
100% if any "goto" statement is used, else
~45% if any compiler or linker error, else
~25% if any warning, else
~15% if any advisory, else
0% if any recommendation.

-0.9

C1A5E1: YOUR MINIMUM DEDUCTION: 0.9 points (~15%) To avoid deductions please correct this exercise and resubmit to the assignment checker before the assignment deadline.

The custom validator found 1 problem.
(http://www.MeanOldTeacher.com/AssignmentCheckerKnownIssues.pdf)

??????????
C1A5E1_main.c(...) advisory A212: The parentheses indicated by the ^ markers are unnecessary:
*** NOTE *** If you are viewing this as email text and the markers are misaligned, change your viewer's font to "Courier New" or a similar "monospace" font.
19. #define CHOICES ((BEST) - (WORST) + 1)
^ ^ ^ ^
*** EXPLANATION ***
Parentheses should typically only be used in the following situations:
1. As required by programming language syntax.
2. To alter default precedence.
3. To clarify complex expressions.
4. Around multi-token macro replacement lists.
5. Around each parameter instance in a macro's replacement list.
Other usages are often more cluttering than helpful and should be avoided.

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 02/07/2022
7  // C1A5E1_main.c
8  // Win10
9  // g++ 11.2.0
10 //
11 // A program that gets ratings from a number of shoppers
12 // and prints the aggregate results.
13 //
14 #include <stdio.h>
15
16 #define SHOPPERS 3
17 #define BEST (-3)
18 #define WORST (-3)
19 #define CHOICES ((BEST) - (WORST) + 1)
20 #define MAX_ERRORS 3
21
22 int main(void) {
23     //declare and initialize variables
24     int ratings[CHOICES] = {0};
25     int shopper = 0;
26     int errors = 0;
27     int rating;
28
29     //Print the rules to the game
30     printf("Allowed ratings are between %i and %i (max errors allowed is %i)\n"
31            , WORST, BEST, MAX_ERRORS);
32
33     //Loop through shopper inputs until we've hit max errors
34     //or we got through all shoppers
35     do
36     {
37         //Get input and do error checking
38         printf("Input Rating: ");
39         scanf("%i", &rating);
40         if (rating < WORST || rating > BEST)//input out of bounds
41         {
42             printf("Input Out of Bounds\n");
43             errors++;
44         }
45         else//good value, log it and reset errors
46         {
47             ratings[rating - WORST]++;
48             errors = 0;
49             //increment to the next shopper
50             shopper++;
51         }
52     } while (shopper < SHOPPERS && errors < MAX_ERRORS);
53     int ratingVal = BEST;
54     for (int ratingIndex = CHOICES - 1; ratingIndex >= 0; ratingIndex--)
55     {
56         printf("Rating      Quantity\n");
57         printf("------      --------\n");
58         printf("%6i      %8i\n", ratingVal, ratings[ratingIndex]);
59         ratingVal--;
60     }
61
```

```
62      return(0);
63  }
```

```
********** C1 ASSIGNMENT 5 EXERCISE 1 AUTOMATIC PROGRAM RUN RESULTS **********

************* THE RESULTS BELOW HAVE BEEN PARTIALLY CHECKED AND **************
*************   NO ERRORS WERE FOUND.  HOWEVER, THIS DOES NOT   **************
*************  NECESSARILY MEAN THAT THERE ARE NO ERRORS.  THE  **************
*************  INSTRUCTOR WILL DO A MORE THOROUGH CHECK DURING  **************
*************                   MANUAL GRADING.                  **************


-------------------------- CODE CHANGES FOR 1ST RUN -------------------------
SHOPPERS = 3  MAX_ERRORS = 3
WORST = -3  BEST = -3
------------------------------ START OF 1ST RUN ------------------------------

Allowed ratings are between -3 and -3 (max errors allowed is 3)
Input Rating: -3
Input Rating: -4
Input Out of Bounds
Input Rating: -3
Input Rating: -2
Input Out of Bounds
Input Rating: 0
Input Out of Bounds
Input Rating: -3
Rating      Quantity
------      --------
    -3             3

