Practical No.1

Title: Implement Conflation algorithm to generate document representative of a text file.

Program:

```
package assign1;
import java.io.File;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class Conflation
{
	public static void main(String[] args) throws IOException
	{
		try
		{
			File fi=new File("Input.txt");
			Scanner sc1=new Scanner(new File("Input.txt"));
			int ch,i,ans;
			do
			{
				System.out.println("1. Display the file");
				System.out.println("2. Remove Stop Words");
				System.out.println("3.Suffix Stripping");
				System.out.println("4. Count Frequency");
				System.out.println("Enter your choice");
				Scanner sc=new Scanner(System.in);
				ch=sc.nextInt();
				switch(ch)
				{
					case 1:
					while(sc1.hasNext())
					{
					System.out.print(sc1.next()+" ");
					}
					System.out.println(" ");
					break;
					case 2:
					remove_punctutaion(fi);
					//remove_stop_words(fi);
					break;
					case 3:
					suffix_stripping();
					break;
```

```
                                        case 4:
                                        frequency_count();
                                        break;
                                }
                        }while(ch!=4);

                }catch (FileNotFoundException e)
                {
                System.out.println(e);
                }
        }

        private static void remove_punctutaion(File fi)
        {
                try {
                        Scanner sc_punctuation=new Scanner(fi);
                        BufferedWriter out = new BufferedWriter(
                        new FileWriter("without_punctuation_and_stopwords.txt"));
                        while(sc_punctuation.hasNext())
                        {
                                String str_p=sc_punctuation.next();
                                String str_r=str_p.replaceAll("[^a-zA-Z\\s]", "");
                                if (!str_r.toLowerCase().equals("the") &&
!str_r.toLowerCase().equals("is") && !str_r.toLowerCase().equals("and") &&
!str_r.toLowerCase().equals("of") && !str_r.toLowerCase().equals("are") &&
!str_r.toLowerCase().equals("for") && !str_r.toLowerCase().equals("in"))
                                {
                                        out.write(str_r+" ");
                                }
                        }
                        out.close();
                        System.out.println("File after punctuation and stopwords:");
                        File testfile = new File("without_punctuation_and_stopwords.txt");
                   BufferedReader br= new BufferedReader(new FileReader(testfile));
                   String z;
                   while ((z = br.readLine()) != null)
                        System.out.println(z);
                   br.close();
                }
                catch (IOException e) {
                        System.out.println("exception occurred" + e);
                }
        }

        private static void suffix_stripping() throws FileNotFoundException,IOException
        {
                Scanner sc1=new Scanner(new
File("without_punctuation_and_stopwords.txt"));

                BufferedWriter out = new BufferedWriter(
                new FileWriter("suffix_stripping2.txt"));
                while (sc1.hasNext())
                {
                        String str=sc1.next();
                        str=str+"/";
                        if(str.endsWith("ier/"))
                        {
                        str=str.replaceAll("ier/", "y");
                        }
                        else if (str.endsWith("ied/"))
                        {
                        str=str.replaceAll("ied/", "y");
                        }
                        else if (str.endsWith("iage/"))
                        {
                        str=str.replaceAll("iage/", "y");
                        }
                        else if (str.endsWith("iest/"))
                        {
                        str=str.replaceAll("iest/", "y");
                        }
                        else if (str.endsWith("ies/"))
                        {
                        str=str.replaceAll("ies/", "y");
                        }
                        else if (str.endsWith("iful/"))
                        {
                        str=str.replaceAll("iful/", "y");
                        }
                        else if (str.endsWith("ify/"))
                        {
                        str=str.replaceAll("ify/", "y");
                        }
                        else if (str.endsWith("iness/"))
                        {
                        str=str.replaceAll("iness/", "y");
                        }
                        else if (str.endsWith("ness/"))
                        {
                        str=str.replaceAll("ness/", "y");
                        }
                        else if (str.endsWith("ily/"))
                        {
                        str=str.replaceAll("ily/", "y");
                        }
                        else if (str.endsWith("yer/"))
                        {
                        str=str.replaceAll("yer/", "y");
                        }
                        else if (str.endsWith("ying/"))
```

```
{
str=str.replaceAll("ying/", "y");
}
else if (str.endsWith("ys/"))
{
str=str.replaceAll("ys/", "y");
}
else if (str.endsWith("yable/"))
{
str=str.replaceAll("yable/", "y");
}
else if (str.endsWith("yful"))
{
str=str.replaceAll("yful", "y");
}
else if (str.endsWith("al/"))
{
str=str.replaceAll("al/", "y");
}
else if (str.endsWith("ly/"))
{
if(str.endsWith("ely/"))
{
str=str.replaceAll("ely/", "e");
}
else
{
str=str.replaceAll("ly/", "");
}
}
else if (str.endsWith("ing/"))
{
str=str.replaceAll("ing/", "y");
}
else if (str.endsWith("ed/"))
{
str=str.replaceAll("ed/", "y");
}
else if (str.endsWith("es/"))
{
str=str.replaceAll("es/", "y");
}
else if (str.endsWith("es/"))
{
str=str.replaceAll("es/", "y");
}
else if (str.endsWith("s/"))
{
str=str.replaceAll("s/", " ");
}
```

```
                else if (str.endsWith("is/"))
                {
                str=str.replaceAll("is", "y");
                }
                else if (str.endsWith("ment/"))
                {
                str=str.replaceAll("ment/", " ");
                }

                else if (str.endsWith("eing/"))
                {
                str=str.replaceAll("eing/", " ");
                }

                else if (str.endsWith("led/"))
                {
                str=str.replaceAll("led/", " ");
                }

                else if (str.endsWith("lex/"))
                {
                str=str.replaceAll("lex/", " ");
                }

                else if (str.endsWith("ling/"))
                {
                str=str.replaceAll("ling/", " ");
                }
                str=str.replace("/", " ");
                out.write(str+" ");

        }
        out.close();
        sc1.close();

        System.out.println("File after suffix Stripping:");
        File testfile = new File("suffix_stripping2.txt");
        BufferedReader br= new BufferedReader(new FileReader(testfile));
        String z;
        while ((z = br.readLine()) != null)
                System.out.println(z);
        br.close();
}

private static void frequency_count() throws FileNotFoundException,IOException
{
        Scanner sc3=new Scanner(new File("suffix_stripping2.txt"));
        int flag=0,i=0,l=0,ct=0,flag_w=0;
        String w[]=new String[1000];
        int cnt[]=new int[1000];
```

```
		while(sc3.hasNext())
		{
			w[i]=sc3.next();
			i++;
		}
		sc3.reset();
		Scanner sc5=new Scanner(new File("suffix_stripping2.txt"));
		while (sc5.hasNext())
		{
			String str1=sc5.next();
			for(int j=0;j<i;j++)
			{
				if(str1.equalsIgnoreCase(w[j]))
				{
				flag=1;
				cnt[j]++;
				}
			}
			if(flag==0)
			{
				w[i]=str1;
				cnt[i]=1;
				i++;
			}
		}
		for(int j=0;j<i;j++)
		{
			for(int k=j+1;k<i;k++)
			{
				if(w[j].equalsIgnoreCase(w[k]))
				{
					flag_w=0;
					break;
				}
				else
				{
					flag_w=1;
				}
			}
			if(flag_w==1)
			{
			System.out.println(w[j]+"."+cnt[j]+" ");
			}
		}
	}
}
```

