Consolidated Assignment 6 Report

This report contains the graded results for the newest of each exercise submitted to the assignment checker prior to 2/23/2022 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: 2/20/2022 5:20:08 PM PST
   Exercise 1: 2/14/2022 12:22:35 PM PST
   Exercise 2: 2/16/2022 7:44:25 AM PST
   Exercise 3: 2/20/2022 6:51:57 PM PST
```

Score (out of 20 possible): 14

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: C1A6E0_162461_U09339367
   Submitted: 2/20/2022 5:20:08 PM PST
   Course: C/C++ Programming I (Section 162461)
   Student's name: Phillip Ward
   Contact email: phillip.ward@seagate.com
   Student ID: U09339367
   Assignment 6, Exercise 0  (002529261M01005X64529)
   Exercise point value: 6
   File submitted:
      C1A6E0_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.

-6

```
Phillip Ward U09339367
Phillip.Ward@seagate.com
C/C++ Programming I
162461 Ray Mitchell
02/20/2022
C1A6E0_Quiz.txt
Quiz Answers

1. C <---A
2. E <---B
3. A <---D
4. D <---E
5. C <---A
6. B <---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: C1A6E1_162461_U09339367
   Submitted: 2/14/2022 12:22:35 PM PST
   Course: C/C++ Programming I (Section 162461)
   Student's name: Phillip Ward
   Contact email: phillip.ward@seagate.com
   Student ID: U09339367
   Assignment 6, Exercise 1  (001425723M01005X3425)
   Exercise point value: 4
   Files submitted:
      C1A6E1_main.c
      C1A6E1_MyStrlen.c

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

```
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 02/12/2022
// C1A6E1_main.c
// Win10
// g++ 11.2.0
//
// A program that computes the length of an input string
//
#include <stdio.h>
#include <string.h>
#define BUF_SIZE 100
size_t MyStrlen(const char *s1);

int main(void) {
    char buffer[BUF_SIZE];
    //get input
    printf("Input a string: ");
    fgets(buffer, BUF_SIZE, stdin);
    //replace newline with null terminator
    buffer[strcspn(buffer, "\n")] = '\0';
    //compute and print results
    printf("strlen(\"%s\") returned %zu\n", buffer, strlen(buffer));
    printf("MyStrlen(\"%s\") returned %zu\n", buffer, MyStrlen(buffer));
    return(0);
}
```

```
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 02/14/2022
// C1A6E1_MyStrlen.c
// Win10
// g++ 11.2.0
//
// A file containing the function to compute an input string length
//
#include <stddef.h>
size_t MyStrlen(const char *s1)
{
    //store the starting val of the input pointer
    const char * const START = s1;
    //increment the pointer until we find the null terminator
    while (*s1 != '\0')
    {
        s1++;
    }
    return (size_t)(s1 - START);//return the length of the string
}
```

```
********** C1 ASSIGNMENT 6 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.                  **************


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

Input a string: a
strlen("a") returned 1
MyStrlen("a") returned 1

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

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

Input a string: HELLO
strlen("HELLO") returned 5
MyStrlen("HELLO") returned 5

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

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

Input a string: Programming is fun
strlen("Programming is fun") returned 18
MyStrlen("Programming is fun") returned 18

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

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

Input a string:
strlen("") returned 0
MyStrlen("") returned 0

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

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

Input a string: &=#~!
strlen("&=#~!") returned 5
MyStrlen("&=#~!") returned 5

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

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

Input a string: aaaaaaaaaaaaaaa
strlen("aaaaaaaaaaaaaaa") returned 15
MyStrlen("aaaaaaaaaaaaaaa") returned 15

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

------------------------------ START OF 7TH RUN ------------------------------

Input a string: The cat in the fiddle, The cow jumped over the moon.
```

```
strlen("The cat in the fiddle, The cow jumped over the moon.") returned 52
MyStrlen("The cat in the fiddle, The cow jumped over the moon.") returned 52

------------------------------- END OF 7TH 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: C1A6E2_162461_U09339367
   Submitted: 2/16/2022 7:44:25 AM PST
   Course: C/C++ Programming I (Section 162461)
   Student's name: Phillip Ward
   Contact email: phillip.ward@seagate.com
   Student ID: U09339367
   Assignment 6, Exercise 2  (001610979M01005X76610)
   Exercise point value: 4
   Files submitted:
      C1A6E2_main.c
      C1A6E2_MyStrcmp.c

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

that?

```
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 02/12/2022
// C1A6E2_main.c
// Win10
// g++ 11.2.0
//
// A program tha compares two input strings
//
#include <stdio.h>
#include <string.h>
#define BUF_SIZE 100

int MyStrcmp(const char *s1, const char *s2);

int main(void) {
    char string1[BUF_SIZE];
    char string2[BUF_SIZE];
    //get input
    printf("Input a string: ");
    fgets(string1, BUF_SIZE, stdin);
    printf("Input another string: ");
    fgets(string2, BUF_SIZE, stdin);
    //replace newlines with null terminators
    string1[strcspn(string1, "\n")] = '\0';
    string2[strcspn(string2, "\n")] = '\0';
    //compute and print results
    printf("strcmp(\"%s\", \"%s\") returned %i\n",
        string1, string2, strcmp(string1, string2));
    printf("MyStrcmp(\"%s\", \"%s\") returned %i\n",
        string1, string2, MyStrcmp(string1, string2));
    return(0);
}
```

