Task Document - Week 3

***Note to learner**: You MUST complete these exercises and upload your answers to the academy weekly, for your lecturer to review your progress and provide you feedback.*

Chapter 4 Exercise 1

StockSnackz Vending Machine

We dealt with our vending machine dispensing snacks throughout the day by a single highly abstract action "Dispense snacks." Now we have the constructs available to deal with optional and repeated actions we can extend the solution. Using pseudo-code notation, add any iterations and selections necessary to your solution from Chapter 3 to show many individual items being dispensed. You might want to start with the subproblem of dispensing the correct snack that corresponds to the button that was pressed. After that, move on to solving the problem of allowing this to happen repeatedly.

Chapter 4 Exercise 2

Stocksfield Fire Service

Using pseudo-code notation, add any selections and iterations necessary to your solution from Chapter 3 to the problem of decoding the hazchem Emergency Action Code.

Chapter 4 Exercise 3

Puzzle World: Roman Numerals and Chronograms

Go back to the problem of translating Roman numbers into decimal that you looked at in Chapter 3. You should be aware that just as English words can be misspelled, so can Roman numeral strings be malformed. For example, the Roman numeral string MCMC is malformed as it contradicts the syntax of the system. Before we can translate a Roman number into decimal we must first determine whether the Roman number is valid. If it is, then we can translate it into decimal. The rules for forming valid Roman numeral strings are:[7]

1. Smaller numerals follow larger numerals (see Rule 3 below). Summing the values of the numerals gives the value of the number.

2. The numerals I, X, C, and M (1, 10, 100, 1000 - all powers of 10) may be repeated up to three times in a row. No other numerals may be repeated.

3. Sometimes, a smaller numeral may precede a larger one (as in IV). These cases form compound numerals which are evaluated by subtracting the value of the smaller numeral from the larger one. To form a compound numeral all the following conditions must be met:

    a) The smaller numeral must be a power of ten (1, 10, 100, 1000).

b) The smaller numeral must be either one-fifth or one-tenth the value of the larger one.

c) The smaller numeral must either be the first numeral in the number, or follow a numeral of at least ten times its value.

d) If the compound numeral is followed by another numeral, that numeral must be smaller than the one that comes first in the compound numeral (i.e., you can have XCI but not IXI).
Work through the HTTLAP strategy to draft an outline solution to the problem of validating a Roman number.

Chapter 4 Exercise 4

Pangrams: Holoalphabetic Sentences

Using iteration and selection constructs (WHILE and IF) update the sequence of actions you produced in Chapter 3 for determining whether a sentence is a pangram.

Chapter 4 Exercise 5

Online Bookstore: ISBNs

The check digit in an ISBN is calculated by a Modulus 11 technique using the weights 10 to 2. This means that each of the first nine digits is multiplied by a number in a sequence from 10 to 2. If you add these products together and then add the value of the check digit, this total sum should be divisible by 11 without leaving a remainder. If it does give a remainder then the check digit does not match the rest of the number and we know the ISBN has been copied down incorrectly. The check digit is calculated in the following manner:

Multiply the first nine digits by 10, 9, 8, 7, . . ., 2 respectively and add the results. Divide this sum by 11 and take the remainder. Finally, subtract this remainder from 11 to give the check digit. If the value is 10 the check digit becomes "X." For example, we can validate the ISBN 0-14-012499-3 as follows:

$$\text{Check digit} = \frac{(0\times10)+(1\times9)+(4\times8)+(0\times7)+(1\times6)+(2\times5)+(4\times4)+(9\times3)+(9\times2)}{11}$$

$$= 118 \div 11$$

$$= 10, \text{ remainder } 8$$

So, the check digit = 11 - 8 = 3. As this is the same as the last number in 0-14-012499-3 we know that 0-14-012499-3 is a valid ISBN.
Using HTTLAP, write down the basic sequence of actions necessary to calculate the check digit for any ten-digit ISBN (assume the ISBN will be in a raw format without hyphens). Once you have identified the sequence, look to see if you can identify any repeated actions which could be better expressed using an iteration.