Output:

```
1. Display the file
2. Remove Stop Words
3.Suffix Stripping
4. Count Frequency
Enter your choice
1
Astronomy is the study of everything in the universe beyond Earth's atmosphere. That
includes objects we can see with our naked eyes, like the Sun , the Moon , the planets, and
the stars . It also includes objects we can only see with telescopes or other instruments, like
faraway galaxies and tiny particles.
1. Display the file
2. Remove Stop Words
3.Suffix Stripping
4. Count Frequency
Enter your choice
2
File after punctuation and stopwords:
Astronomy study everything universe beyond Earths atmosphere That includes objects we
can see with our naked eyes like Sun  Moon  planets stars  It also includes objects we can
only see with telescopes or other instruments like faraway galaxies tiny particles
1. Display the file
2. Remove Stop Words
3.Suffix Stripping
4. Count Frequency
Enter your choice
3
File after suffix Stripping:
Astronomy  study  everythy universe  beyond  Earth  atmosphere  That  includy object  we
can  see  with  our  naky eyy like  Sun  Moon  planet  star  It  also  includy object  we  can  on
see  with  telescopy or  other  instrument  like  faraway  galaxy tiny  particly
1. Display the file
2. Remove Stop Words
3.Suffix Stripping
4. Count Frequency
Enter your choice
4
Astronomy.1
study.1
everythy.1
universe.1
beyond.1
Earth.1
atmosphere.1
That.1
our.1
naky.1
```

eyy.1
Sun.1
Moon.1
planet.1
star.1
It.1
also.1
includy.2
object.2
we.2
can.2
on.1
see.2
with.2
telescopy.1
or.1
other.1
instrument.1
like.2
faraway.1
galaxy.1
tiny.1
particly.1

Practical No.2

Title : Implement Single-pass Algorithm for clustering of files.(Consider 4 to 5 files).

Program:

```
package com.prac.prac;
 import java.io.BufferedReader;
 import java.io.IOException;
 import java.io.InputStreamReader;
 import java.util.ArrayList;
 public class singlepass {
     public static void main(String[] args) throws IOException{
         BufferedReader stdInpt = new BufferedReader(new InputStreamReader(System.in));
         System.out.println("Enter the no of Tokens");
         int noOfDocuments=Integer.parseInt(stdInpt.readLine());
         System.out.println("Enter the no of Documents");
         int noOfTokens=Integer.parseInt(stdInpt.readLine());
         System.out.println("Enter the threshhold");
         float threshhold=Float.parseFloat(stdInpt.readLine());
         System.out.println("Enter the Document Token Matrix");
         int [][]input= new int [noOfDocuments][noOfTokens];
         for(int i=0;i          {
             for(int j=0;j              {
                 System.out.println("Enter("+i+","+j+")");
                 input[i][j]=Integer.parseInt(stdInpt.readLine());
             }
         }
         SinglePassAlgorithm(noOfDocuments, noOfTokens, threshhold, input);
     }
     private static void SinglePassAlgorithm(int noOfDocuments,int noOfTokens,float threshhold,int
[][]input)
     {
         int [][] cluster = new int [noOfDocuments][noOfDocuments+1];
```

```
        ArrayList clusterRepresentative = new ArrayList();
        cluster [0][0]=1;
        cluster [0][1]=0;
        int noOfClusters=1;
        Float []temp= new Float[noOfTokens];
        temp=convertintArrToFloatArr(input[0]);
        clusterRepresentative.add(temp);
        for(int i=1;i          {
            float max=-1;
            int clusterId=-1;
            for(int j=0;j               {
                float
similarity=calculateSimilarity(convertintArrToFloatArr(input[i]),clusterRepresentative.get(j) );
                if(similarity>threshhold)
                {
                    if(similarity>max)
                    {
                     max=similarity;
                    clusterId=j;
                    }
                }
            }
            if(max==-1)
            {
                cluster[noOfClusters][0]=1;
                cluster[noOfClusters][1]=i;
                noOfClusters++;
                clusterRepresentative.add(convertintArrToFloatArr(input[i]));
            }
            else
            {
                cluster[clusterId][0]+=1;

                int index=cluster[clusterId][0];

                cluster[clusterId][index]=i;

clusterRepresentative.set(clusterId,calculateClusterRepresentative(cluster[clusterId],input,
noOfTokens));

            }

        }

        for(int i=0;i        {

            System.out.print("\n"+i+"\t");

            for(int j=1;j<=cluster[i][0];++j)

            {

                System.out.print(" "+cluster[i][j]);

            }

        }

    }

    private static Float[] convertintArrToFloatArr(int[] input)

    {

        int size=input.length;

        Float[] answer = new Float[size];

        for(int i=0;i        {

            answer[i]=(float)input[i];

        }

        return answer;

    }

    private static float calculateSimilarity(Float[] a,Float[] b)

    {

        float answer=0;

        for(int i=0;i        {

            answer+=a[i]*b[i];

        }

        return answer;
    }
    private static Float[] calculateClusterRepresentative(int[] cluster,int [][] input,int noOFTokens)
    {
        Float[] answer= new Float[noOFTokens];
        for(int i=0;i          {
            answer[i]=Float.parseFloat("0");
        }
        for(int i=1;i<=cluster[0];++i)
        {
            for(int j=0;j               {
                answer[j]+=input[cluster[i]][j];
            }
        }
        for(int i=0;i          {
            answer[i]/=cluster[0];
        }
        return answer;
    }
}
```

Output :

```
Enter the no of Tokens
5
Enter the no of Documents
5
Enter the threshhold
10
Enter the Document Token Matrix
Enter(0,0)
1
Enter(0,1)
3
Enter(0,2)
3
Enter(0,3)
2
Enter(0,4)
2
Enter(1,0)
2
Enter(1,1)
1
Enter(1,2)
0
Enter(1,3)
1
Enter(1,4)
2
Enter(2,0)
0
Enter(2,1)
2
Enter(2,2)
0
Enter(2,3)
0
Enter(2,4)
1
Enter(3,0)
0
Enter(3,1)
3
Enter(3,2)
1
Enter(3,3)
0
Enter(3,4)
5
Enter(4,0)
```

```
1
Enter(4,1)
0
Enter(4,2)
1
Enter(4,3)
0
Enter(4,4)
1


0 0 1 3
1 2
2 4
```

Practical No.3

Title : Implement a program for retrieval of documents using inverted files.

Program:

```
#include<iostream>
#include<vector>
#include<map>
#include<string>
#include<fstream>
#include<sstream>
using namespace std;

struct word_position
{
  string file_name;
  int line;
  int index;
};

class InvertedIndex
{
  map<string,vector<word_position> > Dictionary;
  vector<string> filelist;

  public:
    void addfile(string filename);
    void show_files();
    void search(string word);
};
```

```
void InvertedIndex::addfile(string filename)
{
  ifstream fp;
  fp.open(filename + ".txt",ios::in);

  if(!fp)
  {
    cout<<"File Not Found\n";
    return ;
  }

  filelist.push_back(filename);

  string line,word;
  int line_number=0,word_number=0;
  while(getline(fp,line))
  {
    line_number++;
    word_number = 0;
    stringstream s(line);
    while(s>>word)
    {
      word_number++;
      word_position obj;
      obj.file_name = filename;
      obj.line = line_number;
      obj.index = word_number;
      Dictionary[word].push_back(obj);
    }
  }
  fp.close();
}

void InvertedIndex::show_files()
{
  int size = (int)filelist.size();
  for(int i=0;i<size;i++) cout<<i+1<<": "<<filelist[i]<<endl;

  if(!size) cout<<"No files added\n";
}

void InvertedIndex::search(string word)
{
  if(Dictionary.find(word)==Dictionary.end())
  {
    cout<<"No instance exist\n";
    return ;
  }



  int size = (int)Dictionary[word].size();
  for(int counter = 0;counter < size ;counter++)
  {
    cout<<counter+1<<":\n";
    cout<<"   Filename: "<<Dictionary[word][counter].file_name<<endl;
    cout<<"   Line Number: "<<Dictionary[word][counter].line<<endl;
    cout<<"   Index: "<<Dictionary[word][counter].index<<endl;
  }
```

```
}

int main(int argc, char*argv[])
{
  InvertedIndex Data;
  for(int i = 1 ; i< argc ; i++)
  {
    Data.addfile(argv[i]);
  }

  int choice = 0;
  do
  {
    cout<<"1: See files\n2: Add File\n3: Query Word\n4: Exit\n";
    cin>>choice;
    switch(choice)
    {
      case 1: Data.show_files(); break;
      case 2:
      {
        cout<<"Enter File Name: ";
        string name;
        cin>>name;
        Data.addfile(name);
        break;
      }

      case 3:
      {
```

```
      cout<<"Enter Word: ";
      string word;
      cin>>word;
      Data.search(word);
      break;
    }

    case 4: break;

    default : continue;
  }
}while(choice!=4);

return 0;
```

Output :

```
1: See files 2: Add File 3: Query Word 4: Exit
1
No files added 1: See files 2: Add File 3: Query Word 4: Exit
2
Enter File Name:
ABC.txt
File Not Found 1: See files 2: Add File 3: Query Word 4: Exit
3
Enter Word:
ABC
No instance exist 1: See files 2: Add File 3: Query Word 4: Exit
4
** Process exited - Return Code: 0 **
```

Practical No.4

Title : Implement a program to calculate precision and recall for sample input. (Answer set A, Query q1, Relevant documents to query q1- Rq1)