------------------------------- END OF 1ST RUN -------------------------------

-------------------------- CODE CHANGES FOR 2ND RUN -------------------------
SHOPPERS = 10  MAX_ERRORS = 2
WORST = 0  BEST = 3
------------------------------ START OF 2ND RUN ------------------------------

Allowed ratings are between 0 and 3 (max errors allowed is 2)
Input Rating: 0
Input Rating: 1
Input Rating: 1
Input Rating: 2
Input Rating: -1
Input Out of Bounds
Input Rating: 2
Input Rating: 2
Input Rating: -1
Input Out of Bounds
Input Rating: -2
Input Out of Bounds
Rating      Quantity
------      --------
     3             0
Rating      Quantity
------      --------
     2             3
Rating      Quantity
------      --------
     1             2
Rating      Quantity
------      --------
     0             1
```

```
------------------------------ END OF 2ND RUN ------------------------------

-------------------------- CODE CHANGES FOR 3RD RUN -------------------------
SHOPPERS = 1  MAX_ERRORS = 1
WORST = 1  BEST = 1
------------------------------ START OF 3RD RUN ------------------------------

Allowed ratings are between 1 and 1 (max errors allowed is 1)
Input Rating: 1
Rating      Quantity
------      --------
     1             1

------------------------------ END OF 3RD RUN ------------------------------

-------------------------- CODE CHANGES FOR 4TH RUN -------------------------
SHOPPERS = 1  MAX_ERRORS = 1
WORST = 1  BEST = 1
------------------------------ START OF 4TH RUN ------------------------------

Allowed ratings are between 1 and 1 (max errors allowed is 1)
Input Rating: 0
Input Out of Bounds
Rating      Quantity
------      --------
     1             0

------------------------------ END OF 4TH RUN ------------------------------

-------------------------- CODE CHANGES FOR 5TH RUN -------------------------
SHOPPERS = 5  MAX_ERRORS = 4
WORST = -1  BEST = 2
------------------------------ START OF 5TH RUN ------------------------------

Allowed ratings are between -1 and 2 (max errors allowed is 4)
Input Rating: -1
Input Rating: 0
Input Rating: 0
Input Rating: -27
Input Out of Bounds
Input Rating: -27
Input Out of Bounds
Input Rating: -27
Input Out of Bounds
Input Rating: -27
Input Out of Bounds
Rating      Quantity
------      --------
     2             0
Rating      Quantity
------      --------
     1             0
Rating      Quantity
------      --------
     0             2
Rating      Quantity
------      --------
    -1             1
```

```
------------------------------ END OF 5TH RUN ------------------------------

-------------------------- CODE CHANGES FOR 6TH RUN --------------------------
SHOPPERS = 13  MAX_ERRORS = 3
WORST = -10  BEST = -4
------------------------------ START OF 6TH RUN ------------------------------

Allowed ratings are between -10 and -4 (max errors allowed is 3)
Input Rating: -10
Input Rating: -4
Input Rating: -10
Input Rating: -4
Input Rating: -10
Input Rating: -4
Input Rating: -4
Input Rating: -4
Input Rating: -4
Input Rating: -4
Input Rating: -4
Input Rating: -4
Input Rating: -4
Rating      Quantity
------      --------
    -4            10
Rating      Quantity
------      --------
    -5             0
Rating      Quantity
------      --------
    -6             0
Rating      Quantity
------      --------
    -7             0
Rating      Quantity
------      --------
    -8             0
Rating      Quantity
------      --------
    -9             0
Rating      Quantity
------      --------
   -10             3

