(H) Nothing But Net(works) (1/3) [15 points]

You have just crashed your spaceship at the Viterbi Spaceport. Being unfamiliar with spaceship repair, you're very much at a loss--but then a friendly-looking being from Rigel sidles up to you and says:

"ζ Ψ δ ξ   ϖ Ξ N ϕ   A ϕ Ω υ   Π Π α Σ"

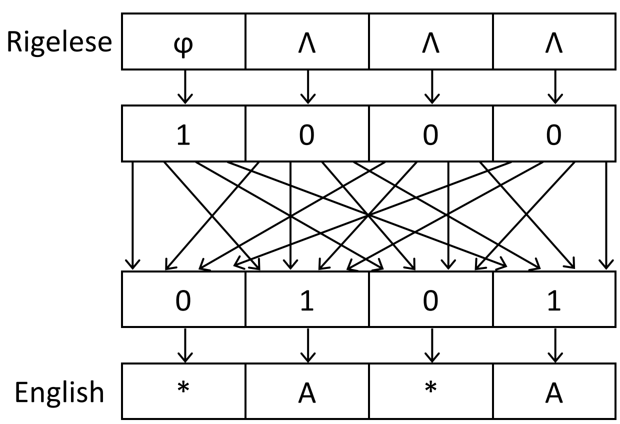
(Okay, maybe that's not so helpful after all.)

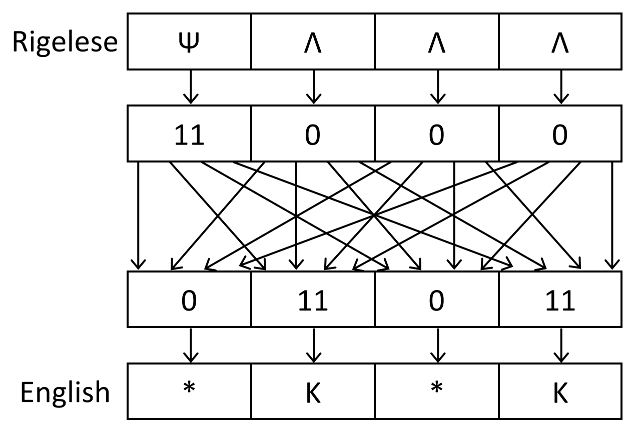
Luckily for you, English and Rigelese are related languages, and you own a GalactiLang translation device that can translate from the Rigelese sound system into more familiar English. This translator first turns the Rigelese word into a sequence of 4 numbers, then uses a neural network to transform those 4 numbers in some way (more about this in a minute), and then it transforms those final numbers into English letters using the following table:

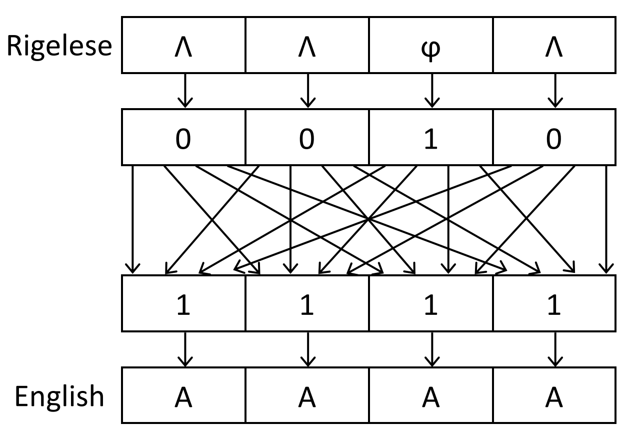
0	1	2	3	4	5	6	7	8	9	10	11	12	13
*	A	B	C	D	E	F	G	H	I	J	K	L	M

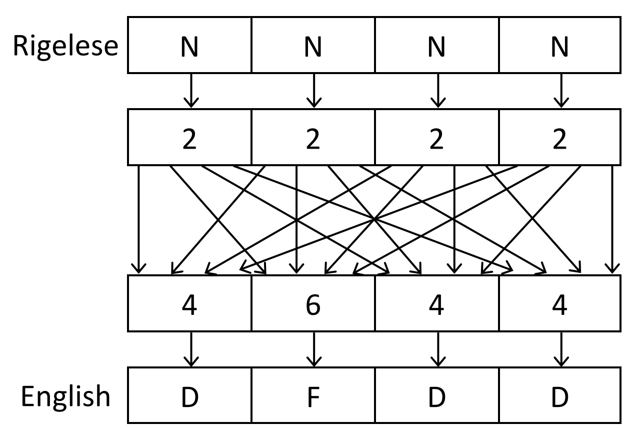
14	15	16	17	18	19	20	21	22	23	24	25	26
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Here are a few examples of the translator in action:

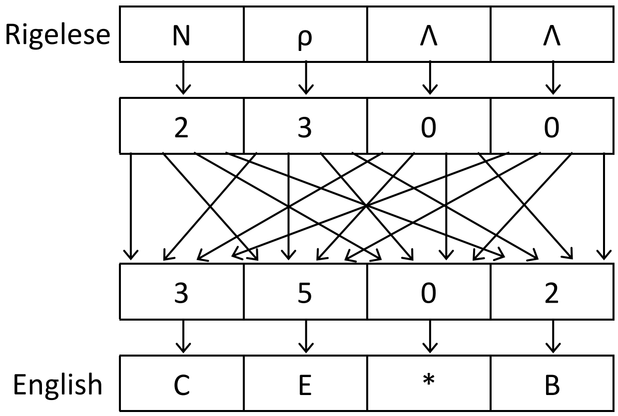


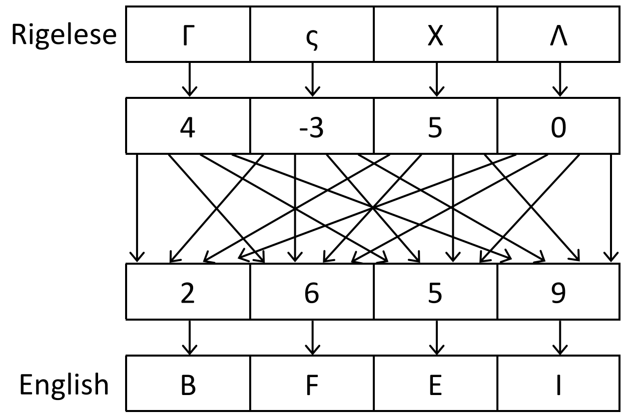




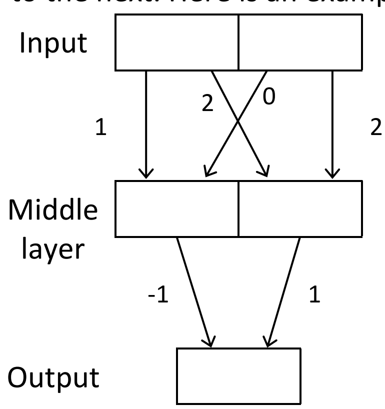


(H) Nothing But Net(works) (2/3)

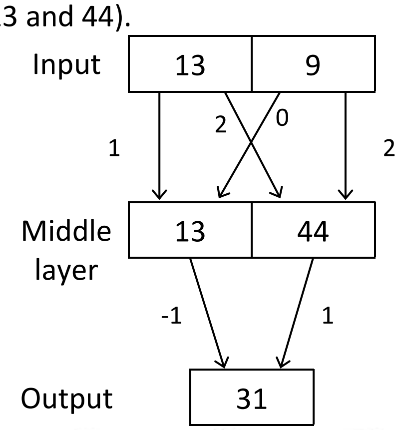




"But wait," you ask, "what's that big jumble of arrows in the middle of each translation?" To which we respond: The jumble of arrows stands for a neural network, which is an abstract computational structure that can be used to approximate any function. The network consists of several layers, including an input layer (the data to be processed), an output layer (the result of the computation), and potentially some middle layers in between the input and output layers. The network is trained on real data, and from this training process it learns how to transition from one layer to the next. Here is an example of a neural network:



This network takes two numbers as its input, then transitions from those two numbers to another two numbers in the middle layer, and then those two middle numbers get turned into a single output. The transitions between the layers are governed by the numbers written next to the arrows (these numbers are called weights). Here is an example of this network in action: Given the inputs 13 and 9, it yields the output 31 (after computing the middle layer of 13 and 44).

