Consolidated Assignment 3 Report

This report contains the graded results for the newest of each exercise submitted to the assignment checker prior to 2/2/2022 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: 1/30/2022 12:50:51 PM PST
   Exercise 1: 1/30/2022 8:54:18 PM PST
   Exercise 2: 1/30/2022 8:53:49 PM PST
   Exercise 3: 1/30/2022 8:52:55 PM PST

Credit to be deducted for uncorrected assignment checker issue(s):
   Exercise 1:  0.8 points (~25%) minimum plus a runtime issue
                deduction to be determined.
   Exercise 2:  1.2 points (~25%) minimum plus a runtime issue
                deduction to be determined.
   Exercise 3:  1.5 points (~25%) minimum plus a runtime issue
                deduction to be determined.
```

Score (out of 20 possible): 9.2

```
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: C1A3E0_162461_U09339367
   Submitted: 1/30/2022 12:50:51 PM PST
   Course: C/C++ Programming I (Section 162461)
   Student's name: Phillip Ward
   Contact email: phillip.ward@seagate.com
   Student ID: U09339367
   Assignment 3, Exercise 0  (001270723M01005X24270)
   Exercise point value: 6
   File submitted:
      C1A3E0_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.
```

-3

```
Phillip Ward U09339367
Phillip.Ward@seagate.com
C/C++ Programming I
162461 Ray Mitchell
01/30/2022
C1A3E0_Quiz.txt
Quiz Answers

1. A
2. C <---A
3. B <---A
4. C
5. A
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: C1A3E1_162461_U09339367
   Submitted: 1/30/2022 8:54:18 PM PST
   Course: C/C++ Programming I (Section 162461)
   Student's name: Phillip Ward
   Contact email: phillip.ward@seagate.com
   Student ID: U09339367
   Assignment 3, Exercise 1  (003568264M01005X9568)
   Exercise point value: 3
   File submitted:
      C1A3E1_main.c

"Static analysis" results:
   11 warnings as follows:
      11 poor practice warnings (custom validator);
   15 advisories as follows:
      14 inter-token spacing advisories (custom validator);
       1 miscellaneous advisory (custom validator);
"Runtime" results:
   Program ran - ERRORS WERE DETECTED (SEE ATTACHMENT);

STANDARD GRADING POLICY:
The MINIMUM deduction is the greater of the following for compile-time 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        -0.8
     0% if any recommendation.

C1A3E1: YOUR MINIMUM DEDUCTION: 0.8 points (~25%) plus a runtime issue deduction to be
determined.  To avoid deductions please correct this exercise and resubmit to the
assignment checker before the assignment deadline.


##### The custom validator found 26 problems. #####
(http://www.MeanOldTeacher.com/AssignmentCheckerKnownIssues.pdf)

??????????
C1A3E1_main.c(22-23) warning W657:  Unnecessary consecutive uses of "printf" on lines
22-23
*** EXPLANATION ***
In this exercise there is no reason to use "printf" more than once on lines 22-23.  It
would be more appropriate, more efficient, and less cluttering to combine them into a
single "printf" statement that starts on line 22 and continues onto additional lines if
necessary or desired.  While it is often desireable to put the code for an entire
"printf" (or similar) statement on a single line, you not should use multiple
statements arbitrarily or just because the line is not wide enough for it to fit.  For
example, pretend the following statement is too wide to fit on a single line (dog, cat,
and fox are variables):
   printf("%d %d %d\n", dog, cat, fox);
DO NOT solve the problem by doing:
   printf("%d ", dog);
   printf("%d ", cat);
   printf("%d\n", fox);
Instead, split the original statement onto additional lines in a logical and readable
fashion and indent them to show their association with the first line:
   printf(
      "%d %d %d\n",
      dog, cat, fox);

