Faculty of Engineering, Environment and Computing
4043CEM Introduction to Programming and Algorithms

Assignment Brief 2022

Module Title **Introduction to Programming and Algorithms**	Ind	Cohort 2122JANMAY	Module Code: 4043CEM
Coursework Title: 4043CEM-CW			Hand out date: 22/02/2022
Lecturer: Dr Amjad Saeed Khan			Due date and time: Date: 31/03/2022, 18:00 BST
Estimated Time (hrs): 25 Word Limit*:	Coursework type: Individual		% of Module Mark 50%
Submission arrangement online via Aula and Coventry University GitHub File types and method of recording: **Source code programs (Python3 scripts)** Mark and Feedback date (DD/MM/YY): Mark and Feedback method (e.g. in lecture, electronic via Aula): **via Aula**			

Module Learning Outcomes Assessed:

1. Demonstrate an ability to use basic control flow syntax to produce working solutions to problems in a programming language.
2. Reason about simple algorithms, selecting or creating algorithms to solve specific and generalised problems.
3. Understand the need for, and begin to use, practices such as code testing, documentation and version control in professional programming environments.
4. Express, implement, use, and compare simple searching and sorting algorithms.

Task 1 and Mark distribution:

The World Health Organization (WHO) compiles data about immunization levels around the world. The file named "`measles.txt`" contains data about the level of measles vaccinations in various countries over time. Each line of the file contains the following fields, where there is one space between fields:

Country (50 characters)
Income Level (6 characters)
Percent Vaccinated (3 characters)
Region (25 characters)
Year (4 characters)

The "Country" field contains the name of the country.

The "Income Level" field identifies the category assigned to that country by the World Bank:

`WB_LI`	low income
`WB_LMI`	lower middle income
`WB_UMI`	upper middle income
`WB_HI`	high income

The "Percent Vaccinated" field contains an integer number representing the percentage of children in that country who have received measles vaccine by the age of one.

The "Region" field identifies the region assigned to that country by WHO.

The “Year” field contains the year for which the data was compiled.

Write two programs (using python3) entitled **Program_A** and **Program_B** that must have the following requirements.

Part		Assignment details	Marks
A		The **Program_A** will copy selected lines from “measles.txt” into a file selected by the user	
	a)	The program will always read from “measles.txt” (it will not prompt the user for the name of the input file). If it is unable to open that file, the program will halt.	1%
	b)	The program will prompt the user for the name of the output file. If that file does not exist, the program will create it and continue. If that file does exist, the program will discard the current contents of the file and continue.	1%
	c)	The program will prompt the user to enter a year, and will copy all lines of “measles.txt” selected by the user’s response. A line is selected if the user’s response matches the Year field or any of its prefixes. All lines are selected if the user’s response is any of the values in the set {“”, “all”, “ALL”}. Note that “” is the empty string. For example, a line whose Year field contains “1987” would be selected by any of the following user responses: {“1”, “19”, “198”, “1987”, “”, “all”, “”ALL”}.	6%
	d)	Document your code properly that should include at least a brief description of the functionality of your code, its input parameters or arguments, and about the function’s returns. In addition, added comments to lines of code where appropriate.	3%
	e)	Finally, write a test script for verifying the functional accuracy of your code. It should display appropriate messages to inform the user about any unusual circumstances.	4%
B		The **Program_B** will display one summary report to the user.	
	a)	The program will prompt the user to enter the name of the input file. If it is unable to open that file, the program will prompt the user again until the user enters a valid file name.	1%
	b)	The program will prompt the user to enter a year, and will then prompt the user to enter an income level. The income level must be one of the characters in the set {1, 2, 3, 4}, where 1 corresponds to “low income”, 2 corresponds to “lower middle income”, 3 corresponds to “upper middle income” and 4 corresponds to “high income”.	1%
	c)	The program will identify all records (lines) in the input file which match the user’s criteria for year and income level, and the program will display a report with the following information:	6%

		--The count of records in the input file which match the user's criteria --The average percentage for those records (displayed with one fractional digit) --The country with the lowest percentage for those records --The country with the highest percentage for those records The name of the country and the percent of children vaccinated will be displayed for the last two items (lowest percentage and highest percentage).	
	d)	Document your code properly that should include at least a brief description of the functionality of your code, its input parameters or arguments, and about the function's returns. In addition, added comments to lines of code where appropriate.	3%
	e)	Finally, write a test script for verifying the functional accuracy of your code. It should display appropriate messages to inform the user about any unusual circumstances.	4%

Task 2 and marks assignments:

A rotation cipher is one of the simplest, plain-text ciphers, known since at least the time of Julius Caesar. It takes in a plain-text string, and translates it into a new string based on a rotation of the alphabet being used. The basis is a "rotation", a re-sequencing of an alphabet. Consider the following example. Consider the alphabet being a single string consisting of the lower-case English letters as below (shown with each letter's associated index):

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

then a rotation of 3 means that the first three letters of the alphabet are moved to the end of the sequence, while the other letters move up, as shown below. Notice that the movement is done one letter at a time.

d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

A cipher is created by using the rotated alphabet to replace each letter in the original string with its rotated equivalent letter. Using the example above, the word "this" is translated as follows:

- the 't' is found at index 19 in the alphabet. The letter in the rotated alphabet is 'w'.
- the 'h' is found at index 7, the rotated letter is 'k'
- the 'i' is found at index 8, the rotated letter is 'l'
- the 's' is found at index 18, the rotated letter is 'v'

Thus, the string 'this' becomes the string 'wklv' using a rotation of 3.