Chapter 5 Exercise 1

StockSnackz Vending Machine

Up till now we assumed the vending machine had unlimited supplies. It is time to put away such childish notions. Therefore, extend your solution so that the machine now shows a "sold out" message if it has run out of a selected item.

[13] Hint: One of the subproblems is finding out how far one turn of the pedals will move the bicycle. In gear 1, turning the pedals one full turn will rotate the rear wheel one full turn as the gear ratio is 1:1, thus the bicycle will travel 27 inches; In gear 5, the rear wheel will make 1.2857 full turns. Another subproblem is finding out how many times the pedal has to turn to move 1 km: how many inches are there in 1 km? You can start off by knowing that 1 inch 2.54 cm and there are 100 cm in 1 m and 1000 m in 1 km.

For testing purposes set the initial stock levels to 5 of each item (otherwise it will take your friend a long time to work through the algorithm!).
Previously if the buttons 0, 7, 8, 9 were pressed the machine simply did nothing. Now it is time to design a more typical response. The machine should behave as before when buttons 1-6 are pressed but should now show an "Invalid choice" message if the buttons 0, 7, 8, 9 are pressed. For both problems remember to write down all the variables (together with their ranges of values) that are needed.

Chapter 5 Exercise 2

Stocksfield Fire Service

Look at the algorithm you created in Chapter 4 for the hazchem problem. Identify and write down all the variables you think you may need for this solution. Remember to also indicate the typical ranges of values that each variable can take.

Chapter 5 Exercise 3

Puzzle World: Roman Numerals and Chronograms

Look at the algorithm you created in Chapter 4 for the Roman numerals validation and translation problems. Identify and write down all the variables you think you may need for your solutions. Remember to also indicate the typical ranges of values that each variable can take.

Chapter 5 Exercise 4

Pangrams: Holoalphabetic Sentences

Look at the algorithm you created in Chapter 4 for the pangram problem. Identify and write down all the variables you think you may need for this solution. Remember to also indicate the typical ranges of values that each variable can take.

Chapter 5 Exercise 5

Online Bookstore: ISBNs

Identify likely variables and their ranges of values for:

a) the ISBN hyphenation problem
b) the ISBN validation problem

Write your solution to the ISBN validation problem using an iteration to work through the nine main digits of the ISBN.

Chapter 6 Exercise 1

StockSnackz Vending Machine

Now that you have been introduced to the IF. ..ELSE construct, revise your previous vending machine solution to deal more elegantly with the problem of dispensing the chosen snack. Using IF...ELSE will also make the problem of dealing with buttons 0, 7, 8, and 9 much simpler.

Now that a proper selection construct has been used it is time to make the vending machine much more interesting. The University of Stocksfield is losing too much money through greedy staff stocking up on free snacks. With the exception of the sales summary (Button 6) all items must now be paid for and cost 10 pence each. If a user presses a button for a snack before sufficient money has been inserted an "insufficient funds" message should be displayed. If the user has deposited sufficient money for an item then the machine will dispense the chosen snack. Assume no change is given. Extend your solution to reflect these new requirements.

Now extend your solution so that if the user has deposited more than 10 pence the machine gives any required change after dispensing the chosen item. You may assume that the machine always has sufficient stock of each denomination of coin to be able to make exact change. The machine accepts (and gives back) the following denominations of coins: 1, 2, 5, 10, 20, 50 pence. Change should be dispensed using the fewest coins possible. Note, you have already solved the change-giving problem (see the exercises for Chapter 3) so see if that solution can be reused (perhaps with some amendments) here. There is a mathematical operator called modulo which gives the remainder after division. It often has the symbol %, but our pseudo-code uses MOD. It works like this: 20 ÷ 7 = 2 : 7 goes into 20 twice. 20 MOD 7 = 6 : the left over after dividing 20 by 7 is 6. You will find this useful for working out how to give change.