??????????
C1A3E1_main.c(...) warning W528:  10 meaningless/cryptic identifiers as follows:
   Line 27, column 13:  n
   Line 28, column 20:  n
   Line 28, column 22:  n
   Line 28, column 24:  n
   Line 28, column 26:  n
   Line 28, column 28:  n
   Line 28, column 30:  n
   Line 28, column 32:  n
   Line 29, column 25:  n
   Line 31, column 41:  n
*** EXPLANATION ***
Whenever possible and practical the names used for variables, functions, macros, etc.
should convey the meaning but not the value of what they represent.  For example, to
count executions of a loop's body names like "i", "j", "k", etc. convey absolutely no
meaning, whereas names like "studentNumber" or "responseCount" provide some insight
into their purposes.  Although there are some cases in which less meaningful names may
be appropriate, such as when a variable is being used for multiple unrelated purposes
or when names like "x", "y", "z", etc. are used to represent coordinates or abstract
equation variables, this is usually not the situation for exercises in this course.
However, if you believe that your name choice is appropriate in this case, please
contact the instructor to discuss it.

??????????
C1A3E1_main.c(...) advisory A205:  14 missing spaces as follows:
   Line 28, column 32 (between '*' and 'n')
   Line 28, column 31 (between 'n' and '*')
   Line 28, column 30 (between '*' and 'n')
   Line 28, column 29 (between 'n' and '*')
   Line 28, column 28 (between '*' and 'n')
   Line 28, column 27 (between 'n' and '*')
   Line 28, column 26 (between '*' and 'n')
   Line 28, column 25 (between 'n' and '*')
   Line 28, column 24 (between '*' and 'n')
   Line 28, column 23 (between 'n' and '*')
   Line 28, column 22 (between '*' and 'n')
   Line 28, column 21 (between 'n' and '*')
   Line 29, column 25 (between '*' and 'n')
   Line 29, column 24 (between '7' and '*')

??????????
C1A3E1_main.c(...) advisory A206:  Unwanted blank line as follows:
   Line  18
*** EXPLANATION ***
Although thoughtfully placed blank lines can make code more readable, excessive or
inappropriately placed blank lines only reduce readablity.
```

```
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 01/30/2022
// C1A3E1_main.c
// Win10
// g++ 11.2.0
//
// Could you use scientific notation?
//
```

Title block provides no meaningful information about the purpose/functionality of what is in the file.

It would have its own set of problems.

```
#include <stdio.h>
~~#define LEADER_CHAR '^'~~
~~#define DIAGONAL_CHAR '@'~~
```

?

-0.5

```
int main(void) {

    int input;
    printf("input any positive decimal integer:");
    scanf("%d", &input);
    printf("n       n^7       n^8");
    printf("\n---------------------\n");
    //loop through number values
    for (int lineNum = 0; lineNum <= input; lineNum++)
    {
        int n = lineNum;
        int exp7 = n*n*n*n*n*n*n;
        int exp8 = exp7*n;

        printf("%d       %d      %d\n", n, exp7, exp8);
    }

    return(0);
}
```

-1

```
********** C1 ASSIGNMENT 3 EXERCISE 1 AUTOMATIC PROGRAM RUN RESULTS **********

***************************** ERROR(S) DETECTED ******************************


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

input any positive decimal integer:1
n       n^7       n^8
---------------------
0       0      0
1       1      1


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

-------------------- IMPLEMENTATION SPECIFICS FOR 2ND RUN --------------------
This test uses a 32-bit int, which overflows for values > 14^8 and 15^7.
The results for these values are not tested.
------------------------------ START OF 2ND RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Items in the table are not properly aligned.


input any positive decimal integer:25
n       n^7       n^8
---------------------
0       0      0
1       1      1
2       128      256
3       2187      6561
4       16384      65536
5       78125      390625
6       279936      1679616
7       823543      5764801
8       2097152      16777216
9       4782969      43046721
10       10000000      100000000
11       19487171      214358881
12       35831808      429981696
13       62748517      815730721
14       105413504      1475789056
15       170859375      -1732076671
16       268435456      0
17       410338673      -1614177151
18       612220032      -1864941312
19       893871739      -196306143
20       1280000000      -169803776
21       1801088541      -831846303
22       -1800609408      -958701312
23       -890141849      1001573953
24       291504128      -1593835520
25       1808548329      -2030932031



<<EXPECTED>> (Different user prompt wording is okay.)