------------------------------ END OF 6TH RUN ------------------------------
```

THIS WAS SENT FROM A NOTIFICATION-ONLY ADDRESS THAT CANNOT ACCEPT INCOMING MAIL.
For help please contact the instructor at the email address provided on the "Home" page of the course's Canvas website. The assignment checker DOES NOT GRADE your submissions but merely reports on issues so you can avoid credit loss by making corrections and resubmitting. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise that was submitted BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

From: Phillip Ward <mailto:phillip.ward@seagate.com>
Subject: C1A5E2_162461_U09339367
Submitted: 2/12/2022 10:58:45 AM PST
Course: C/C++ Programming I (Section 162461)
Student's name: Phillip Ward
Contact email: phillip.ward@seagate.com
Student ID: U09339367
Assignment 5, Exercise 2 (00180088M01005X20080)
Exercise point value: 4
Files submitted:
C1A5E2_ComputeMaximum.cpp
C1A5E2_ComputeMinimum.cpp
C1A5E2_ComputeMinMax.h
C1A5E2_main.cpp

"Static analysis" results:
1 advisory as follows:
1 miscellaneous advisory (custom validator);
"Runtime" results:
Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

STANDARD GRADING POLICY:
The MINIMUM deduction is the greater of the following for static analysis issues plus a possible additional deduction for runtime issues, if any:
100% if any "goto" statement is used, else
~45% if any compiler or linker error, else
~25% if any warning, else
~15% if any advisory, else
0% if any recommendation. -0.2

C1A5E2: YOUR MINIMUM DEDUCTION: 0.6 points (~15%) To avoid deductions please correct this exercise and resubmit to the assignment checker before the assignment deadline.

The custom validator found 1 problem.
(http://www.MeanOldTeacher.com/AssignmentCheckerKnownIssues.pdf)

??????????
C1A5E2_main.cpp(20) advisory A262: Unnecessary comment: //Declare variables
*** EXPLANATION ***
This comment is probably unnecessary. Some of the comments in my demonstration code are only to provide basic information to beginning programmers, information so elementary that it will become second nature very quickly. Commenting such things in your code merely clutters and should be avoided. Typical examples of obvious and unnecessary comments include describing the purpose or content of standard header files, stating that something is a macro, variable, or function definition/declaration, and stating that simple operations are assignments, additions, multiplications, etc. Instead, comments should describe the operation of your functions, algorithms, code snippets, and other things that might not be immediately obvious to someone seeing them for the first time. If you believe the assignment checker has incorrectly interpreted

```
your use of this comment, please contact the instructor to discuss it.
```

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 02/08/2022
7  // C1A5E2_ComputeMinMax.h
8  // Win10
9  // g++ 11.2.0
10 //
11 // Header file for compute min and max functions
12 //
13 #ifndef C1A5E2_COMPUTEMINMAX_H
14 #define C1A5E2_COMPUTEMINMAX_H
15 double &ComputeMaximum(const double &num1, const double &num2);
16 double &ComputeMinimum(const double &num1, const double &num2);
17 #endif
```

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 02/08/2022
7  // C1A5E2_main.cpp
8  // Win10
9  // g++ 11.2.0
10 //
11 // A program that returns the max and min between two values
12 //
13
14 #include <iostream>
15 using namespace std;
16
17 #include "C1A5E2_ComputeMinMax.h"
18
19 int main() {
20     //Declare variables
21     double val1, val2;
22     //Get input
23     cout << "Input two decimal numbers separated by a space: ";
24     cin >> val1 >> val2;
25     //Compute and Output
26     cout << "ComputeMinimum(" << val1 << ", " << val2 << ") returned "
27          << ComputeMinimum(val1, val2) << "\n";
28     cout << "ComputeMaximum(" << val1 << ", " << val2 << ") returned "
29          << ComputeMaximum(val1, val2) << "\n";
30     return(0);
31 }
```

Erroneous/misleading information: No information at all is preferable to erroneous or misleading information. This program returns 0

1 80

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 02/04/2022
7  // C1A5E2_ComputeMaximum.cpp
8  // Win10
9  // g++ 11.2.0
10 //
11 // Contains a function to return the maximum between two values
12 //
13
14 double &ComputeMaximum(const double &num1, const double &num2) {
15     return (double &)((num1 > num2) ? num1 : num2);
16 }
```