Write a program (using python3) that must follow the instructions below:

Part		Assignment details	Marks
A	a)	The program should prompt the user for one of three commands: 1) 'e' to encode a string 2) 'd' to decode a string 3) 'q' to quit Any other command should raise an error and re-prompt.	1%

b) If the command is encode, then the program prompts for a string to encode and a rotation integer in the range of 1-25. The program then returns the encoded string. — 10%

1) Important, the program should **not encode** any letter that is not in the lower case alphabet. Those letters should simply be passed through to the encoded string

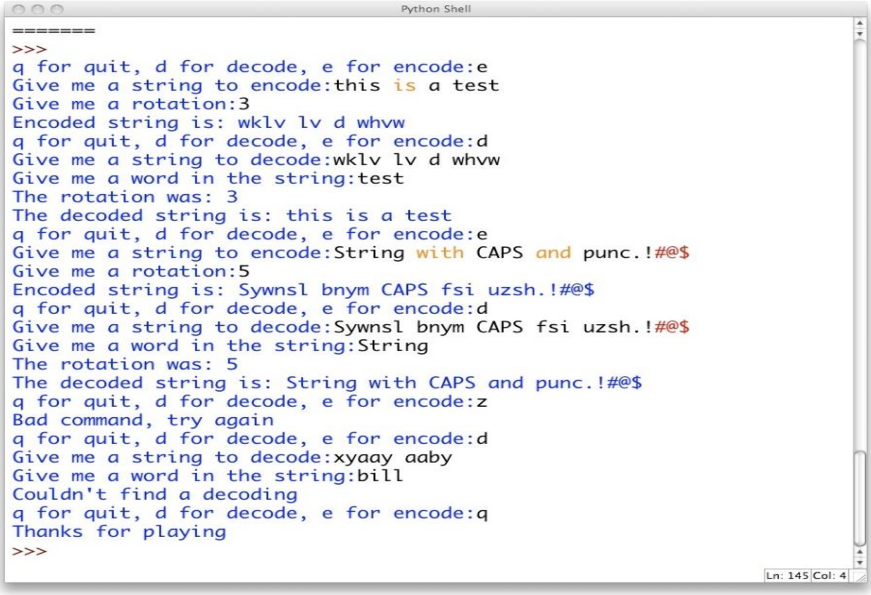
c) If the command is decode, then the program should prompt for a string to decode and a plain-text word that appears in the text (decoded string). The output should be the rotation needed to decode the string and the decoded string (text). — 10%

(1) If the program receives one word that belongs in the decoded string, then the program searches for a rotation that finds that word in the decoded string. That is the proper rotation. Finding that rotation is the goal of the program.

(2) If no rotation is found, then the program should indicate this fact.

d) If the command is quit, then the program ends and prints a nice exit message. — 1%

Example output:



```
=======
>>>
q for quit, d for decode, e for encode:e
Give me a string to encode:this is a test
Give me a rotation:3
Encoded string is: wklv lv d whvw
q for quit, d for decode, e for encode:d
Give me a string to decode:wklv lv d whvw
Give me a word in the string:test
The rotation was: 3
The decoded string is: this is a test
q for quit, d for decode, e for encode:e
Give me a string to encode:String with CAPS and punc.!#@$
Give me a rotation:5
Encoded string is: Sywnsl bnym CAPS fsi uzsh.!#@$
q for quit, d for decode, e for encode:d
Give me a string to decode:Sywnsl bnym CAPS fsi uzsh.!#@$
Give me a word in the string:String
The rotation was: 5
The decoded string is: String with CAPS and punc.!#@$
q for quit, d for decode, e for encode:z
Bad command, try again
q for quit, d for decode, e for encode:d
Give me a string to decode:xyaay aaby
Give me a word in the string:bill
Couldn't find a decoding
q for quit, d for decode, e for encode:q
Thanks for playing
>>>
```

e) Document your code properly that should include at least a brief description of the functionality of your code, its input parameters or arguments, and about the function’s returns. In addition, added comments to lines of code where appropriate. — 3%

f) Finally, write a test script for verifying the functional accuracy of your code. It should display appropriate messages to inform the user about any unusual circumstances. — 5%

Task 3 and marking distribution:

Use Turtle graphics to draw a picture containing multiple shapes of multiple colours and arranged to be visually pleasing. Although you are free to decide the precise shape(s) and layout for your picture, some aspect of the picture must depend on a numeric value input by the user. For example, the input might determine the size of the shapes, the number of shapes, or their spacing.