Program:

```
#include <iostream>

#include <string.h>

#include <iomanip>

#include <fstream>


using namespace std;


string left(const string s, const int w)

{ // Left aligns input string in table

    stringstream ss, spaces;

    int padding = w - s.size(); // count excess room to pad

    for (int i = 0; i < padding; ++i)

        spaces << " ";

    ss << s << spaces.str() << '|'; // format with padding

    return ss.str();

}


string center(const string s, const int w)

{ // center aligns input string in table

    stringstream ss, spaces;

    int padding = w - s.size(); // count excess room to pad

    for (int i = 0; i < padding / 2; ++i)

        spaces << " ";

    ss << spaces.str() << s << spaces.str(); // format with padding

    if (padding > 0 && padding % 2 != 0)     // if odd #, add 1 space
```

```
        ss << " ";
    return ss.str();
}

string prd(float x, int decDigits, int width)
{ // right aligns float values with specified no. of precision digits in a table
    stringstream ss;
    ss << fixed << right;
    ss.fill(' ');             // fill space around displayed #
    ss.width(width);          // set  width around displayed #
    ss.precision(decDigits); // set # places after decimal
    ss << x;
    return ss.str();
}

string printDocs(string state[], int size)
{
    // prints each document at a specific iteration inside the table
    stringstream ss;
    ss << '|' << ' ';
    for (int i = 0; i < size; i++)
    { // convert the array into a string of comma seprated values
        ss << state[i];
        if (state[i].compare("") != 0 and i + 1 < size and state[i + 1].compare("") != 0)
            ss << ',' << ' ';
    }
    return left(ss.str(), 98);
}
float E_value(float b, float rj, float pj)
```

```
{ // calculates E value
    return 1 - (((1 + b * b) * rj * pj) / (b * b * pj + rj));
}


int main()
{ // Hardcoded Rq and A
    string Rq[10] = {"d3", "d5", "d9", "d25", "d39", "d44", "d56", "d71", "d89", "d123"};
    string A[15] = {"d123", "d84", "d56", "d6", "d8", "d9", "d511", "d129", "d187", "d25",
"d38", "d48", "d250", "d113", "d3"};


    // Creating and opening output file
    ofstream write("Recall_Precision_Evaluation_output.txt");


    // required constants and arrays for calculations
    float modRq = sizeof(Rq) / sizeof(Rq[0]);
    string Ra[sizeof(A) / sizeof(A[0])];
    float P[sizeof(A) / sizeof(A[0])];
    float R[sizeof(A) / sizeof(A[0])];
    float modRa = 0;
    float modA = 0;
    double precision;
    double recall;


    // table header formatting and printing
    std::cout << setprecision(2) << fixed;
    write << setprecision(2) << fixed;
    std::cout << string(45 * 3 + 11, '-') << "\n";
    write << string(45 * 3 + 11, '-') << "\n";
    std::cout << '|' << center("Documents", 96) << " | "
              << center("|Ra|", 8) << " | "
```

```
          << center("|A|", 8) << " | "
          << center("Precision(%)", 5) << "|"
          << center("Recall(%)", 5) << " | " << endl;
write << '|' << center("Documents", 96) << " | "
      << center("|Ra|", 8) << " | "
      << center("|A|", 8) << " | "
      << center("Precision(%)", 5) << "|"
      << center("Recall(%)", 5) << " | " << endl;
std::cout << string(45 * 3 + 11, '-') << "\n";
write << string(45 * 3 + 11, '-') << "\n";
// Algorithm to calculate and print all the values in the output table, MAIN algo
for (int i = 0; i < sizeof(A) / sizeof(A[0]); i++)
{
    Ra[i] = A[i];
    modA++;
    for (int j = 0; j < modRq; j++)
    {
        if (A[i] == Rq[j])
        {
            modRa++;
            break;
        }
    }
    precision = (modRa / modA) * 100;
    P[i] = precision / 100;
    recall = (modRa / modRq) * 100;
    R[i] = recall / 100;
    // Printing documents and other values of current iteration within the table
    std::cout << printDocs(Ra, sizeof(Ra) / sizeof(Ra[0]));
```

```
        write << printDocs(Ra, sizeof(Ra) / sizeof(Ra[0]));
        std::cout << prd(modRa, 2, 10) << "|"
                  << prd(modA, 2, 10) << "|"
                  << prd(precision, 2, 13) << "|"
                  << prd(recall, 2, 10) << "|"
                  << endl;
        write << prd(modRa, 2, 10) << "|"
              << prd(modA, 2, 10) << "|"
              << prd(precision, 2, 13) << "|"
              << prd(recall, 2, 10) << "|"
              << endl;
    }
    // closing the table
    std::cout << string(45 * 3 + 11, '-') << "\n";
    write << string(45 * 3 + 11, '-') << "\n";

    // taking user input for calculation of Fj and Ej
    int j;
    do
    {
        std::cout << "Harmonic mean and E-value\nEnter value of j(0 - " << (sizeof(A) /
sizeof(A[0])) - 1 << ") to find F(j)and E(j):" << endl;
        cin >> j;
    } while (j > sizeof(Ra) / sizeof(Ra[0]));

    // calculating Harmonic mean and printing in table
    float Fj = (2 * P[j] * R[j]) / (P[j] + R[j]);
    std::cout << string(15 * 2 + 3, '-') << "\n"
              << "| Harmonic mean (F" << j << ") is: |" << Fj << " |\n"
              << string(15 * 2 + 3, '-') << "\n";
```

```
write << string(15 * 2 + 3, '-') << "\n"
      << "| Harmonic mean (F" << j << ") is: |" << Fj << " |\n"
      << string(15 * 2 + 3, '-') << "\n";

// table header
std::cout << string(15 * 2 + 4, '-') << "\n"
          << "|" << center("E-Value", 32) << "|\n"
          << string(15 * 2 + 4, '-') << "\n";
write << string(15 * 2 + 4, '-') << "\n"
      << "|" << center("E-Value", 32) << "|\n"
      << string(15 * 2 + 4, '-') << "\n";

// table header (sub columns)
std::cout << "|" << center("b>1", 10) << "|"
          << center("b=0", 10) << "|"
          << center("b<1", 10) << "|\n"
          << string(15 * 2 + 4, '-') << "\n";
write << "|" << center("b>1", 10) << "|"
      << center("b=0", 10) << "|"
      << center("b<1", 10) << "|\n"
      << string(15 * 2 + 4, '-') << "\n";

// Calculating and Printing E-Values in table
std::cout << "|" << prd(E_value(1.1, R[j], P[j]), 2, 10) << "|"
          << prd(E_value(0, R[j], P[j]), 2, 10) << "|"
          << prd(E_value(0.9, R[j], P[j]), 2, 10) << "|\n";
write << "|" << prd(E_value(1.1, R[j], P[j]), 2, 10) << "|"
      << prd(E_value(0, R[j], P[j]), 2, 10) << "|"
      << prd(E_value(0.9, R[j], P[j]), 2, 10) << "|\n";
```

```
    // Closing table
    std::cout << string(15 * 2 + 4, '-') << "\n";
    write << string(15 * 2 + 4, '-') << "\n";
    write.close();
    return 0;
}
```