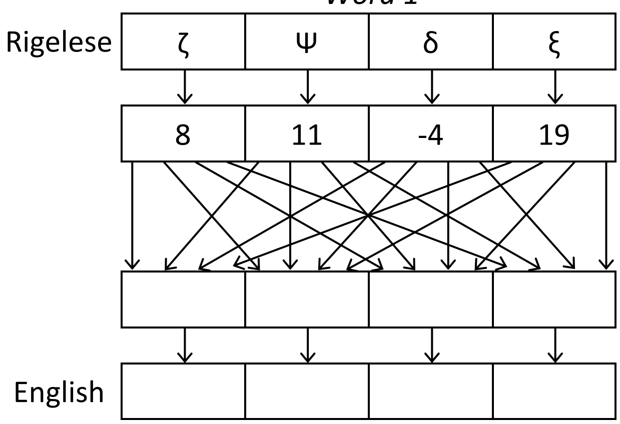


(H) Nothing But Net(works) (3/3)

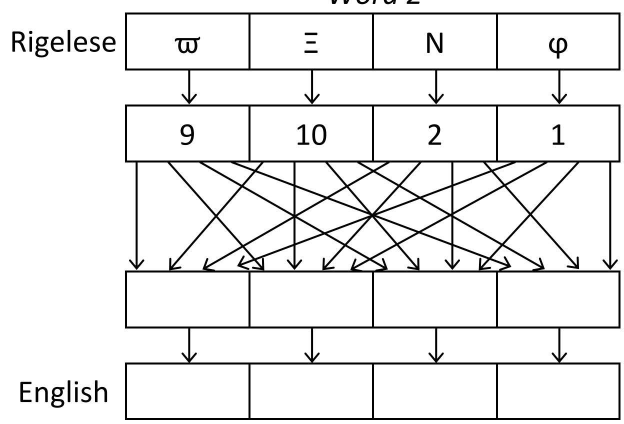
It is left to you to figure out exactly how the transitions are computed. In this case, if we call the inputs *a* and *b*, the output can be easily represented as $a + 2b$. However, neural networks can also represent many other more complex calculations that cannot be as easily expressed otherwise, and these other calculations have proven to be extremely useful in computational linguistic applications.

Now, returning to the Rigel example: When you try to translate the message from the Rigelian, your translator runs out of power after only computing one step of the translation. As a result, this is all that it gives you (each diagram represents the translation process for a single word):

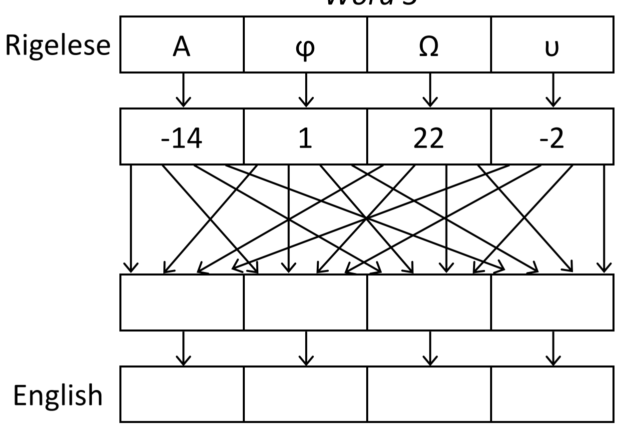
Word 1



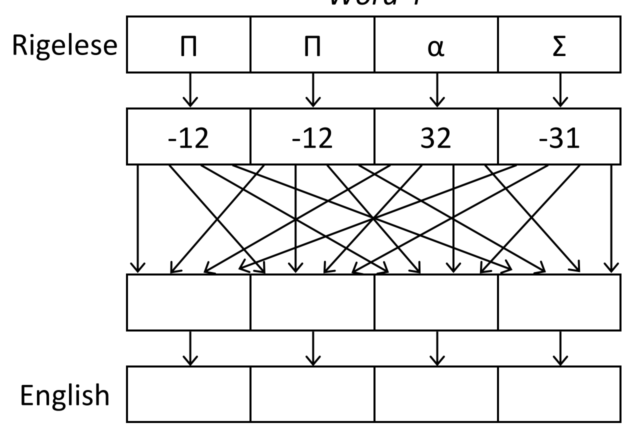
Word 2



Word 3



Word 4



H1. Finish the translation that the translator started. Write your answers in the Answer Sheets. Although you can see the six example translations at the start of this problem, you do not know what weights are attached to the arrows in the diagram (although you do know that the weights are the same across the translations for all four words). Therefore, you will have to use those diagrams to figure out the exact inner workings of the translator.

Word 1 | | | | |

Word 2 | | | | |

Word 3 | | | | |

Word 4 | | | | |