1 80

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 02/04/2022
7  // C1A5E2_ComputeMinimum.cpp
8  // Win10
9  // g++ 11.2.0
10 //
11 // Contains a function to return the minimum between two values
12 //
13
14 double &ComputeMinimum(const double &num1, const double &num2) {
15     return (double &)((num1 < num2) ? num1 : num2);
16 }
```

```
********** C1 ASSIGNMENT 5 EXERCISE 2 AUTOMATIC PROGRAM RUN RESULTS **********

************* THE RESULTS BELOW HAVE BEEN PARTIALLY CHECKED AND **************
*************   NO ERRORS WERE FOUND.  HOWEVER, THIS DOES NOT   **************
*************  NECESSARILY MEAN THAT THERE ARE NO ERRORS.  THE  **************
*************  INSTRUCTOR WILL DO A MORE THOROUGH CHECK DURING  **************
*************                  MANUAL GRADING.                    **************


------------------------------ START OF 1ST RUN ------------------------------

Input two decimal numbers separated by a space: 3 3
ComputeMinimum(3, 3) returned 3
ComputeMaximum(3, 3) returned 3


------------------------------- END OF 1ST RUN -------------------------------

------------------------------ START OF 2ND RUN ------------------------------

Input two decimal numbers separated by a space: -7.98 7.13
ComputeMinimum(-7.98, 7.13) returned -7.98
ComputeMaximum(-7.98, 7.13) returned 7.13


------------------------------- END OF 2ND RUN -------------------------------

------------------------------ START OF 3RD RUN ------------------------------

Input two decimal numbers separated by a space: 2000.45 0
ComputeMinimum(2000.45, 0) returned 0
ComputeMaximum(2000.45, 0) returned 2000.45


------------------------------- END OF 3RD RUN -------------------------------

------------------------------ START OF 4TH RUN ------------------------------

Input two decimal numbers separated by a space: 54e-2 86e-1
ComputeMinimum(0.54, 8.6) returned 0.54
ComputeMaximum(0.54, 8.6) returned 8.6


------------------------------- END OF 4TH RUN -------------------------------

------------------------------ START OF 5TH RUN ------------------------------

Input two decimal numbers separated by a space: 86e-1 54e-2
ComputeMinimum(8.6, 0.54) returned 0.54
ComputeMaximum(8.6, 0.54) returned 8.6


------------------------------- END OF 5TH RUN -------------------------------

------------------------------ START OF 6TH RUN ------------------------------

Input two decimal numbers separated by a space: -0 0
ComputeMinimum(-0, 0) returned 0
ComputeMaximum(-0, 0) returned 0
```

```
-------------------------------- END OF 6TH RUN --------------------------------
```

THIS WAS SENT FROM A NOTIFICATION-ONLY ADDRESS THAT CANNOT ACCEPT INCOMING MAIL.
For help please contact the instructor at the email address provided on the "Home" page of the course's Canvas website. The assignment checker DOES NOT GRADE your submissions but merely reports on issues so you can avoid credit loss by making corrections and resubmitting. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise that was submitted BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

From: Phillip Ward <mailto:phillip.ward@seagate.com>
Subject: C1A5E3_162461_U09339367
Submitted: 2/12/2022 11:25:39 AM PST
Course: C/C++ Programming I (Section 162461)
Student's name: Phillip Ward
Contact email: phillip.ward@seagate.com
Student ID: U09339367
Assignment 5, Exercise 3 (001930476M01005X79930)
Exercise point value: 4
Files submitted:
C1A5E3_ComputeMaximum.cpp
C1A5E3_ComputeMinimum.cpp
C1A5E3_ComputeMinMax.h
C1A5E3_main.cpp

"Static analysis" results:
No "static" issues;
"Runtime" results:
Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 02/08/2022
7  // C1A5E3_ComputeMinMax.h
8  // Win10
9  // g++ 11.2.0
10 //
11 // Header file for compute min and max functions
12 //
13 #ifndef C1A5E3_COMPUTEMINMAX_H
14 #define C1A5E3_COMPUTEMINMAX_H
15 double *ComputeMaximum(const double *num1, const double *num2);
16 double *ComputeMinimum(const double *num1, const double *num2);
17 #endif
```

```
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 02/12/2022
// C1A5E3_main.cpp
// Win10
// g++ 11.2.0
//
// Finds the max and min between two input values
//

#include <iostream>
using namespace std;

#include "C1A5E3_ComputeMinMax.h"

int main() {
    double num1, num2;
    //Get input
    cout << "Input two decimal numbers separated by a space: ";
    cin >> num1 >> num2;
    //Print out results
    cout << "ComputeMinimum(&" << num1 << ", &" << num2 << ") returned &"
         << *ComputeMinimum(&num1, &num2) << "\n";
    cout << "ComputeMaximum(&" << num1 << ", &" << num2 << ") returned &"
         << *ComputeMaximum(&num1, &num2) << "\n";
    return(0);
}
```

```
1
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 02/12/2022
// C1A5E3_ComputeMaximum.cpp
// Win10
// g++ 11.2.0
//
// Contains a function to return a pointer to the maximum between two values
//

double *ComputeMaximum(const double *num1, const double *num2) {
    return (double *)((*num1 > *num2) ? num1 : num2);
}
```