Output:

```
|                         Documents                           |  |Ra|  |  |A|   |
Precision(%)|Recall(%) |
---------------------------------------------------------------------------------------------------------------------------
| d123                                                     |    1.00|    1.00|     100.00|    10.00|
| d123, d84                                                |    1.00|    2.00|      50.00|    10.00|
| d123, d84, d56                                           |    2.00|    3.00|      66.67|
20.00|
| d123, d84, d56, d6                                       |    2.00|    4.00|      50.00|
20.00|
| d123, d84, d56, d6, d8                                   |    2.00|    5.00|      40.00|
20.00|
| d123, d84, d56, d6, d8, d9                               |    3.00|    6.00|      50.00|
30.00|
| d123, d84, d56, d6, d8, d9, d511                         |    3.00|    7.00|      42.86|
30.00|
| d123, d84, d56, d6, d8, d9, d511, d129                   |    3.00|    8.00|
37.50|    30.00|
| d123, d84, d56, d6, d8, d9, d511, d129, d187             |    3.00|    9.00|
33.33|    30.00|
| d123, d84, d56, d6, d8, d9, d511, d129, d187, d25        |    4.00|   10.00|
40.00|    40.00|
| d123, d84, d56, d6, d8, d9, d511, d129, d187, d25, d38   |    4.00|   11.00|
36.36|    40.00|
| d123, d84, d56, d6, d8, d9, d511, d129, d187, d25, d38, d48    |    4.00|
12.00|      33.33|    40.00|
| d123, d84, d56, d6, d8, d9, d511, d129, d187, d25, d38, d48, d250    |    4.00|
13.00|      30.77|    40.00|
| d123, d84, d56, d6, d8, d9, d511, d129, d187, d25, d38, d48, d250, d113    |    4.00|
14.00|      28.57|    40.00|
| d123, d84, d56, d6, d8, d9, d511, d129, d187, d25, d38, d48, d250, d113, d3    |    5.00|
15.00|      33.33|    50.00|
Harmonic mean and E-value

Enter value of j(0 - 14) to find F(j)and E(j):

10

---------------------------------

| Harmonic mean (F10) is: |0.38 |

---------------------------------

----------------------------------

|           E-Value            |

----------------------------------

|   b>1    |   b=0    |   b<1    |

----------------------------------

|     0.62|     0.64|     0.62|

----------------------------------
```

Practical No.6

Title : Implement a program for feature extraction in 2D color images (any features like color, texture etc. and to extract features from input image and plot histogram for the features

Program:

STEPS:-

A. Importing an Image:

Importing an image in python is easy. Following code will help you import an image on Python :

```
image = imread(r"C:\Users\Tavish\Desktop\7.jpg")
show_img(image)
```

B. Understanding the underlying data:

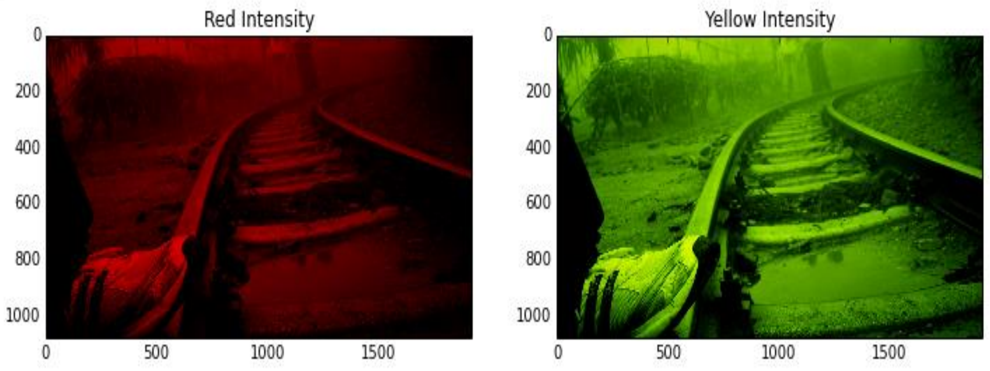
This image has several colors and many pixels.

1. To visualize how this image is stored, think of every pixel as a cell in matrix.
2. Now this cell contains three different intensity information, catering to the color Red, Green and Blue. So a RGB image becomes a 3-D matrix.
3. Each number is the intensity of Red, Blue and Green colors.

```
red, yellow =   image.copy(), image.copy()
```

```
red[:,:,(1,2)] = 0
yellow[:,:,2]=0
```

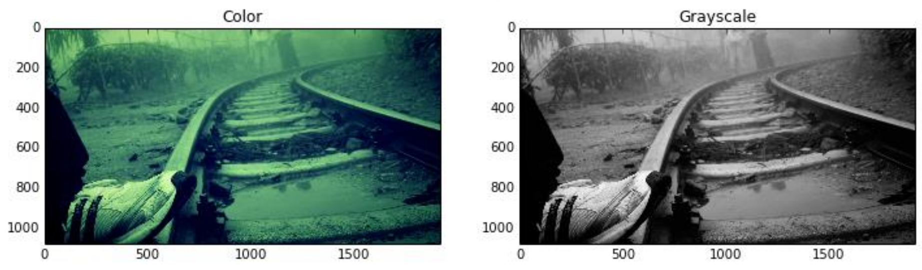
```
show_images(images=[red,yellow], titles=['Red Intensity','Yellow Intensity'])
```



C. Converting Images to a 2-D matrix:-

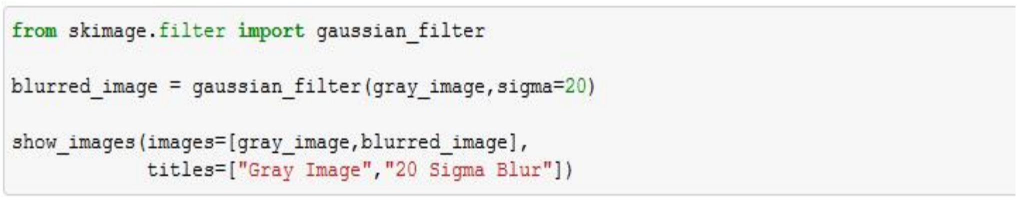
1. Handling the third dimension of images sometimes can be complex and redundant.
2. In feature extraction, it becomes much simpler if we compress the image to a 2-D matrix.
3. This is done by Gray-scaling ,Here is how you convert a RGB image to Gray scale.

```
from skimage.color import rgb2gray

gray_image = rgb2gray(image)
show_images(images=[image,gray_image],
            titles=["Color","Grayscale"])

print "Colored image shape:", image.shape
print "Grayscale image  shape:", gray_image.shape
```

