Assignment 4 & 5

4. Implement a program to calculate precision and recall for sample input.
5.Write a program to calculate harmonic mean (F-measure) and E-measure.

```
#include <iostream>
#include <set>
#include <algorithm>
#include <iterator>
#include <string>

using namespace std;


double calculatePrecision(const set<string>& resultSet, const set<string>&
relevantSet) {
	set<string> commonElements;
	set_intersection(resultSet.begin(), resultSet.end(), relevantSet.begin(),
relevantSet.end(),
		inserter(commonElements, commonElements.begin()));
	return (resultSet.empty()) ? 0.0 :
static_cast<double>(commonElements.size()) / resultSet.size();
}


double calculateRecall(const set<string>& resultSet, const set<string>& relevantSet)
{
	set<string> commonElements;
	set_intersection(resultSet.begin(), resultSet.end(), relevantSet.begin(),
relevantSet.end(),
		inserter(commonElements, commonElements.begin()));
	return (relevantSet.empty()) ? 0.0 :
static_cast<double>(commonElements.size()) / relevantSet.size();
}


double calculateFMeasure(double precision, double recall) {
	if (precision + recall == 0) {
	return 0.0;
	}
	return 2.0 * (precision * recall) / (precision + recall);
}
```

```
double calculateEMeasure(double precision, double recall) {
	return (precision + recall) / 2.0;
}

int main() {
	set<string> resultSet;
	set<string> relevantSet;
	int count;
	string document;

	cout << "Enter the number of documents in the result set: ";
	cin >> count;
	cin.ignore();

	cout << "Enter the documents in the result set:\n";
	for (int i = 0; i < count; ++i) {
	getline(cin, document);
	resultSet.insert(document);
	}

	cout << "Enter the number of relevant documents: ";
	cin >> count;
	cin.ignore();

	cout << "Enter the relevant documents:\n";
	for (int i = 0; i < count; ++i) {
	getline(cin, document);
	relevantSet.insert(document);
	}

	double precision = calculatePrecision(resultSet, relevantSet);
	double recall = calculateRecall(resultSet, relevantSet);
	double fMeasure = calculateFMeasure(precision, recall);
	double eMeasure = calculateEMeasure(precision, recall);

	cout << "Precision: " << precision << endl;
	cout << "Recall: " << recall << endl;
	cout << "F-measure: " << fMeasure << endl;
	cout << "E-measure: " << eMeasure << endl;

	return 0;
}
```

Output:

```
Enter the number of documents in the result set: 4
Enter the documents in the result set:
doc1
doc2
doc3
doc4
Enter the number of relevant documents: 2
Enter the relevant documents:
doc1
doc4
Precision: 0.5
Recall: 1
F-measure: 0.666667
E-measure: 0.75
student@student:~$
```