1 80

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 02/12/2022
7  // C1A5E3_ComputeMinimum.cpp
8  // Win10
9  // g++ 11.2.0
10 //
11 // Contains a function to return a pointer to the minimum between two values
12 //
13
14 double *ComputeMinimum(const double *num1, const double *num2) {
15     return (double *)((*num1 < *num2) ? num1 : num2);
16 }
```

```
********** C1 ASSIGNMENT 5 EXERCISE 3 AUTOMATIC PROGRAM RUN RESULTS **********

************* THE RESULTS BELOW HAVE BEEN PARTIALLY CHECKED AND **************
*************   NO ERRORS WERE FOUND.  HOWEVER, THIS DOES NOT   **************
*************  NECESSARILY MEAN THAT THERE ARE NO ERRORS.  THE  **************
*************  INSTRUCTOR WILL DO A MORE THOROUGH CHECK DURING  **************
*************                  MANUAL GRADING.                    **************


------------------------------ START OF 1ST RUN ------------------------------

Input two decimal numbers separated by a space: 3 3
ComputeMinimum(&3, &3) returned &3
ComputeMaximum(&3, &3) returned &3


------------------------------- END OF 1ST RUN -------------------------------

------------------------------ START OF 2ND RUN ------------------------------

Input two decimal numbers separated by a space: -7.98 7.13
ComputeMinimum(&-7.98, &7.13) returned &-7.98
ComputeMaximum(&-7.98, &7.13) returned &7.13


------------------------------- END OF 2ND RUN -------------------------------

------------------------------ START OF 3RD RUN ------------------------------

Input two decimal numbers separated by a space: 2000.45 0
ComputeMinimum(&2000.45, &0) returned &0
ComputeMaximum(&2000.45, &0) returned &2000.45


------------------------------- END OF 3RD RUN -------------------------------

------------------------------ START OF 4TH RUN ------------------------------

Input two decimal numbers separated by a space: 54e-2 86e-1
ComputeMinimum(&0.54, &8.6) returned &0.54
ComputeMaximum(&0.54, &8.6) returned &8.6


------------------------------- END OF 4TH RUN -------------------------------

------------------------------ START OF 5TH RUN ------------------------------

Input two decimal numbers separated by a space: 86e-1 54e-2
ComputeMinimum(&8.6, &0.54) returned &0.54
ComputeMaximum(&8.6, &0.54) returned &8.6


------------------------------- END OF 5TH RUN -------------------------------

------------------------------ START OF 6TH RUN ------------------------------

Input two decimal numbers separated by a space: -0 0
ComputeMinimum(&-0, &0) returned &0
ComputeMaximum(&-0, &0) returned &0
```

```
-------------------------------- END OF 6TH RUN --------------------------------
```