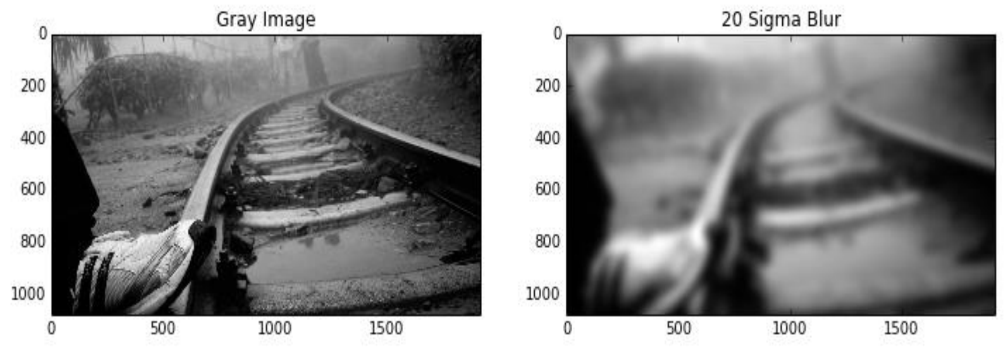


```
Colored image shape: (1080L, 1920L, 3L)
Grayscale image  shape: (1080L, 1920L)
```

Now let’s try to binarize this Gray scale image:-
Blurring an Image:-
Last part of this assignment is more relevant for feature extraction : Blurring of images.



```
from skimage.filter import gaussian_filter

blurred_image = gaussian_filter(gray_image,sigma=20)

show_images(images=[gray_image,blurred_image],
            titles=["Gray Image","20 Sigma Blur"])
```

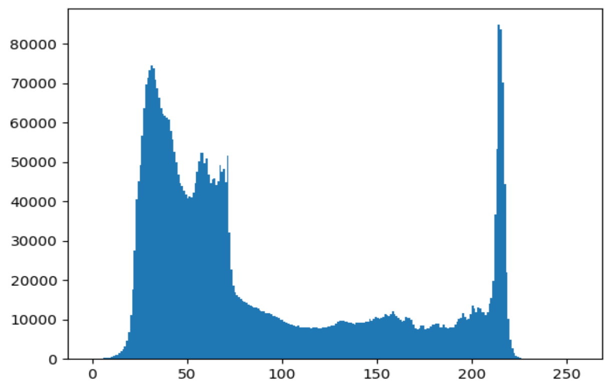


Example:-
image = imread(r"C:\Users\Tavish\Desktop\7.jpg")
show_img(image)
red, yellow =   image.copy(), image.copy()
red[:,:,(1,2)] = 0
yellow[:,:,2]=0
show_images(images=[red,yellow], titles=['Red Intensity','Yellow Intensity'])
 from skimage.color import rgb2gray
gray_image = rgb2gray(image)
show_images(images=[image,gray_image],titles=["Color","Grayscale"])

print "Colored image shape:", image.shape.
print "Grayscale image shape:", gray_image.shape
from skimage.filter
import threshold_otsu
thresh = threshold_otsu(gray_image)
binary = gray_image > thresh
show_images(images=[gray_image,binary_image,binary],titles=["Grayscale","Otsu Binary"])
from skimage.filter import gaussian_filter
blurred_image = gaussian_filter(gray_image,sigma=20)
show_images(images=[gray_image,blurred_image],titles=["Gray Image","20 Sigma Blur"])
Second Part of the Assignment –Plotting the Histogram
histogram is a graphical representation showing how frequently various colour values occur in the image.
Steps:-
Importing image data:-
import matplotlib.pyplot as plt   #importing matplotlib
The image should be used in a PNG file as matplotlib supports only PNG images.
img = plt.imread('flower.png')   #reads image data

Histogram creation using numpy array:-

- To create a histogram of our image data, we use the hist() function.
- plt.hist(n_img.ravel(), bins=256, range=(0.0, 1.0), fc='k', ec='k') #calculating histogram



Histogram Calculation:-

- Here, we use cv2.calcHist()(in-built function in OpenCV) to find the histogram.
- cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])

images : it is the source image of type uint8 or float32 represented as “[img]”.
channels : it is the index of channel for which we calculate histogram.
For grayscale image, its value is [0] and color image, you can pass [0], [1] or [2] to calculate histogram of blue, green or red channel respectively.
mask : mask image. To find histogram of full image, it is given as “None”.

histSize : this represents our BIN count. For full scale, we pass [256].
ranges : this is our RANGE. Normally, it is [0,256].

Example:-

```
# load an image in grayscale mode
img = cv2.imread('ex.jpg',0)
# calculate frequency of pixels in range 0-255
histg = cv2.calcHist([img],[0],None,[256],[0,256])
```

Then, we need to plot histogram to show the characteristics of an image.

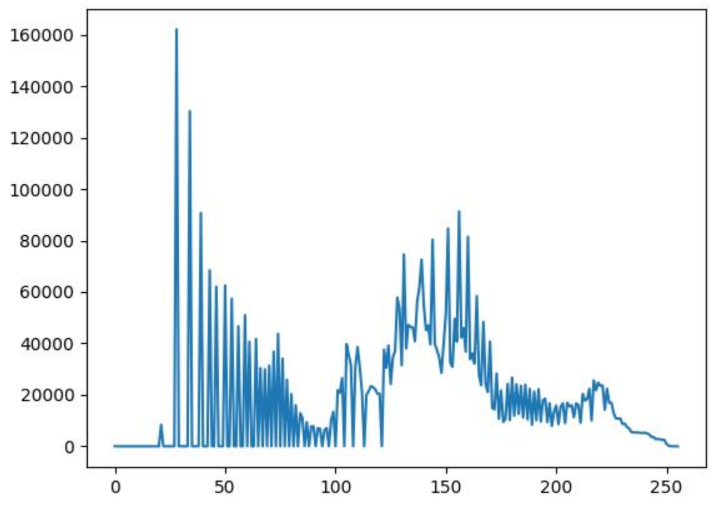
Plotting Histograms

Analysis using Matplotlib:

```
# importing required libraries of opencv
import cv2
# importing library for plotting
from matplotlib import pyplot as plt
# reads an input image
img = cv2.imread('ex.jpg',0)
# find frequency of pixels in range 0-255
histr = cv2.calcHist([img],[0],None,[256],[0,256])
# show the plotting graph of an image
plt.plot(histr)
plt.show()
```