In the United Kingdom and countries using the euro currency, another two denominations of coin are available. The United Kingdom has £1 and £2 coins and the Euro Zone similarly has €1 and €2 coins. Extend your solution to cater for these larger denomination coins. If your machine works on US dollars and you would like to accept the rarer half-dollar and one-dollar coins, by all means go ahead.

Chapter 6 Exercise 2

Stocksfield Fire Service

In this chapter you learned about alternative iteration and selection constructs. Examine your EAC algorithm and replace IF statements with IF. ..ELSE constructs wherever possible. What benefits does this bring?

Chapter 6 Exercise 3

Puzzle World: Roman Numerals and Chronograms

Examine your solutions to the Roman number problems and decide whether you need to use any of the alternative iteration and selection constructs. Update your algorithms accordingly.

Chapter 6 Exercise 4

Pangrams: Holoalphabetic Sentences

The pangram algorithm needs to be amended so that it allows the test to be run on any number of sentences. The algorithm should keep testing sentences until the user decides to finish. Rewrite your solution to incorporate this feature. What iteration constructs could be used? How will you solve the problem of deciding whether the user wants to continue? Explain why you used your chosen solution strategy. What other loop constructs could you have used? How would that affect the way the algorithm behaves? What other ways could you have tested to see if the user wants to finish? What are the principal advantages and disadvantages of your solution and these alternative solutions?

Chapter 6 Exercise 5

Online Bookstore: ISBNs

The ISBN validation problem is best suited to using a count-controlled loop for the part which deals with multiplying the nine digits of the number with their respective weights. Update your solution replacing the WHILE construct with an appropriately phrased FOR loop.

You are now in a position to tackle the hyphenation problem. For correct presentation, the ten digits of an ISBN should be divided into four parts separated by hyphens:

- Part 1: The country or group of countries identifier
- Part 2: The publisher identifier
- Part 3: The title identifier
- Part 4: The check digit

Table 6.9 Hyphenation for Group "O" ISBNs

Group Identifier "O" Publisher Code Ranges	If Publisher Ranges are Between	Insert Hyphens After		
00----------19	00-19	1st digit	3rd digit	9th digit
200----------699	20-69	"	4th "	"
7000----------8499	70-84	"	5th "	"
85000--------89999	85-89	"	6th "	"
900000------949999	90-94	"	7th "	"
9500000----9999999	95-99	"	8th "	"

Table 6.10 Hyphenation for Group "1" ISBNs

Group Identifier "1" Publisher Code Ranges	If Publisher Ranges are between	Insert Hyphens After		
00----------09	00-09	1st digit	3rd digit	9th digit
100----------399	10-39	"	4th "	"
4000----------5499	40-54	"	5th "	"
55000--------86979	5500-8697	"	6th "	"
869800------998999	8698-9989	"	7th "	"
9990000----9999999	9990-9999	"	8th "	"

To keep matters as simple as possible we will only deal with hyphenating ISBNs that have a group/ country code of 0 or 1 (the English language groups). The positions of the hyphens are determined by the publisher codes. To hyphenate correctly, knowledge of the prefix ranges for each country or group of countries is needed. The publisher code ranges in the English group (US, UK, Canada, Australia, New Zealand, etc.) are given in Table 6.9.

Using Table 6.9 develop an algorithm for displaying with correctly placed hyphens any ISBN that starts with digit 0.[13, 14]

For an extra challenge, allow ISBNs with a group code of 1 to be hyphenated.

The rules for this group are slightly different than for group "O" and are given in Table 6.10.

This problem is slightly trickier than the group 0 hyphenation because you are not always just testing the first two digits of the publisher code. Use your solution to the group 0 hyphenation problem as a starting point and then work through the HTTLAP strategy to help you arrive at a solution to this problem.

