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");
printf ("Than To Have Never Loved At ");

putchar ('A');
putchar ('L');
putchar ('L');
putchar (34);
putchar (84);
putchar (72);
putchar (69);
putchar (32);
putchar (69);
putchar (78);

45] _______  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 a 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)

<table>
<thead>
<tr>
<th>Quotient</th>
<th>Remainder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base 2</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

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).

```
    0 1 2 3 4 5 6
```

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).

```
    0 1 2 3 4 5 6
```

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).

```
    0 1 2 3 4 5 6
```

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).

```
    0 1 2 3 4 5 6
```

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).

```
    0 1 2 3 4 5 6
```

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).

```
    0 1 2 3 4 5 6
```

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).

```
    0 1 2 3 4 5 6
```

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).

```
    0 1 2 3 4 5 6
```

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).

```
    0 1 2 3 4 5 6
```
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).

```
 7 6 5 4 3 2 1 0
```

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).

```
 7 6 5 4 3 2 1 0
```

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).

```
 7 6 5 4 3 2 1 0
```

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).

```
 7 6 5 4 3 2 1 0
```

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).

```
 7 6 5 4 3 2 1 0
```

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).

```
 7 6 5 4 3 2 1 0
```

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
```
Write the code to add 2 to the whatever value is currently in \textit{Age}.

Write the code use the keyboard to fill/input the \textit{Age} from the user interaction.

Write the code to replace whatever value is currently in \textit{Age} with 22.

Write the code to declare a short a floating point variable, called \textit{PayRate};

As declared above, \textit{PayRate} contains the value \_?\_.

Write the code to declare a floating point variable, called \textit{PayRate}, and fill/initialize it with 15.75;

Write the code display the contents of variable \textit{PayRate} in the format \textit{PayRate} = 15.75

Write the code to add 1.25 to the \textit{PayRate}.

Write the code to replace whatever value is currently in \textit{PayRate} with 12.345.

Write the code use the keyboard to fill/input the \textit{PayRate} from the user interaction.

Write the code to declare a single character variable, called \textit{Initial}.

As declared above, \textit{Initial} contains the value \_?\_.

Write the code to declare a single character variable, called \textit{Initial}, and fill/initialize it with the letter \textit{T}.

Write the code display the contents of variable \textit{Initial} in the format \textit{Initial} = \textit{T}

Write the code to replace whatever value is currently in \textit{Initial} with \textit{K}.

Write the code use the keyboard to fill/input the \textit{Initial} from the user interaction.

Write the code to declare a logical variable, called \textit{Male}.

As declared above, \textit{Male} contains the value \_?\_.

Write the code to declare a logical variable, called \textit{Male}, and fill/initialize it with \textit{false}.

Write the code display the contents of variable \textit{Male} in the format \textit{Male} = 0
Write the code to replace whatever value is currently in Male with true.

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.

The C programming language enables programmers to create global variables that are available to all functions within a program.

Your book says that the use of global variables is a poor programming practice and should not be done for most programs.

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;

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

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

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

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