**import** java.util.Scanner;

**public** **class** PrecisionRecall

{

**public** **static** **void** main(String[] args)

{

String Relevant\_arr[]= {"d3", "d5", "d9", "d25", "d39", "d44", "d56", "d71", "d89", "d123"}; // R

**int** Relevant\_size = Relevant\_arr.length; // |R|

String Retrieved\_arr [] = {"d123", "d84", "d56", "d6", "d8", "d9", "d511", "d129", "d187", "d25", "d38", "d48", "d250", "d113", "d3"}; // A

**int** Retrieved\_size = Retrieved\_arr.length; // |A|

**int** count = 0;

**double** precision,recall;

**double**[] F\_arr = **new** **double**[15]; // Array for storing F - values

// Arrays for storing values of E having 'b' greater than 1, zero and less than 1

**double**[] E\_b\_greater = **new** **double**[15];

**double**[] E\_b\_lesser = **new** **double**[15];

**double**[] E\_b\_equal = **new** **double**[15];

String Retrived\_string = "";

System.***out***.println(" Input\t\t\t\t\t\t\t\t | Precision | Recall");

**for**(**int** i=0;i<Retrieved\_size;i++)

{

**for**(**int** j=0;j<Relevant\_size;j++)

{

**if**(Retrieved\_arr[i]== Relevant\_arr[j])

count++;

}

**double** c = count;

precision = c/(i+1);

recall = c/ Relevant\_size;

F\_arr[i]= 2 / ((1 / recall) + (1 / precision));

**double** b = 1.1;

E\_b\_greater[i] = 1 -((1 + Math.*pow*(b, 2)) / ((Math.*pow*(b, 2)/recall) + (1 / precision)));

b = 0;

E\_b\_equal[i] = 1 -((1 + Math.*pow*(b, 2)) / ((Math.*pow*(b, 2)/recall) + (1 / precision)));

b = 0.9;

E\_b\_lesser[i]= 1 -((1 + Math.*pow*(b, 2)) / ((Math.*pow*(b, 2)/recall) + (1 / precision)));

Retrived\_string = Retrived\_string + " " + Retrieved\_arr[i];

System.***out***.print(Retrived\_string);

**for**(**int** m=1;m<=70-Retrived\_string.length();m++)

System.***out***.print(" ");

**double** p = precision \* 100;

**double** r = recall \*100;

//System.out.print("Precision :"+ String.format("%.2f", p) + "% "+ " | Recall : "+ String.format("%.2f", r) + " % \n");

System.***out***.print( "| "+ String.*format*("%.2f", p) + "% "+ " | "+ String.*format*("%.2f", r) + " % \n");

}

}

}

/\*OUTPUT:

Input | Precision | Recall

d123 | 100.00% | 10.00 %

d123 d84 | 50.00% | 10.00 %

d123 d84 d56 | 66.67% | 20.00 %

d123 d84 d56 d6 | 50.00% | 20.00 %

d123 d84 d56 d6 d8 | 40.00% | 20.00 %

d123 d84 d56 d6 d8 d9 | 50.00% | 30.00 %

d123 d84 d56 d6 d8 d9 d511 | 42.86% | 30.00 %

d123 d84 d56 d6 d8 d9 d511 d129 | 37.50% | 30.00 %

d123 d84 d56 d6 d8 d9 d511 d129 d187 | 33.33% | 30.00 %

d123 d84 d56 d6 d8 d9 d511 d129 d187 d25 | 40.00% | 40.00 %

d123 d84 d56 d6 d8 d9 d511 d129 d187 d25 d38 | 36.36% | 40.00 %

d123 d84 d56 d6 d8 d9 d511 d129 d187 d25 d38 d48 | 33.33% | 40.00 %

d123 d84 d56 d6 d8 d9 d511 d129 d187 d25 d38 d48 d250 | 30.77% | 40.00 %

d123 d84 d56 d6 d8 d9 d511 d129 d187 d25 d38 d48 d250 d113 | 28.57% | 40.00 %

d123 d84 d56 d6 d8 d9 d511 d129 d187 d25 d38 d48 d250 d113 d3| 33.33% | 50.00 %

\*/