Input:

Output:



160000
140000
120000
100000
80000
60000
40000
20000
0
0
50
100
150
200
250

Practical No.7

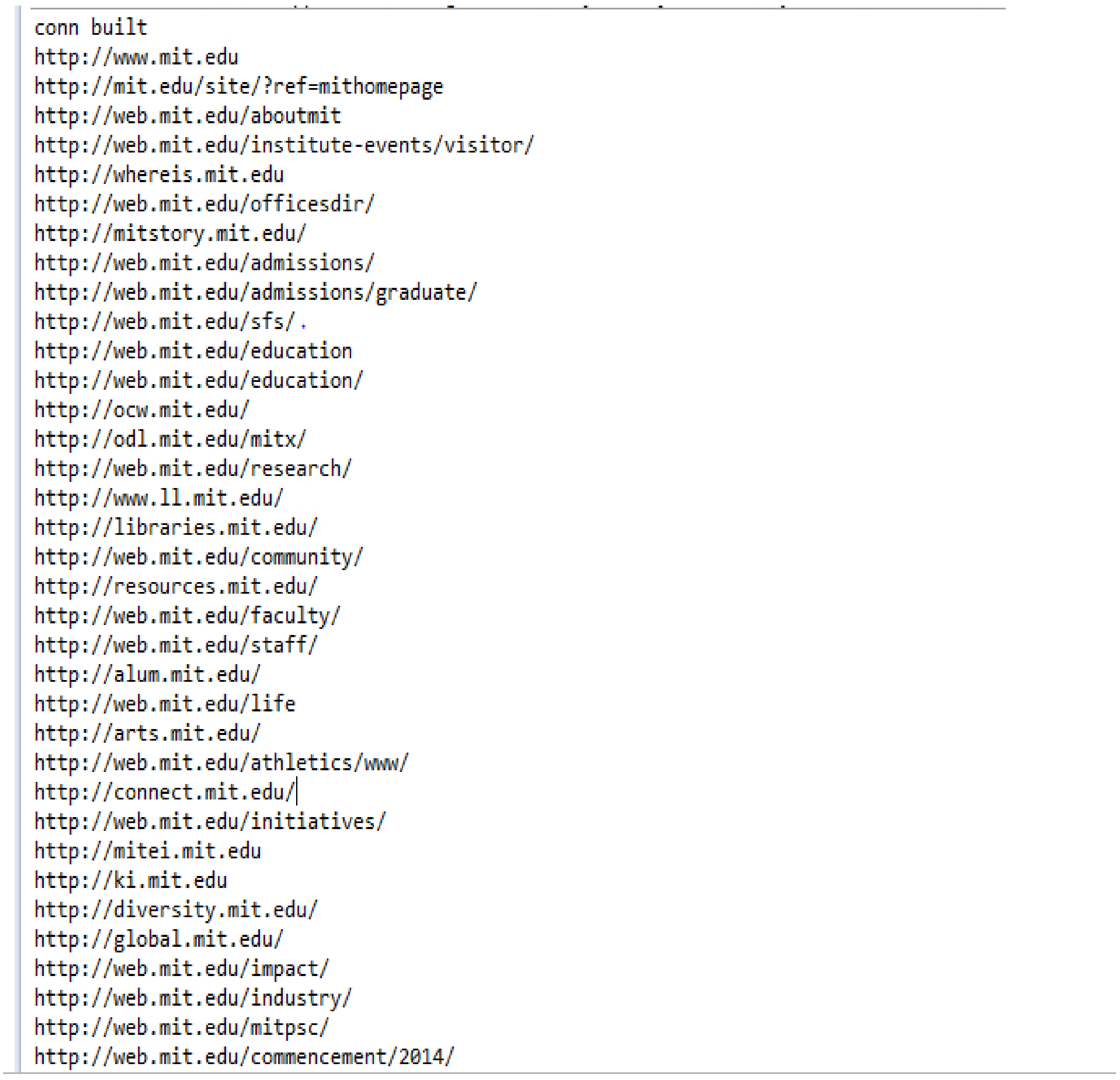
Title : Build the web crawler to pull product information and links from an e-commerce website. (Python).

Program:

```
import java.net.*;

import java.io.*;

public class Crawler{

   public static void main(String[] args) throws Exception{

   String urls[] = new String[1000];

   String url = "https://www.cricbuzz.com/live-cricket-scores/20307/aus-vs-ind-3rd-odi-
india-tour-of-australia-2018-19";

   int i=0,j=0,tmp=0,total=0, MAX = 1000;

   int start=0, end=0;

   String webpage = Web.getWeb(url);

   end = webpage.indexOf("<body");

   for(i=total;i<MAX; i++, total++){

      start = webpage.indexOf("http://", end);

      if(start == -1){

         start = 0;

         end = 0;

         try{

            webpage = Web.getWeb(urls[j++]);

         }catch(Exception e){

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

            System.out.println(urls[j-1]);

            System.out.println("Exception caught \n"+e);

         }

 /*logic to fetch urls out of body of webpage only */

         end = webpage.indexOf("<body");

         if(end == -1)
```

```
     end = start = 0;
                continue;
            }
        }
        end = webpage.indexOf("\"", start);
        tmp = webpage.indexOf("'", start);
        if(tmp < end && tmp != -1){
            end = tmp;
        }
        url = webpage.substring(start, end);
        urls[i] = url;
        System.out.println(urls[i]);
    }
    System.out.println("Total URLS Fetched are " + total);
    }
}
/*This class contains a static function which will fetch the webpage
  of the given url and return as a string */
class Web{
    public static String getWeb(String address)throws Exception{
    String webpage = "";
        String inputLine = "";
        URL url = new URL(address);
        BufferedReader in = new BufferedReader(
        new InputStreamReader(url.openStream()));
        while ((inputLine = in.readLine()) != null)
        webpage += inputLine;
        in.close();
    return webpage; }}
```