Enter a decimal integer value >= zero: 25
 n^1          n^7          n^8
```

```
------------------------------
   0            0            0
   1            1            1
   2          128          256
   3         2187         6561
   4        16384        65536
   5        78125       390625
   6       279936      1679616
   7       823543      5764801
   8      2097152     16777216
   9      4782969     43046721
  10     10000000    100000000
  11     19487171    214358881
  12     35831808    429981696
  13     62748517    815730721
  14    105413504   1475789056
  15    170859375  -1732076671
  16    268435456            0
  17    410338673  -1614177151
  18    612220032  -1864941312
  19    893871739   -196306143
  20   1280000000   -169803776
  21   1801088541   -831846303
  22  -1800609408   -958701312
  23   -890141849   1001573953
  24    291504128  -1593835520
  25   1808548329  -2030932031


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

-------------------- IMPLEMENTATION SPECIFICS FOR 3RD RUN --------------------
This test uses a 32-bit int, which overflows for values > 14^8 and 15^7.
The results for these values are not tested.
------------------------------ START OF 3RD RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Items in the table are not properly aligned.


input any positive decimal integer:36
n       n^7       n^8
---------------------
0       0      0
1       1      1
2       128      256
3       2187      6561
4       16384      65536
5       78125      390625
6       279936      1679616
7       823543      5764801
8       2097152      16777216
9       4782969      43046721
10       10000000      100000000
11       19487171      214358881
12       35831808      429981696
13       62748517      815730721
14       105413504      1475789056
15       170859375      -1732076671
16       268435456      0
17       410338673      -1614177151
```

```
18       612220032      -1864941312
19       893871739      -196306143
20       1280000000      -169803776
21       1801088541      -831846303
22       -1800609408      -958701312
23       -890141849      1001573953
24       291504128      -1593835520
25       1808548329      -2030932031
26       -558124416      -1626332928
27       1870418611      -1038305055
28       608026624      -155123712
29       70007125      2030206625
30       395163520      -1029996288
31       1742810335      -1807454463
32       0      0
33       -331229983      1954312449
34       983742592      -912490240
35       -85212565      1312527521
36       1054752768      -683606016
```

```
<<EXPECTED>> (Different user prompt wording is okay.)

Enter a decimal integer value >= zero: 36
 n^1          n^7          n^8
------------------------------
   0            0            0
   1            1            1
   2          128          256
   3         2187         6561
   4        16384        65536
   5        78125       390625
   6       279936      1679616
   7       823543      5764801
   8      2097152     16777216
   9      4782969     43046721
  10     10000000    100000000
  11     19487171    214358881
  12     35831808    429981696
  13     62748517    815730721
  14    105413504   1475789056
  15    170859375  -1732076671
  16    268435456            0
  17    410338673  -1614177151
  18    612220032  -1864941312
  19    893871739   -196306143
  20   1280000000   -169803776
  21   1801088541   -831846303
  22  -1800609408   -958701312
  23   -890141849   1001573953
  24    291504128  -1593835520
  25   1808548329  -2030932031
  26   -558124416  -1626332928
  27   1870418611  -1038305055
  28    608026624   -155123712
  29     70007125   2030206625
  30    395163520  -1029996288
  31   1742810335  -1807454463
  32            0            0
  33   -331229983   1954312449
  34    983742592   -912490240
  35    -85212565   1312527521
  36   1054752768   -683606016


------------------------------- END OF 3RD 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: C1A3E2_162461_U09339367
   Submitted: 1/30/2022 8:53:49 PM PST
   Course: C/C++ Programming I (Section 162461)
   Student's name: Phillip Ward
   Contact email: phillip.ward@seagate.com
   Student ID: U09339367
   Assignment 3, Exercise 2  (002181933M01005X60181)
   Exercise point value: 5
   File submitted:
      C1A3E2_main.cpp

"Static analysis" results:
    9 warnings as follows:
       2 magic number warnings (custom validator);
       5 poor practice warnings (custom validator);
       2 miscellaneous warnings (custom validator);
    5 advisories as follows:
       5 inter-token spacing advisories (custom validator);
    1 recommendation;
"Runtime" results:
   Program ran - ERRORS WERE DETECTED (SEE ATTACHMENT);

STANDARD GRADING POLICY:
The MINIMUM deduction is the greater of the following for compile-time 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.
```

-1.2

```
C1A3E2: YOUR MINIMUM DEDUCTION: 1.2 points (~25%) plus a runtime issue deduction to be
determined.  To avoid deductions please correct this exercise and resubmit to the
assignment checker before the assignment deadline.


##### The custom validator found 15 problems. #####
(http://www.MeanOldTeacher.com/AssignmentCheckerKnownIssues.pdf)

??????????
C1A3E2_main.cpp(...) warning W807:  Optimal name for "radix/base" variable not found.
*** EXPLANATION ***
The terms "radix" and "base" are used interchangeably when discussing number systems
and both merely mean the number of unique characters a number system uses to represent
values.  For example, the radix of the hexadecimal system is 16.  Because the value of
the radix is needed to perform some of this exercise's calculations, it must be
represented by an appropriately-named "const int" variable to avoid embedding
inappropriate "magic numbers" in the code.  However, even though you might have
attempted to define the radix in this way, an appropriate name for that variable was
```

```
not found.
...continued:
Whenever possible and practical, a name should indicate the fundamental purpose it
serves.  Although students sometimes choose names like DIVIDE_BY, DIVIDER, MULTIPLIER,
SINGLE_DIGIT, LSD, MSD, etc. to represent the radix, its fundamental purpose is still
just to represent the "radix" and nothing more.  For that reason, the most appropriate
name to use is "RADIX" or "BASE", or if you prefer, something else appropriate with
"RADIX" or "BASE" as part of the name.  Of course, by convention the name of a constant
variable should be in uppercase.

??????????
C1A3E2_main.cpp(40) recommendation R150:  No return statement in function "main".
NOTE: There will be no deduction for this but correction is recommended.
*** EXPLANATION ***
The language standards permit the "main" function to be implemented without a return
statement, in which case a value of 0 will be automatically returned if the end of the
function's closing brace is reached.  However, the most accepted programming practice
is to provide an explicit return statement anyway just for consistency with other
functions, all of which require an explicit return statement if they are not declared
to return void.

??????????
C1A3E2_main.cpp(...) warning W549:  Non-conventional case for a non-const variable name
as follows:
   Line 18, column  9:  name: LSD   type: "int"
*** EXPLANATION ***
By convention the alphabetic characters used in the names of "non-const" variables
should not be all uppercase.  This makes it easier to differentiate them from const
variables, which should be all uppercase.

??????????
C1A3E2_main.cpp(23) warning W232:  Statement associated with "if" should be on the
following line.

??????????
C1A3E2_main.cpp(25) warning W232:  Statement associated with "if" should be on the
following line.

??????????
C1A3E2_main.cpp(...) warning W320:  Unnecessarily complex code as follows:
   Line 31: inValue = inValue/16   Use: inValue /= 16
*** EXPLANATION ***
The simplest code is usually the most readable.

??????????
C1A3E2_main.cpp(...) warning W687:  1 overscoped variable as follows:
   Line 18, column  9:  LSD
*** EXPLANATION ***
The scope of an identifier (a name) is defined as the portion of code over which it is
accessible.  The scope of a variable declared inside a block extends from that
declaration to the end of that block, where a "block" is commonly defined as a "curly-
brace enclosed sequence of 0 or more statements".  Good programming practice dictates
that the scopes of non-const variables be as small as possible to prevent their values
from being changed by code that should not change them.  However, because the values of
const variables cannot be changed, if being used in place of macros they should be
declared in the same place the macros would have been defined.  Otherwise they should
be declared first in the function or block that uses them.  For more details please see
the file named "LimitingTheScopeOfVariables.pdf", which is attached to this email.