```c
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 02/14/2022
// C1A6E2_MyStrcmp.c
// Win10
// g++ 11.2.0
//
// A function that compares two strings and returns
// the difference between the first differing characters
//

int MyStrcmp(const char *s1, const char *s2)
{
    //While they're not equal and not at the end of the string
    while ((*s1 == *s2) && (*s1 != '\0'))
    {
        s1++;
        s2++;
    }
    return(*s1 - *s2);//return the difference
}
```

```
********** C1 ASSIGNMENT 6 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 a string: a
Input another string: B
strcmp("a", "B") returned 1
MyStrcmp("a", "B") returned 31

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

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

Input a string: HE
Input another string: HELLO
strcmp("HE", "HELLO") returned -1
MyStrcmp("HE", "HELLO") returned -76

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

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

Input a string: HE
Input another string: EHLLO
strcmp("HE", "EHLLO") returned 1
MyStrcmp("HE", "EHLLO") returned 3

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

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

Input a string: &=#
Input another string: ~!
strcmp("&=#", "~!") returned -1
MyStrcmp("&=#", "~!") returned -88

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

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

Input a string:
Input another string:
strcmp("", "") returned 0
MyStrcmp("", "") returned 0

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

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

Input a string: @
Input another string: @
strcmp("@", "@") returned 0
MyStrcmp("@", "@") returned 0
```

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

------------------------------ START OF 7TH RUN ------------------------------

Input a string: aaaaaaaaaaaaaaa
Input another string: z
strcmp("aaaaaaaaaaaaaaa", "z") returned -1
MyStrcmp("aaaaaaaaaaaaaaa", "z") returned -25

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

------------------------------ START OF 8TH RUN ------------------------------

Input a string: The cat
Input another string: in the fiddle
strcmp("The cat", "in the fiddle") returned -1
MyStrcmp("The cat", "in the fiddle") returned -21

------------------------------- END OF 8TH 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: C1A6E3_162461_U09339367
   Submitted: 2/20/2022 6:51:57 PM PST
   Course: C/C++ Programming I (Section 162461)
   Student's name: Phillip Ward
   Contact email: phillip.ward@seagate.com
   Student ID: U09339367
   Assignment 6, Exercise 3  (00157933M01005X44057)
   Exercise point value: 6
   Files submitted:
      C1A6E3_GetSubstring.c
      C1A6E3_main.c

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

```
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 02/20/2022
// C1A6E3_main.c
// Win10
// g++ 11.2.0
//
// A program that gets a substring from an input string
//
#include <stdio.h>
#include <string.h>
#define BUF_SIZE 256

char *GetSubstring(const char source[], int start, int count, char result[]);

int main(void) {
    char source[BUF_SIZE], result[BUF_SIZE];
    int start, count;

    printf("Input a string: ");
    fgets(source, BUF_SIZE, stdin);
    printf("Input the index of the starting character "
           "and the number of characters to capture:");
    scanf("%i %i", &start, &count);
    //replace newlines with null terminators
    source[strcspn(source, "\n")] = '\0';
    //compute and print results
    printf("\"%s\", %i, %i, extracts \"%s\"\n", source, start, count,
        GetSubstring(source, start, count, result));
    return(0);
}
```

```
//
// Phillip Ward U09339367
// Phillip.Ward@seagate.com
// C/C++ Programming I
// 162461 Ray Mitchell
// 02/20/2022
// C1A6E3_GetSubstring.c
// Win10
// g++ 11.2.0
//
// A function to return a substring
//

char *GetSubstring(const char source[], int start, int count, char result[])
{
    char *temp = result;
    //increment the source pointer to the place we'd like to start copying
    while (start > 0 && *source != '\0')
    {
        source++;
        start--;
    }
    //copy 'count' number of characters over to the result array
    while (count > 0 && *source != '\0')
    {
        *temp = *source;
        temp++;
        source++;
        count--;
    }
    *temp = '\0';
    return(result);
}
```

```
********** C1 ASSIGNMENT 6 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 a string: This is really fun
Input the index of the starting character and the number of characters to capture:2 25
"This is really fun", 2, 25, extracts "is is really fun"

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

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

Input a string: This is really fun
Input the index of the starting character and the number of characters to capture:27 9
"This is really fun", 27, 9, extracts ""

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

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

Input a string: This is really fun
Input the index of the starting character and the number of characters to capture:0 12
"This is really fun", 0, 12, extracts "This is real"

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

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

Input a string: one two three
Input the index of the starting character and the number of characters to capture:5 350
"one two three", 5, 350, extracts "wo three"

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

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

Input a string: one two three
Input the index of the starting character and the number of characters to capture:18 7
"one two three", 18, 7, extracts ""

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

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

Input a string: one two three
Input the index of the starting character and the number of characters to capture:6 5
"one two three", 6, 5, extracts "o thr"

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

------------------------------ START OF 7TH RUN ------------------------------

Input a string: one two three
Input the index of the starting character and the number of characters to capture:0 3
"one two three", 0, 3, extracts "one"

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

------------------------------ START OF 8TH RUN ------------------------------

Input a string:
Input the index of the starting character and the number of characters to capture:3 18
"", 3, 18, extracts ""

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