Data je txt datoteka koja sadrži sledeće brojeve:

A white background with black text

Description automatically generated

Napisati program koji će da učita sadržaj ovog tekstualnog fajla, a zatim ispiše te brojeve kao realne brojeve.

namespace ConsoleApp8

{

internal class Program

{

static void Main(string[] args)

{

string[] niz = { "oop", "1236", "vts", "C#" };

string textFilePath = @"c:\Users\Sanjam\Documents\oktobar25.txt";

File.WriteAllLines(textFilePath, niz);

foreach (string s in File.ReadLines(textFilePath))

{

Console.WriteLine(s);

}

string path = @"c:\tmp\brojevi.txt";

string[] readText = File.ReadAllLines(path);

List<double> brojevi = new List<double>();

foreach (string line in readText)

{

//if (line.Length == 0)

// break;

if (line.Contains(", "))

{

string[] split = line.Split(", ");

foreach (string s in split)

{

double d = double.Parse(s);

brojevi.Add(d);

}

continue;

}

double d1 = double.Parse(line);

brojevi.Add(d1);

}

foreach (double b in brojevi)

{

Console.Write(b + ", ");

}

Console.WriteLine();

// ucitavanje txt fajla i prebrojavanje koliko puta se koja rec javlja

// za ovaj deo zadatka napraviti proizvoljan txt fajl, npr. prekopirati

// text sa wikipedije za programski jezik c#

string path1 = @"c:\tmp\text.txt";

List<string> lines = new List<string>();

lines = File.ReadAllLines(path1).ToList();

//ReadAllLines(path) vraca string[], zato nam treba ToList()

foreach (string s in lines)

{

Console.WriteLine(s);

}

string[] separator = { ".", ",", "!", "?", "\n", "\r", " " };

foreach (string s in lines)

{

string[] words = s.Split(separator, StringSplitOptions.RemoveEmptyEntries);

foreach (string w in words)

{

Console.WriteLine(w);

}

}

//prebrojavanje koliko puta se koja rec javlja

//prebrojati koliko ima jedinstvenih reci

Dictionary<string, int> wordCount = new Dictionary<string, int>();

HashSet<string> uniqueWords = new HashSet<string>();

foreach (string s in lines)

{

string[] words = s.Split(separator, StringSplitOptions.RemoveEmptyEntries);

foreach (string w in words)

{

if (uniqueWords.Add(w)) //da li je rec jedinstvena, ako jeste ide u HashSet

{

wordCount.Add(w, 1);

}

else

{

wordCount[w]++;

}

}

}

foreach (KeyValuePair<string, int> word in wordCount)

{

Console.WriteLine("rec: {0}, brojPonavljanja: {1}", word.Key, word.Value);

}

//ispisivanje reci koje su jedinstvene

Console.WriteLine("Jedinstvene reci u text:");

foreach (string s in uniqueWords)

{

Console.WriteLine(s);

}

Console.WriteLine("Broj jedinstvenih reci je: " + uniqueWords.Count);

}

}

}

**Crvenkapica**

Bara (voda) i suvi delovi nalaze se na šumskoj stazi. Zemlju (suve delove staze) simboliše broj 1, a vodu predstavlja broj 2.

Napravimo jednu listu u kojoj su navedene nasumične vrednosti 1 i 2 tako da je veća verovatnoća da se u ovoj listi nađe voda nego zemljište (na primer, 70% voda 30% zemljište)! U listi (tj.na stazi) prvi i poslednji element mora da bude 1, odnosno zemljište. Prikažite listu na ekranu tako da zemlju predstavlja # (hashmark), a voda je \_ (donja crta).

Na jednom kraju šumske staze nalazi se Crvenkapica koja želi da stigne do bake koja se nalazi na kraju ove šumske staze. Crvenkapica može da preskoči “n” bara u nizu (vrednost za “n”, broj bara koje Crvenkapica može da preskoči, unosi se sa tastature). Napišite program koji generiše stazu (listu) u datom odnosu vode i zemljišta (npr. 70% i 30%), prikažite stazu na ekranu, a zatim odredite da li Crvenkapica može da pređe ovu stazu, a da joj noge ostanu suve.

12122222112221 ->#\_#\_ \_ \_ \_ \_#\_ \_ \_#\_ \_#

using System;

using System.Collections.Generic;

namespace Crvenkapica

{

class Program

{

static void Main(string[] args)

{

List<int> staza = new List<int>();

staza.Add(1); //pocetak staze, mora da bude 1, tj.zemljiste

Console.WriteLine("Unesite duzinu staze:");

int duzinaStaze = int.Parse(Console.ReadLine());

Random rnd = new Random();

for (int i = 0; i < duzinaStaze - 2; i++)

{

int num = rnd.Next(1, 101); //1,2...100

if (num <= 30)

staza.Add(1); // 30% treba da bude zemljiste

else

staza.Add(2); //70% treba da bude voda

}

staza.Add(1); //na kraju puta treba da bude zemljiste

foreach (int i in staza)

{

if (i == 1)

Console.Write("#");

else

Console.Write("\_ ");

}

//odredjivanje najduzeg vodenog dela

//npr staza je 12122222112221

Console.WriteLine("\nNajduzi niz bara:" + najduzaVoda(staza));

int maxBrojBara = 0;

Console.Write("Unesite koliko bara odjednom moze Crvenkapica da preskoci:");

maxBrojBara = int.Parse(Console.ReadLine());

if (maxBrojBara >= najduzaVoda(staza))

{

Console.WriteLine("\nCrvenkapica moze da stigne do bake!");

}

else

{

Console.WriteLine("\nCrvenkapica ne moze da stigne do bake!");

}

//

}

static int najduzaVoda(List<int> lista)

{

int najduziNizBara = 0;

int duzina = 0;

foreach (int i in lista)

{

if (i == 2)

duzina++;

else

duzina = 0;

if (duzina > najduziNizBara)

najduziNizBara = duzina;

}

return najduziNizBara;

}

}

}