??????????
```

```
C1A3E2_main.cpp(...) warning W350:  2 inappropriate "Magic Numbers" as follows:
   Line 30, column 25:  16
   Line 31, column 27:  16
*** EXPLANATION ***
The term "magic number" most commonly refers to a number embedded in code or comment
but can also refer to an embedded character literal or string literal, or to an
identifier named for the value it represents rather than its purpose.  For more details
please see the file named "AvoidingInappropriateMagicNumbers.pdf", which is attached to
this email.  NOTE: If you are getting this warning about literal values in an array
initializer list and using indentifiers to represent them is not appropriate, try
declaring the array constant.  If this does not fix the problem or if doing so causes a
compiler error, please contact the instructor to discuss it.

??????????
C1A3E2_main.cpp(...) warning W301:  The body of the following function contains
insufficiently commented code:
   Line 16, function "main"
*** EXPLANATION ***
This warning is issued when the code WITHIN THE BODY of a function is not commented
sufficiently.  This requirement cannot be met by comments placed in a file's title
block because these should only describe the file's general contents, nor can it be met
by comments placed just before a function's definition because these should only
describe the function's parameters, return value, and general operation.  Instead,
comments related to the details of the code itself should be placed just before or to
the right of that code as appropriate.  For more details please see the file named
"HowAndWhatToComment.pdf", which is attached to this email.

??????????
C1A3E2_main.cpp(...) advisory A205:  4 missing spaces as follows:
   Line 23, column 16 (between ')' and '{')
   Line 25, column 15 (between ')' and '{')
   Line 31, column 27 (between '/' and '1')
   Line 31, column 26 (between 'e' and '/')

??????????
C1A3E2_main.cpp(...) advisory A204:  1 unwanted space as follows:
   Line 23, column  9 (between '(' and 'i')
       //
       // Phillip Ward U09339367
       // Phillip.Ward@seagate.com
       // C/C++ Programming I
       // 162461 Ray Mitchell
       // 01/30/2022
       // C1A3E2_main.cpp
       // Win10
       // g++ 11.2.0
       //
       // A program that prints the reverse of a hex value

       #include <iostream>
       using namespace std;

       int main() {
           int inValue;
           int LSD;

           cout << "Enter any hexadecimal integer value: ";
           cin >> hex >> inValue;
           bool isNeg = inValue < 0;
           if ( isNeg){inValue = inValue * -1;}
           cout << "\"";
           if (isNeg){cout << "-";}
           cout << hex << inValue << "\" in reverse is \"";

           do//iterate through LSD's until none are left
           {
               LSD = inValue % 16;
               inValue = inValue/16;
               cout << LSD;
           } while (inValue > 0);

           if (isNeg)
           {
               cout << "-";
           }
           cout << "\"";
       }
```

-0.8

```
********** C1 ASSIGNMENT 3 EXERCISE 2 AUTOMATIC PROGRAM RUN RESULTS **********

**************************** ERROR(S) DETECTED ******************************


------------------------------ START OF 1ST RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: abcd
"abcd" in reverse is "dcba"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

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

------------------------------ START OF 2ND RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: -2645
"-2645" in reverse is "5462-"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

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

------------------------------ START OF 3RD RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: 100
"100" in reverse is "001"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

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

------------------------------ START OF 4TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: fedcba
"fedcba" in reverse is "abcdef"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

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

------------------------------ START OF 5TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: -002A
"-2a" in reverse is "a2-"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

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

------------------------------ START OF 6TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: 000
"0" in reverse is "0"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

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

------------------------------ START OF 7TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: 10C0
"10c0" in reverse is "0c01"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

------------------------------- END OF 7TH RUN -------------------------------

------------------------------ START OF 8TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: -1010
"-1010" in reverse is "0101-"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

------------------------------- END OF 8TH RUN -------------------------------

------------------------------ START OF 9TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: -0007000
"-7000" in reverse is "0007-"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>
```

```
------------------------------- END OF 9TH RUN -------------------------------

------------------------------ START OF 10TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   Except for lines (if any) that prompt the user for input, a newline
   character must be output at the end of all other lines.  However, that did
   not occur at the end of the line indicated below.


Enter any hexadecimal integer value: F
"f" in reverse is "f"
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>

------------------------------- END OF 10TH 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: C1A3E3_162461_U09339367
   Submitted: 1/30/2022 8:52:55 PM PST
   Course: C/C++ Programming I (Section 162461)
   Student's name: Phillip Ward
   Contact email: phillip.ward@seagate.com
   Student ID: U09339367
   Assignment 3, Exercise 3  (002896518M01005X51896)
   Exercise point value: 6
   File submitted:
      C1A3E3_main.cpp

"Static analysis" results:
   15 warnings as follows:
       5 magic number warnings (custom validator);
       7 poor practice warnings (custom validator);
       3 miscellaneous warnings (custom validator);
   18 advisories as follows:
      16 inter-token spacing advisories (custom validator);
       2 miscellaneous advisories (custom validator);
    1 recommendation;
"Runtime" results:
   Program ran - ERRORS WERE DETECTED (SEE ATTACHMENT);

STANDARD GRADING POLICY:
The MINIMUM deduction is the greater of the following for compile-time 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.
```

-1.5

```
C1A3E3: YOUR MINIMUM DEDUCTION: 1.5 points (~25%) plus a runtime issue deduction to be
determined.  To avoid deductions please correct this exercise and resubmit to the
assignment checker before the assignment deadline.


##### The custom validator found 34 problems. #####
(http://www.MeanOldTeacher.com/AssignmentCheckerKnownIssues.pdf)

??????????
C1A3E3_main.cpp(...) warning W807:  Optimal name for "radix/base" variable not found.
*** EXPLANATION ***
The terms "radix" and "base" are used interchangeably when discussing number systems
and both merely mean the number of unique characters a number system uses to represent
values.  For example, the radix of the octal system is 8.  Because the value of the
radix is needed to perform some of this exercise's calculations, it must be represented
by an appropriately-named "const int" variable to avoid embedding inappropriate "magic
numbers" in the code.  However, even though you might have attempted to define the
```

```
radix in this way, an appropriate name for that variable was not found.
...continued:
Whenever possible and practical, a name should indicate the fundamental purpose it
serves.  Although students sometimes choose names like DIVIDE_BY, DIVIDER, MULTIPLIER,
SINGLE_DIGIT, LSD, MSD, etc. to represent the radix, its fundamental purpose is still
just to represent the "radix" and nothing more.  For that reason, the most appropriate
name to use is "RADIX" or "BASE", or if you prefer, something else appropriate with
"RADIX" or "BASE" as part of the name.  Of course, by convention the name of a constant
variable should be in uppercase.

??????????
C1A3E3_main.cpp(79) recommendation R150:  No return statement in function "main".
NOTE: There will be no deduction for this but correction is recommended.
*** EXPLANATION ***
The language standards permit the "main" function to be implemented without a return
statement, in which case a value of 0 will be automatically returned if the end of the
function's closing brace is reached.  However, the most accepted programming practice
is to provide an explicit return statement anyway just for consistency with other
functions, all of which require an explicit return statement if they are not declared
to return void.

??????????
C1A3E3_main.cpp(...) warning W549:  Non-conventional case for a non-const variable name
as follows:
   Line 40, column 13:  name: MSD   type: "int"
*** EXPLANATION ***
By convention the alphabetic characters used in the names of "non-const" variables
should not be all uppercase.  This makes it easier to differentiate them from const
variables, which should be all uppercase.

??????????
C1A3E3_main.cpp(...) warning W359:  Inappropriate magic number as follows:
   Line 69, column 36:  7
*** EXPLANATION ***
Although it is sometimes appropriate to place magic numbers in string literals (such as
for printf and scanf field width specifications), doing so for any other purpose in
this exercise is not appropriate.

