All of the work in this project is my own! I have not left copies of my code in public folders on university computers. I have not given any of this project to others. I will not give any portion of this project to others taking this class. I realize that the penalty for turning in work that is not my own can range from an "F" in the class to dismissal from Trinity University.

Print Title _____ Time Required = _____.____ Hrs.

Signature _____ (pledged)

Chapter 2 Homework

Individual Assignment
15 Points

Answers To These Questions Must Be Handwritten; No Electronic Solutions Will Be Accepted!
All programming/code questions refers to the C programming language.

When writing program statements, include semicolons where necessary! Remember that C is case sensitive!

Book – Chapter 2

1] P_____ D_____ The very first item in the program is the P_?_ C_?_.

2] _____ The function required of each and every C program is called _?_.

3] _____ Write the preprocessor command to include the library of input-output functions.

4] _____ Write the preprocessor command to include the library of string functions.

5] _____ Write the combination of symbols which begin an in-line comment.

6] _____ _____ The comment box begins with symbols _?_ and ends with symbols _?_.

7] _____ I_?_ allow us to name data and other objects in our program; since each object in the computer is stored at a unique address, these items allow us to manipulate the objects with the symbolic name..

8] _____ The_?_ type has no operations or values.

9] _____ _____ A _?_ is the amount of space necessary to store a character. It is _?_ bits in size.

10-12] _____ _____ _____ The standard **short int** type is _?_ bytes in size. This container may hold an integer is low as _?_ or as high as _?_

13-15] _____ _____ _____ Our **short int** type is _?_ bytes in size. This container may hold an integer is low as _?_ or as high as _?_

16-18] _____ _____ _____ The standard **int** type is _?_ bytes in size. This container may hold an integer is low as _?_ or as high as _?_

19-21] _____ _____ _____ Our **int** type is _?_ bytes in size. This container may hold an integer is low as _?_ billion or as high as _?_ billion

22] _____ Write the line of C Code that should be placed immediately before the brace at the bottom of function main to return control back to the operating system.

23] Place a check-mark beside each of the valid integers in the list below? If the value is not a valid integer, then explain why the others are invalid in the space to the right.

a. _____ 521 _____

b. _____ –32.0521 _____

c. _____ 5,621521 _____

d. _____ +00784521 _____

e. _____ +65521 _____

f. _____ 6521492183521 _____

g. _____ –0521 _____

24] Do a google search. Find and print a copy of the standard ASCII table.

25] _____ASCII is an acronym for _?_

Your job is to play computer. What is the output from the following segment of code that would be produced from each of the following output statements. Write the results in the space provided to the right. (if not sure, check it on the computer)

26] printf ("Average 1 = *%f*\n", 87.567); _____

27] printf ("Average 2 = *%3.2f*\n", 87.567); _____

28] printf ("Average 3 = *%3.2f*\n", 87.567); _____

29] printf ("Average 4 = *%8.2f*\n", 87.567); _____

30] printf ("Average 5 = *%-8.2f*\n", 87.567); _____

31] printf ("Average 6 = *%5.3f*\n", 87.567); _____

32] printf ("Average 7 = *%6.3f*\n", 87.567); _____

33] printf ("Exam 1 = *%i*\n", 87); _____

34] printf ("Exam 2 = *%d*\n", 87); _____

35] printf ("Exam 3 = *%1i*\n", 87); _____

36] printf ("Exam 4 = *%4ld*\n", 87); _____

37] printf ("Exam 5 = *%-4i*\n", 87); _____

38] printf ("Name 1 = *%s*\n", "Jane Doe"); _____

39] printf ("Name 2 = *%12s*\n", "Jane Doe"); _____

40] printf ("Name 3 = *%-12s*\n", "Jane Doe");_____

41] Puts ("\"It Is Better To"); _____

42] Puts ("Have Loved And Lost"); _____

43] printf ("Than To Have Never Loved At "); _____

44] putchar ('A'); _____

45] putchar ('L'); _____

46] putchar ('L'); _____

47] putchar (34); _____

48] putchar('\n'); _____

49] putchar (84); _____

50] putchar (72); _____

51] putchar (69); _____

52] putchar (32); _____

53] putchar (69); _____

54] putchar (78); _____

55] _____ What data type would be most appropriate for a **single exam score**
a) int
b) short int
c) long int