[13] Hint: You can see that the publisher code takes between two and seven digits. Every ISBN with group identifier of 0 or 1 has a hyphen after the first digit (the group code) and after the ninth digit (i.e., immediately before the check digit). The third hyphen is inserted after the publisher code. For an ISBN in group 0 you can tell from the first two digits of the publisher code how long the rest of the code is. For example, looking at the table above we see that publisher codes beginning with digits in the range 20.69 are three digits in length, while those beginning 95..99 are seven digits long.

[14] Another hint: You need to manipulate the ISBN digit-by-digit. We have not covered how to access individual characters, so when it comes to that aspect use informal HTTLAP pseudo-code.

Chapter 9 Exercise 1

StockSnackz Vending Machine

Amend your solution to take into account any necessary data types introduced in this chapter. Consider carefully the data types needed to handle the monetary values.

Chapter 9 Exercise 2

Stocksfield Fire Service

Write your solution to the EAC decoding problem as an algorithm using the more formalised HTTLAP pseudo-code introduced in this chapter. Make sure you declare all your variables with appropriate data types. You need to think carefully about what you are going to use to store the EAC. The easiest method to get you started is to store each of the three characters of the EAC in separate character variables.

Chapter 9 Exercise 3

Puzzle World: Roman Numerals and Chronograms

Use strings and formal pseudo-code to rewrite the Roman number validator algorithm. Chronograms (also called eteostichons) are sentences in which certain letters when rearranged, stand for a date and the sentence itself is about the subject t which the date refers. All letters that are also Roman numerals (4, V, X, L, C, D, M are used to form the date. Sometimes the sentence is written such that the Roman numeral letters already give a well-formed Roman number. For example, in the sentence:

> *My Day Closed Is In Immortality*

if we ignore the lower-case letters we get the number MDCIII which equals 1603. The sentence commemorates the death of Queen Elizabeth the First of England in 1603. More commonly, the Roman numbers are not well formed and the date is obtained by adding the values of all the Roman numerals in the sentence, as in:

> *LorD haVe MerCI Vpon Vs. (V used as a U, mercy spelt with an 'i')*

This is a chronogram about the Great Fire of London in 1666. The date is given by

> L+D+ V+ M+ C+I+V+ V= 50 + 500 + 5 + 1000 + 100 + 1 + 5 + 5 = 1666.

Outline the basic algorithm for finding and displaying in decimal the date "hid den" in a chronogram. To begin, assume that only upper-case letters are used for roman numerals (I = 1, but i is a letter). Also, assume that the Roman numerals

[19] Hint: Think about the ordinal values of the characters. Some arithmetic is involved. Use the **Chr** and **Ord** commands mentioned in this chapter.

do not have to form a valid string of numerals and that the hidden date is obtained simply by summing the values of all roman numerals found (as in the 'Lord have mercy upon us" example above).

For an extra challenge, extend this solution to accept only chronograms that have a well-formed Roman number in them. Thus "My Day Closed Is In Immortality" would give the valid date MDCIII, whilst "LorD have MerCI Vpon Vs" would not give us a result as LDVMCIVV is not a well-formed number (1666 should be written as MDCLXVI).

Chapter 9 Exercise 4

Pangrams: Holoalphabetic Sentences

Update the variable list for your pangram solution by assigning proper HTTLAP types. Rewrite your algorithm using the more formal HTTLAP pseudo-code, assignment statements, Boolean expressions in IF and WHILE conditions, and so forth.

Chapter 9 Exercise 5

Online Bookstore: ISBNs

The ISBN validation problem is best solved using an iteration construct. Write your solution as an algorithm using the more formalized HTTLAP pseudo-code introduced in this chapter. Make sure you declare all your variables with appropriate data types. You need to think carefully about what you are going to use to store the whole raw 10-digit ISBN: you need to deal with each digit individually and the final digit may be an 'X'.