??????????
C1A3E3_main.cpp(23) warning W232:  Statement associated with "if" should be on the
following line.

??????????
C1A3E3_main.cpp(25) warning W232:  Statement associated with "if" should be on the
following line.

??????????
C1A3E3_main.cpp(36) warning W232:  Statement associated with "if" should be on the
following line.

??????????
C1A3E3_main.cpp(...) warning W320:  Unnecessarily complex code as follows:
   Line 32: divisor = divisor*8   Use: divisor *= 8
   Line 33: dividend = dividend/8   Use: dividend /= 8
   Line 73: dividend = dividend - (MSD*divisor)   Use: dividend -= MSD*divisor
   Line 74: divisor = divisor/8   Use: divisor /= 8
*** EXPLANATION ***
The simplest code is usually the most readable.

??????????
```

```
C1A3E3_main.cpp(...) warning W350:  4 inappropriate "Magic Numbers" as follows:
   Line 30, column 33:  7
   Line 32, column 27:  8
   Line 33, column 29:  8
   Line 74, column 27:  8
*** EXPLANATION ***
The term "magic number" most commonly refers to a number embedded in code or comment
but can also refer to an embedded character literal or string literal, or to an
identifier named for the value it represents rather than its purpose.  For more details
please see the file named "AvoidingInappropriateMagicNumbers.pdf", which is attached to
this email.  NOTE: If you are getting this warning about literal values in an array
initializer list and using indentifiers to represent them is not appropriate, try
declaring the array constant.  If this does not fix the problem or if doing so causes a
compiler error, please contact the instructor to discuss it.

??????????
C1A3E3_main.cpp(...) warning W301:  The body of the following function contains
insufficiently commented code:
   Line 16, function "main"
*** EXPLANATION ***
This warning is issued when the code WITHIN THE BODY of a function is not commented
sufficiently.  This requirement cannot be met by comments placed in a file's title
block because these should only describe the file's general contents, nor can it be met
by comments placed just before a function's definition because these should only
describe the function's parameters, return value, and general operation.  Instead,
comments related to the details of the code itself should be placed just before or to
the right of that code as appropriate.  For more details please see the file named
"HowAndWhatToComment.pdf", which is attached to this email.

??????????
C1A3E3_main.cpp(...) advisory A205:  15 missing spaces as follows:
   Line 23, column 16 (between ')' and '{')
   Line 25, column 15 (between ')' and '{')
   Line 30, column  8 (between 'r' and '(')
   Line 32, column 27 (between '*' and '8')
   Line 32, column 26 (between 'r' and '*')
   Line 33, column 29 (between '/' and '8')
   Line 33, column 28 (between 'd' and '/')
   Line 36, column 14 (between ')' and '{')
   Line 36, column  7 (between 'f' and '(')
   Line 40, column 28 (between '/' and 'd')
   Line 40, column 27 (between 'd' and '/')
   Line 73, column 36 (between '*' and 'd')
   Line 73, column 35 (between 'D' and '*')
   Line 74, column 27 (between '/' and '8')
   Line 74, column 26 (between 'r' and '/')

??????????
C1A3E3_main.cpp(...) advisory A204:  1 unwanted space as follows:
   Line 23, column  9 (between '(' and 'i')

??????????
C1A3E3_main.cpp(...) advisory A206:  Unwanted blank lines as follows:
   Line  77
   Line  78
*** EXPLANATION ***
Although thoughtfully placed blank lines can make code more readable, excessive or
inappropriately placed blank lines only reduce readablity.
```

```
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 01/30/2022
// C1A3E3_main.cpp
// Win10
// g++ 11.2.0
//
// A program that prints the word equivalent of an octal value

#include <iostream>
using namespace std;

int main() {
    int inValue;

