We have collected a corpus of 10 documents from wikipedia on topics: Cricket, Swimming, diving and batting and saved it in **corpus.txt** file.
We have come up with the following queries:

Query1: run_out cricket batter

- Relevant documents for this query: 1,2,3,4,5
- Our program will return the documents 1-5 in order based on cosine similarity

Query2: swimming_pool water diving

- Relevant documents for this query: 6,7,8,9,10
- Our program will return the documents 6-10 in order based on cosine similarity

The results after running our program are tabulated below:

	Res[1..5]	Rel	Precision@5	Recall@5
Query1	1,2,3,4,5	1,2,3,4,5	1	1
Query2	6,7,8,9,10	6,7,8,9,10	1	1

Table showing precision@5 and recall@5 for both queries

From the above table, we can conclude that our program performs as expected and return the relevant results.

Query that includes rare words found only in a few documents in the corpus will perform well
This comes from the tf-idf score associated with the terms in the following way
Tf-idf = $(\log(f_t , d) + 1)*(\log(N/N_t))$
The first component in the above product increases logarithmically with increasing frequency of a term in the query. An increase in the frequency of the term, will only result in a logarithmic increase in the first component.
The second component indicates that in the case where query term is contained only in a few documents, those documents will be ranked significantly higher because of the reciprocated N_t