56] _____ What data type would be most appropriate for the **world population**
a) int
b) short int
c) long int

57] _____ What data type would be most appropriate for the **number of planets in the universe**
a) int
b) short int
c) long int

58] _____ What data type would be most appropriate for the **elevation (in feet) with respect to sea level**
a) int
b) short int
c) long int

59] _____ What data type would be most appropriate for an **hourly pay rate**
a) float
b) double
c) long double

60] _____ What data type would be most appropriate for an **gross national product**
a) float
b) double
c) long double

61] _____ What data type would be most appropriate for the **minimum wage**
a) float
b) double
c) long double

62] _____ What data type would be most appropriate for an **hourly pay rate**

a) float
b) double
c) long double

62] _____ What data type would be most appropriate for the **size of micro–organisms**

a) float
b) double
c) long double

62] What data type would be most appropriate for the **carpet size for a room**

a) float
b) double
c) long double

Convert Base 10 to Base 2 – Decimal to Binary

175 (base 10) = _____ (base 2)

[Euler's process for converting base 10 to base x]

Step 1 : Set Problem to Divide the quotient side by the base (175/2)
Step 2: 175/2 = New Quotient of 87 and New Remainder of 1
{Repeat Step 2 Until New Quotient = 0
Step 2: 87/2 = New Quotient of 43 and New Remainder of 1
Step 2: 43/2 = New Quotient of 21 and New Remainder of 1
Step 2: 21/2 = New Quotient of 10 and New Remainder of 1
Step 2: 10/2 = New Quotient of 5 and New Remainder of 0
Step 2: 5/2 = New Quotient of 2 and New Remainder of 1
Step 2: 2/2 = New Quotient of 1 and New Remainder of 0
Step 2: 1/2 = New Quotient of 0 and New Remainder of 1
Step 3: Remainder Side contains the solution in Reverse Order
(bottom–up) = 10101111 (base 2)

	Quotient	Remainder
Base 2	175	
	87	1
	43	1
	21	1
	10	1
	5	0
	2	1
	1	0
	0	1

63] 255 (base 10) = _____(base 2)

64] 127 (base 10) = _____(base 2)

65] 128 (base 10) = _____(base 2)

66] 64 (base 10) = _____(base 2)

67] 65 (base 10) = _____(base 2)

68] 65 (base 10) = _____(base 2)

68] 32767 (base 10) = _____(base 2)

69] 111011 (base 2) = _____(base 10)

70] 111010011 (base 2) = _____(base 10)

71] 11110101 (base 2) = _____(base 10)

72] 111010011 (base 2) = _____(base 10)

73] 1111011 (base 2) = _____(base 10)

74] Place a check-mark beside each of the valid identifiers for variables in the list below. (if not sure, check it on the computer)

_____ a. 7Up

_____ b. Payroll

_____ c. room222

_____ d. Name List

_____ e. a

_____ f. A1

_____ g. 1A

_____ h. Time&Place

_____ k. ListOfEmployees

_____ l. Lima,Ohio

_____ m. _Date

_____ n. No_employees

_____ o. NO_EMPLOYEES

76] Place a check-mark beside each of the valid valid constant declarations (if not sure, check it on the computer)

_____ a. # define PAY_RATE 6.25

_____ b. # define PAY_RATE = 6.25

_____ c. # define COMPANY 'General Motors'

_____ d. # define COMPANY 'General Motors';

_____ e. # define COMPANY "General Motors'

_____ f. # define COMPANY "General Motors";

_____ g. # define COMPANY "General Motors"

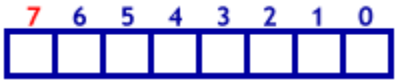
_____ h. # define Name "Peter Pan";

_____ i. # define LETTER_A 'A'

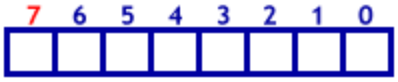
_____ j. # define LETTER_B "B"

_____ k. # define LETTER_C 'C';

77] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Sketch the binary representation of 100 (base 10).