    //get input and print first part of formatted text
    cout << "Enter any octal integer value: ";
    cin >> oct >> inValue;
    bool isNeg = inValue < 0;
    if ( isNeg){inValue = inValue * -1;}
    cout << "\"";
    if (isNeg){cout << "-";}
    cout << oct << inValue << "\" in words is \"";

    int dividend = inValue;
    int divisor;
    for(divisor = 1; dividend > 7; divisor++)
    {
        divisor = divisor*8;
        dividend = dividend/8;
    }
    //don't forget the minus sign
    if(isNeg){cout << "minus ";}
    dividend = inValue;
    do//find the word equivalent of each MSD
    {
        int MSD = dividend/divisor;
        cout << dividend << "/" << divisor;
        switch (MSD)
        {
        case 0:
            cout << "zero";
            break;
        case 1:
            cout << "one";
            break;
        case 2:
            cout << "two";
            break;
        case 3:
            cout << "three";
            break;
        case 4:
            cout << "four";
            break;
        case 5:
            cout << "five";
            break;
        case 6:
            cout << "six";
            break;
        case 7:
            cout << "seven";
            break;
        default:
            cout << "MSD was not 0-7";
            break;
        }
        cout << " ";
        dividend = dividend - (MSD*divisor);
        divisor = divisor/8;
    } while (divisor != 0);
    cout << "\"";


}
```

-2

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

***************************** ERROR(S) DETECTED ******************************


------------------------------ START OF 1ST RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "3657/1111three 324/111two
      102/11seven 3/1three "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.


Enter any octal integer value: 3657
"3657" in words is "3657/1111three 324/111two 102/11seven 3/1three "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: 3657
"3657" in words is "three six five seven"


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

------------------------------ START OF 2ND RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "minus 2645/1111two 423/111three
      70/11six 2/1two "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.


Enter any octal integer value: -2645
"-2645" in words is "minus 2645/1111two 423/111three 70/11six 2/1two "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: -2645
"-2645" in words is "minus two six four five"


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

------------------------------ START OF 3RD RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "100/111zero 100/11seven 1/1one "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.
```

```
Enter any octal integer value: 100
"100" in words is "100/111zero 100/11seven 1/1one "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: 100
"100" in words is "one zero zero"


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

------------------------------ START OF 4TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "120/111one 7/11zero 7/1seven "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.


Enter any octal integer value: 000120
"120" in words is "120/111one 7/11zero 7/1seven "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: 000120
"120" in words is "one two zero"


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

------------------------------ START OF 5TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "minus 23/11two 1/1one "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.


Enter any octal integer value: -0023
"-23" in words is "minus 23/11two 1/1one "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: -0023
"-23" in words is "minus two three"


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

------------------------------ START OF 6TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "0/1zero "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.


Enter any octal integer value: 000
"0" in words is "0/1zero "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: 000
"0" in words is "zero"


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

------------------------------ START OF 7TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "1010/1111zero 1010/111seven 11/11one
      0/1zero "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.


Enter any octal integer value: 1010
"1010" in words is "1010/1111zero 1010/111seven 11/11one 0/1zero "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: 1010
"1010" in words is "one zero one zero"


------------------------------- END OF 7TH RUN -------------------------------

------------------------------ START OF 8TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "minus 1010/1111zero 1010/111seven
      11/11one 0/1zero "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.


Enter any octal integer value: -1010
"-1010" in words is "minus 1010/1111zero 1010/111seven 11/11one 0/1zero "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: -1010
```

```
"-1010" in words is "minus one zero one zero"


------------------------------- END OF 8TH RUN -------------------------------

------------------------------ START OF 9TH RUN ------------------------------
THIS RUN FAILED BECAUSE:
   1. Results were not as expected.
   2. Unwanted space before close quote: "3/1three "
   3. Except for lines (if any) that prompt the user for input, a newline
      character must be output at the end of all other lines.  However, that
      did not occur at the end of the line indicated below.


Enter any octal integer value: 3
"3" in words is "3/1three "
        <<<<< MISSING NEWLINE CHARACTER AT END OF PREVIOUS LINE >>>>>


<<EXPECTED>> (Different user prompt wording is okay.)

Enter an octal integer value: 3
"3" in words is "three"


------------------------------- END OF 9TH RUN -------------------------------
```