Output:

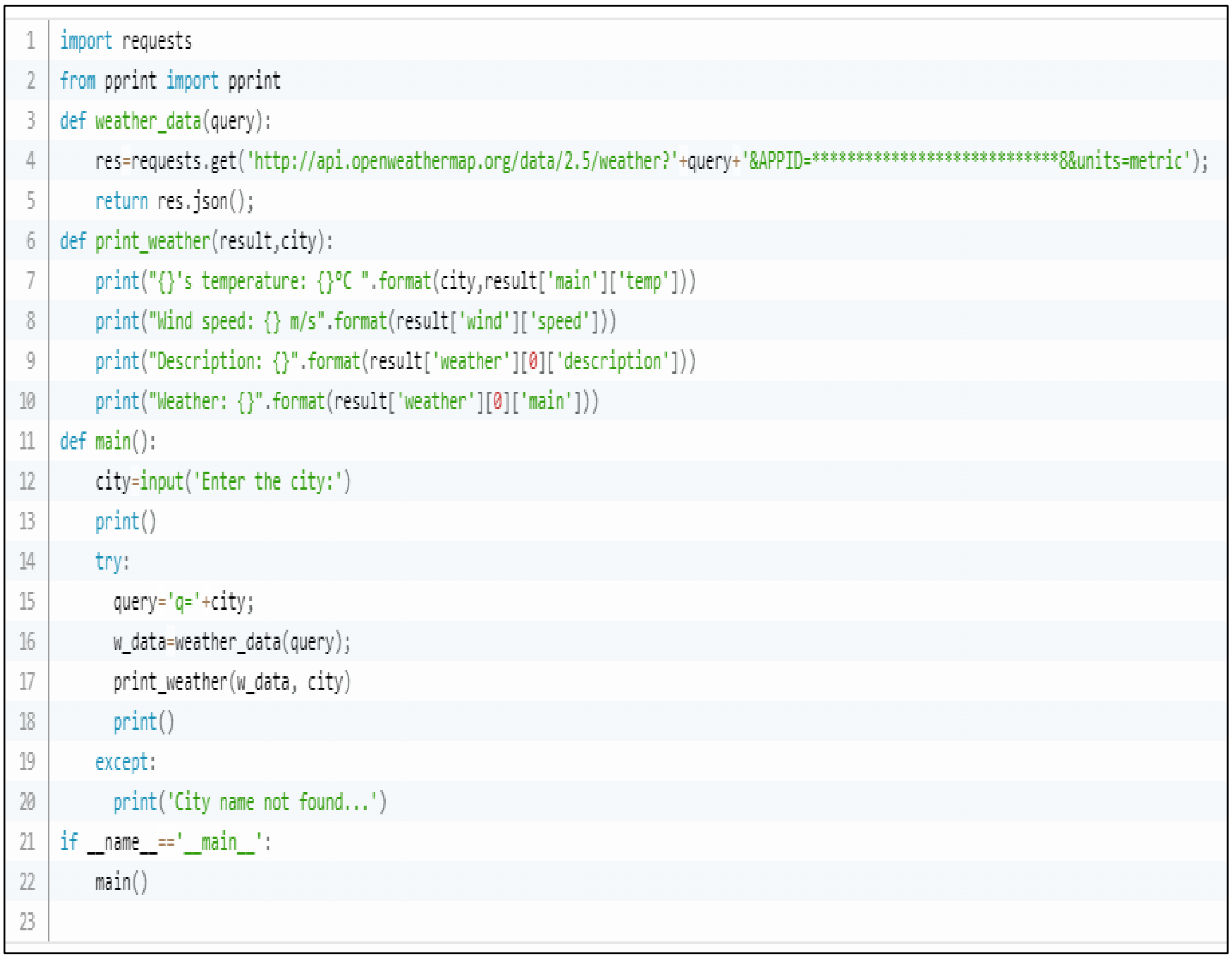


```
conn built
http://www.mit.edu
http://mit.edu/site/?ref=mithomepage
http://web.mit.edu/aboutmit
http://web.mit.edu/institute-events/visitor/
http://whereis.mit.edu
http://web.mit.edu/officesdir/
http://mitstory.mit.edu/
http://web.mit.edu/admissions/
http://web.mit.edu/admissions/graduate/
http://web.mit.edu/sfs/
http://web.mit.edu/education
http://web.mit.edu/education/
http://ocw.mit.edu/
http://odl.mit.edu/mitx/
http://web.mit.edu/research/
http://www.ll.mit.edu/
http://libraries.mit.edu/
http://web.mit.edu/community/
http://resources.mit.edu/
http://web.mit.edu/faculty/
http://web.mit.edu/staff/
http://alum.mit.edu/
http://web.mit.edu/life
http://arts.mit.edu/
http://web.mit.edu/athletics/www/
http://connect.mit.edu/
http://web.mit.edu/initiatives/
http://mitei.mit.edu
http://ki.mit.edu
http://diversity.mit.edu/
http://global.mit.edu/
http://web.mit.edu/impact/
http://web.mit.edu/industry/
http://web.mit.edu/mitpsc/
http://web.mit.edu/commencement/2014/
```

Practical No.8

Title : Write a program to find the live weather report (temperature, wind speed, description, and weather) of a given city. (Python).

Program:



```python
import requests
from pprint import pprint
def weather_data(query):
    res=requests.get('http://api.openweathermap.org/data/2.5/weather?'+query+'&APPID=****************************8&units=metric');
    return res.json();
def print_weather(result,city):
    print("{}'s temperature: {}°C ".format(city,result['main']['temp']))
    print("Wind speed: {} m/s".format(result['wind']['speed']))
    print("Description: {}".format(result['weather'][0]['description']))
    print("Weather: {}".format(result['weather'][0]['main']))
def main():
    city=input('Enter the city:')
    print()
    try:
      query='q='+city;
      w_data=weather_data(query);
      print_weather(w_data, city)
      print()
    except:
      print('City name not found...')
if __name__=='__main__':
    main()

```

Output:

Sample Output:

```
Enter the city: Brazil

Brazil's temperature: 16.45°C

Wind speed: 2.1 m/s

Description: clear sky

Weather: Clear
```