Computer Science A Level Transition Task

Programming section

Task 1

Write a program in Python that will output the Fibonacci Sequence.

The user should enter the number of outputs they would like.

The Fibonacci Sequence begins with 0 and 1. Each subsequent number is found by adding the two numbers before it.

The first 10 numbers are: 0, 1, 1, 2, 3, 5, 8, 13, 21 and 34.

You may use the internet for general programming tips, but don't copy the code; you need to be able to describe how your code works. Comment your code with techniques you have used: variable declaration and assignment, iteration, selection etc.

Extension work

- Ask the user for a name of a text file that has the file extension '.txt', e.g. myFile.txt
- Save the sequence to the file.

Task 2

Pig Latin is a game where words in a sentence are altered based on these rules.

1. If a word begins with a vowel, the string 'ay' is added to the end of the word.
2. If a word begins with a consonant, *all consonants before the first vowel* are removed and added to the end of the word. The string 'ay' is then added to the end of the word.

Examples

Banana would become Ananabay.

Cheese would become EeseChay.

Apple would become Appleay.

Your task

- Write a program where the user enters a string of characters and chooses whether to translate from English to Pig Latin or vice-versa.
- The program should convert the text based on the option selected by the user.
- The user should have the option to load the string to be entered from a text file.
- The user should have the option to save the output to a text file of a name of their choice.

Extension work

- Make your application have a graphical interface, using a library such as Tkinter.

Theory section

Task 3

Familiarise yourself with the following sets of numbers. Write down some examples of each.

- Natural numbers ($\mathbb{N}$)
- Integers ($\mathbb{Z}$)
- Rational numbers ($\mathbb{Q}$)
- Irrational numbers
- Real numbers ($\mathbb{R}$)
- Ordinal numbers

Draw a Euler diagram to represent the relationships between the sets.

Task 4

How many Bytes are in each of the following units of data? Express in the form of 2^n or 10^n.

- kibibyte, mebibyte, gibibyte, tebibyte
- kilobyte, megabyte, gigabyte, terabyte

Task 5

Hopefully, you can remember binary representation of numbers. In the GCSE we looked at whole numbers 0-255. In the A-Level course you will become familiar with binary representation of negative and fractional numbers initially.

Binary Representation of Fractional Numbers

One method of representing fractional numbers is fixed-point representation. This is where the binary point stays in the same place. In this 8-bit number, there binary point is between the 5^{th} and 6^{th} bits, although it is not shown:

Denary place value	16	8	4	2	1	0.5	0.25	0.125
Bit	1	0	1	0	1	0	1	0

The binary number shown in the table above is 21.25 or 21 ¼: 16 + 4 + 1 + 0.25

Calculate the decimal value of each of the following bit patterns.

	Place value							
	16	**8**	**4**	**2**	**1**	**0.5**	**0.25**	**0.125**
3a	1	1	1	0	1	0	0	0
3b	0	1	0	1	0	1	1	1
3c	1	1	1	0	0	1	1	1
3d	0	0	0	0	0	1	1	1
3e	1	1	1	0	0	1	1	0
3f	0	1	1	1	0	0	1	1
3g	1	0	1	0	1	1	1	0
3h	0	0	0	1	0	1	1	1
3i	0	1	0	1	0	1	0	0
3j	1	1	0	1	0	1	1	1

Binary Representation of Negative Numbers Using Two's Complement

One method of representing negative numbers is two's complement representation. This is where the most significant bit (the left-most) is the negative of whatever it would usually be:

Denary place value	**-128**	64	32	16	8	4	2	1
1Bit	1	0	1	0	1	0	1	0

The binary number shown in the table above is -86: -128 + 32 + 8 + 2

Any number with a 1 in the most significant bit will be negative. Any number with a 0 in the most significant bit will be positive.

Calculate the value of each of the following bit patterns:

	Place value							
	-128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
3k	1	1	1	0	1	0	0	0
3l	0	1	0	1	0	1	1	1
3m	1	1	1	0	0	1	1	1
3n	0	0	0	0	0	1	1	1
3o	1	1	1	0	0	1	1	0
3p	0	1	1	1	0	0	1	1
3q	1	0	1	0	1	1	1	0
3r	0	0	0	1	0	1	1	1
3s	0	1	0	1	0	1	0	0
3t	1	1	0	1	0	1	1	1