For assignment 3, describe the decoder section of the Transformer architecture. Include description of different layers - output embedding, Masked multi-head attention layer, Multi-head attention layer, Normalization layer, Feed forward network, and Residual connections. Also, answer this question - How is the output of the encoder used in the decoder. Submit a word or pdf file with your answers.

Transformer Decoder

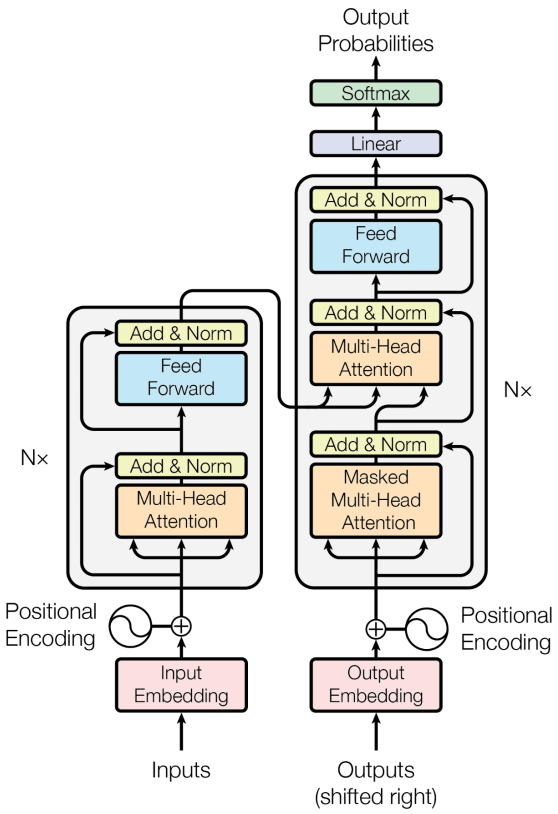
In [2]:

```
from PIL import Image
from IPython.display import Image, display, HTML
```

In [3]:

```
Image(filename='Transformer.jpg',width=300)
```

Out[3]:



The Transformer Decoder is an important part of Transformer model, since the model did not use RNN, this method is mainly use to solving sequence to sequence tasks.

Decoder section of the Transformer architecture

- Output Embedding
- positional Encoding
- Masked Multi-Head Attention layer
- Multi-Head Attention layer

- Normlization
- Feed foward layer
- Final output prediction with SoftMax activation function

Output Embedding and positional Encoding

same function as we described in class, converting tokens into continuous vector representation since Transformer is not using RNN model, it need to have its own way to remember the sequence of the word

Masked Multi-Head Attention layer

- a variation of the Multi-head Self-Attention mechanism used specifically in the Transformer Decoder. The purpose of the "masking" is to prevent the Decoder from attending to future tokens in the output sequence that it is trying to generate, thus ensuring that the model generates the output tokens in the correct order.

Multi-Head Attention layer

Same as the MHA we talk about in class, but here are some differences.

- we all know that MHA are using Q(Query),K(Key),V(Value) to calculate the weight sum. However, as we can see in the image above, the layer are using the Key and values from the encoding section and Query from the first layer of decoding section.

Normlization and Feed foward layer

- same as the encoding section use to normalize the paramaters and transform the input data by applying non-linear transformations

Final output prediction with SoftMax activation function

- linear layer and the final softmax layer work together to generate the output sequence, typically a probability distribution over the target vocabulary for each token in the sequence.

How is the output of the encoder used in the decoder.

As we described in above, the key and values from the encoder section are being used in the decoder section and bring the query from first layer of decoder section use to for final calculation

In [ ]: