This report describes the code in `tfidf.py`. The program calculates the cosine similarity between a query and a list of documents, which are represented as TF-IDF (Term Frequency Inverse Document Frequency) vectors.

The program reads a text as input where each line is interpreted as a document. The first document is assumed to be the query. The vector representation of each document is of length V, where V is the vocabulary size in the input corpus D. Each index i in the vectors represents a word ID according to a word-to-ID dictionary. The value of i represents the TF-IDF value for the word with that ID in a given document. TF-IDF for a term $t \in V$ in a document $d \in D$ is calculated as in equation 1:

$$\text{TF-IDF} = \frac{f_{t,d}}{|d|} \times \frac{|D|}{|\{d \in D \quad s.t. \quad t \in d\}|} \tag{1}$$

Where $f_{f,d}$ is the frequency of t in d. $|d|$ is the length of the document in words and $|D|$ is the number of documents in the corpus.

The similarity between these vector representations of a query and a document can be calculated using cosine similarity, which is simply the normalised dot-product as in equation 2:

$$cos(q, d) = \frac{\sum_{i=1}^{N} q_i d_i}{\sqrt{\sum_{i=1}^{N} q_i^2}\sqrt{\sum_{i=1}^{N} d_i^2}} \tag{2}$$

The program returns the line of the document in the corpus with the highest similarity with the query (note that the lines are counted from 1).

To make an example, imagine having three documents: $d_1 = \{a\ b\ c\}$, $d_2 = \{d\ e\ f\}$, $d_3 = \{a\ g\ f\}$. The query is d_1. Each document's vector representation will be a vector of length 7. The query's representation, for example, would be:

$$[0.135, 0.366, 0.366, 0.0, 0.0, 0.0, 0.0] \tag{3}$$

where each dimension is the TFIDF value for a term (in this case, only the first three a b c are non-zero). We can then calculate the cosine similarities between the query and the each of the other documents to find that d_3 is the most similar with similarity 0.039, while d_2 has similarity 0. Therefore, the program will output 3.