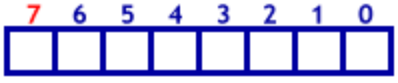


78] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Sketch the binary representation of 64 (base 10).

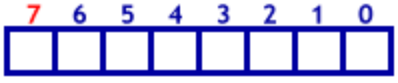


79] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Sketch the binary representation of 65 (base 10).

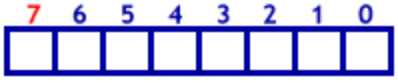
80] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Sketch the binary representation of 127 (base 10).



81] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Sketch the binary representation of 128 (base 10).

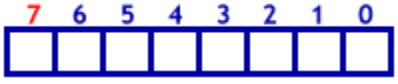


82] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Sketch the binary representation of 9 (base 10).

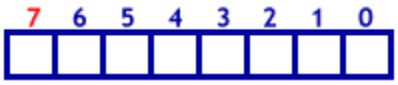


83] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “sign magnitude”. Sketch the binary representation of -100 (base 10).

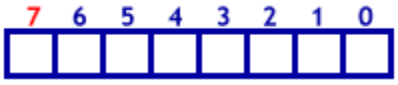
84] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “sign magnitude”. Sketch the binary representation of -64 (base 10).



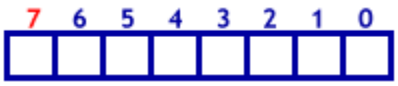
85] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “sign magnitude”. Sketch the binary representation of -65 (base 10).



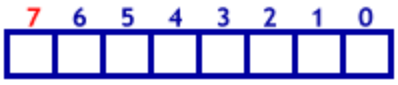
86] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “sign magnitude”. Sketch the binary representation of -127 (base 10).



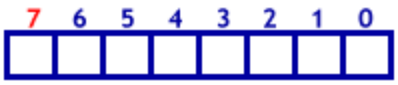
87] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “sign magnitude”. Sketch the binary representation of -128 (base 10).



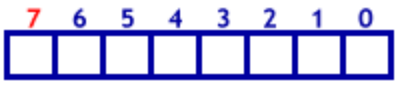
88] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “sign magnitude”. Sketch the binary representation of -9 (base 10).



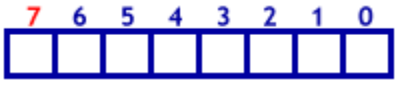
89] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “one’s complement”. Sketch the binary representation of -100 (base 10).



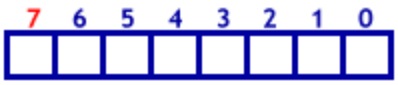
90] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “one’s complement”. Sketch the binary representation of -64 (base 10).



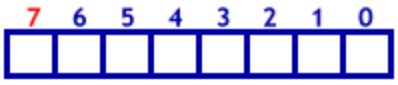
91] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “one’s complement”. Sketch the binary representation of -65 (base 10).



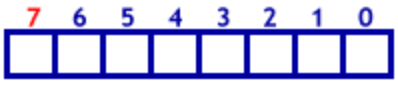
92] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “one’s complement”. Sketch the binary representation of -127 (base 10).



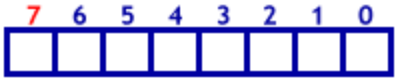
93] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “one’s complement”. Sketch the binary representation of -128 (base 10).



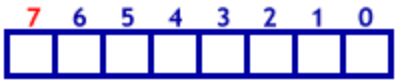
94] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “one’s complement”. Sketch the binary representation of -9 (base 10).



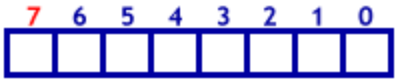
95] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “two’s complement”. Sketch the binary representation of -100 (base 10).



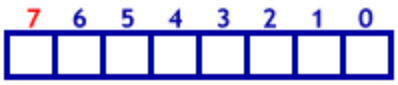
96] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “two’s complement”. Sketch the binary representation of -64 (base 10).



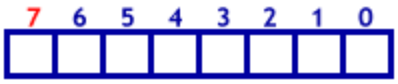
97] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “two’s complement”. Sketch the binary representation of -65 (base 10).



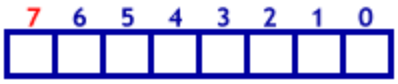
98] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “two’s complement”. Sketch the binary representation of -127 (base 10).



99] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “two’s complement”. Sketch the binary representation of -128 (base 10).



100] Assume that the compiler on this system follows the minimal standard of 1 byte for a short int. Assume, also, that negative numbers are stored using “two’s complement”. Sketch the binary representation of -9 (base 10).



101] _____ A _?_ {variable/constant} datatype can not be changed during program execution.

102] _____ A _?_ {variable/constant} datatype can be changed during program execution.

103] _____ Write the code to declare a constant, called **PI** and fill it with **3.14** without using **const** (Hint : use #define)

104] _____ Write the code to declare a short integer, called **No**;

105] _____ As declared above, **No** contains the value _?_.

106] _____ Write the code to declare a short integer, called **Age**, and fill/initialize it with **22;**

107] _____ Write the code display the contents of variable **Age** in the format (skip to the next line after the display!)
Age = 22

108] _____ Write the code to add **2** to the whatever value is currently in **Age**.

109] _____ Write the code use the keyboard to fill/input the **Age** from the user interaction.

110] _____ Write the code to replace whatever value is currently in **Age** with 22.

111] _____ Write the code to declare a short a floating point variable, called **PayRate**;

112] _____ As declared above, **PayRate** contains the value _?_.

113] _____ Write the code to declare a floating point variable, called **PayRate**, and fill/initialize it with **15.75**;

114] _____ Write the code display the contents of variable **PayRate** in the format
PayRate = 15.75

115] _____ Write the code to add **1.25** to the **PayRate**.

116] _____ Write the code to replace whatever value is currently in **PayRate** with **12.345**.

116] _____ Write the code use the keyboard to fill/input the **PayRate** from the user interaction.

117] _____ Write the code to declare a single character variable, called **Initial**.

118] _____ As declared above, **Initial** contains the value _?_.

119] _____ Write the code to declare a single character variable, called **Initial**, and fill/initialize it with the letter **T**.

120] _____ Write the code display the contents of variable **Initial** in the format
Initial = T

121] _____ Write the code to replace whatever value is currently in **Initial** with **K**.

122] _____ Write the code use the keyboard to fill/input the **Initial** from the user interaction.

123] _____ Write the code to declare a logical variable, called **Male**.

124] _____ As declared above, **Male** contains the value _?_.

125] _____ Write the code to declare a logical variable, called **Male**, and fill/initialize it with **false**.

126] _____ Write the code display the contents of variable **Male** in the format
Male = 0

127] _____ Write the code to replace whatever value is currently in **Male** with **true**.

128] I_____ Another principle of good structured programming is the use of I_?_ data names; this means that the variable name itself should give the reader a good idea as to what the data represents.

129] _____ {T/F} The C programming language enables programmers to create global variables that are available to all functions within a program.

130] _____ {T/F} Your book says that the use of global variables is a poor programming practice and should not be done for most programs.

131] _____ {T/F} We should never use any of the reserved key words of the language as variable identifiers (i.e. we should not do
char main;
int if;

132] _____ Appendix B has a list of the reserved key words in the C language. List ten of them.

132] _____ Write the line of code to clear the terminal window on a linux system.

133] A_____ O_____ In C, the ampersand (&) is known as the A_?_ O_?_.