Write a program (using python3) that must follow the instructions below:

Part		Assignment details	Marks
A	a)	Output a brief descriptive message of what the program does.	1%
	b)	Repeatedly prompt for the input until the user supplies values of the correct form (discard incorrect inputs). Your prompt should say what form of input is needed.	1%
	c)	Draw a picture containing multiple shapes of multiple colors, where the input value(s) is (are) used to determine some aspect of the shapes and/or their layout. **Note:** In programming your solution, you must: 1. Use at least two repetition (*while* or *for*) statements. 2. Use at least one selection (*if*) statement.	10%
	d)	Document your code properly that should include at least a brief description of the functionality of your code, its input parameters or arguments, and about the function's returns. In addition, added comments to lines of code where appropriate.	4%
	e)	Finally, write a test script for verifying the functional accuracy of your code. It should display appropriate messages to inform the user about any unusual circumstances.	4%

Task 4 and marking distribution:

A sorting algorithm is an algorithm that takes an array of data as input, performs specified operations and puts its elements in a certain order. In this task, you are required to implement any two sorting algorithms of your own choice and compare their performance in terms of their execution time. For this purpose, you must write two scripts ProgramA and ProgramB. ProgramA should create a data file of random numbers (e.g., 1 million random numbers), and Program B should first read the data and then apply two different sorting algorithms (of your choice) on it (independently) as well as measures the sorting time corresponding to each algorithm. Following are the detailed instructions for this task:

Part		Assignment details	Marks

A)	a)	The programA will prompt the user for the name of the output file. If that file does not exist, the program will create it and continue. If that file does exist, the program will discard the current contents of the file and continue.	2%
	b)	The programA will then generate random numbers (let's say 1 million or your own choice of numbers) of any data type (i.e., integers or floating point) and write into the output file.	2%
B)	a)	The programB will prompt the user to enter the name of the input file. If it is unable to open that file, the programB will prompt the user again until the user enters a valid file name.	2%
	b)	The programB will apply two different sort algorithms (as per your choice) on the data independently and measure the time consumption of each algorithm. Finally, the program will display individual time consumptions of the algorithms as well as percentage difference of the times.	10%
	c)	Document your code properly that should include at least a brief description of the functionality of your code, its input parameters or arguments, and about the function's returns. In addition, added comments to lines of code where appropriate.	2%
	d)	Finally, write a test script for verifying the functional accuracy of your code. It should display appropriate messages to inform the user about any unusual circumstances.	2%

Deliverables: Your source code solutions (remember to include task number, and comments in each file).

- Your submitted files should be python scripts with **.py** extension, e.g., '**filename.py'**
- All the development work must be completed on Codio and to be submitted via the Coventry University GitHub repository at https://github.coventry.ac.uk
- Provide a link of your GitHub source code repository on Aula.

Notes:

1. You are expected to use the Coventry University APA style for referencing. For support and advice on this students can contact Centre for Academic Writing (CAW).
2. Please notify your registry course support team and module leader for disability support.
3. Any student requiring an extension or deferral should follow the university process as outlined here.
4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computer. Students should therefore regularly back-up any work and are advised to save it on the University system.
5. If there are technical or performance issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours or the next working day if the deadline falls on a Friday or over the weekend period. This will be communicated via your Module Leader.
6. You are encouraged to check the originality of your work by using the draft Turnitin links on Aula.
7. Collusion between students (where sections of your work are similar to the work submitted by other students in this or previous module cohorts) is taken extremely seriously and will be reported to the academic conduct panel. This applies to both courseworks and exam answers.

8. A marked difference between your writing style, knowledge and skill level demonstrated in class discussion, any test conditions and that demonstrated in a coursework assignment may result in you having to undertake a Viva Voce in order to prove the coursework assignment is entirely your own work.
9. If you make use of the services of a proof reader in your work you must keep your original version and make it available as a demonstration of your written efforts.
10. You must not submit work for assessment that you have already submitted (partially or in full), either for your current course or for another qualification of this university, with the exception of resits, where for the coursework, you maybe asked to rework and improve a previous attempt. This requirement will be specifically detailed in your assignment brief or specific course or module information. Where earlier work by you is citable, i.e. it has already been published/submitted, you must reference it clearly. Identical pieces of work submitted concurrently may also be considered to be self-plagiarism.

Mark allocation guidelines to students

0-39	40-49	50-59	60-69	70+	80+
Work mainly incomplete and /or weaknesses in most areas	Most elements completed; weaknesses outweigh strengths	Most elements are strong, minor weaknesses	Strengths in all elements	Most work exceeds the standard expected	All work substantially exceeds the standard expected