Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 1

дисципліни “ Дискретна математика ”

на тему

«ВИЗНАЧЕННЯ УМОВНО НЕЗАЛЕЖНИХ ПОДІЇ»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Гермак В.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

**Лабораторная работа №1**

**Тема**: Определение условно независимых события.

**Цель**: Овладеть методикой определения вероятности произвольного события по устойчивости накопленной частоты.

**ЗАДАНИЕ**

1. Определить вероятность появления буквы, что порядковый  номер, который совпадает с порядковым номером вашего фамилии в журнале группы:

    а) вручную для текста на укр.мовi размером 5Кб.

    б) программно для текста на укр.мовi размером> 50 Кб.

2. Определить вероятность появления слога, который начинается   буквой пункта 1 и условную (байесивську) зависимость между ними.

3.Построить таблицу вероятностей всех букв алфавита**.**

**Вариант 37**

Текст

Весна - це та пора року, який усі чекають, утомившись від зими.

Весну часто називають довгоочікуваною.

Перші ознаки весни, буває, проявляються ще взимку, у лютому:ледве посильнее посвітить сонечко - і починають танути бурульки, дзенькає весела капель, нагадуючи про наближення весни. Після цього ще можуть ударити морози, може піти сніг, але всі розуміють: весна от-наступить, порадує своїм теплом От і прийшла весна. Сонце починає з'являтися на небі всі частіше, небо робиться більше яскравим, синій^-синім-густо-синім, щосили тане сніг і біжать струмки.

Земля ще холодна, занадто сира й гола, а через залишки снігу пробиваються найперші квіти, які так і називають: проліски

Весна прийшла! У лісі починається пожвавлення: прокидаються від зимової спячки звірі, вертаються після зимівлі птаха. Усі зайняті парованием і облаштованістю затишних гніздечок і норок, усе готуються до виведення потомства, для цього потрібні надійні будинки. Настання весни - це пробудження природи, початок активного життя природи. Швидко набухають і лопаються бруньки, з яких з'являються ніжно-зелені листочки, а потім вони ростуть із неуявною швидкістю. За тиждень дерева повністю покривається листям, що ніжно шарудить на весняному вітерці. Усюди відчувається життя

порция 1

Весна - це та пора року, який усі чекають, утомившись від зими.Весну часто називають довгоочікуваною.Перші ознаки весни,

======================================================================

порция 2

буває, проявляються ще взимку, у лютому:ледве посильнее посвітить сонечко - і починають танути бурульки, дзенькає весела капель,

======================================================================

порция 3

нагадуючи про наближення весни. Після цього ще можуть ударити морози, може піти сніг, але всі розуміють: весна от-наступить,

======================================================================

порция 4

порадує своїм теплом От і прийшла весна. Сонце починає з'являтися на небі всі частіше, небо робиться більше яскравим

======================================================================

порция 5

мсиній^-синім-густо-синім, щосили тане сніг і біжать струмки. Земля ще холодна, занадто сира й гола, а через залишки

======================================================================

порция 6

снігу пробиваються найперші квіти, які так і називають: проліски Весна прийшла! У лісі починається пожвавлення: прокидаються

======================================================================

порция 7

від зимової спячки звірі, вертаються після зимівлі птаха. Усі зайняті парованием і облаштованістю затишних гніздечок і норок,

======================================================================

порция 8

усе готуються до виведення потомства, для цього потрібні надійні будинки. Настання весни - це пробудження природи, початок

======================================================================

порция 9

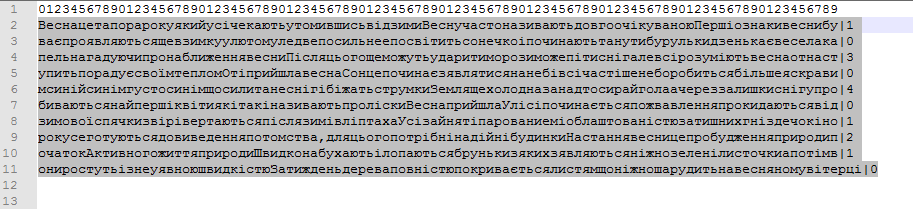
Активного життя природи. Швидко набухають і лопаються бруньки, з яких з'являються ніжно-зелені листочки, а потім вони

======================================================================

порция 10

ростуть із неуявною швидкістю. За тиждень дерева повністю покривається листям, що ніжно шарудить на весняному вітерці. Усюди відчувається життя

======================================================================



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № порции | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Кол. сим «Г» и «г» | 1 | 0 | 3 | 0 | 4 | 0 | 1 | 2 | 1 | 0 |
| Накопленная частота | 0,01 | 0,005 | 0,0133 | 0,01 | 0,016 | 0,0133 | 0,0128 | 0,01375 | 0,0133 | 0,012 |

Средне-арифметическое значение - 0,011948

б) программно для текста на укр.мовi размером> 50 Кб.

#ifndef ANALYSIS\_H

#define ANALYSIS\_H

#include <string>

#include <vector>

using namespace std;

class Analysis

{

private:

vector<string> dictionary;

vector <char> symbols;

vector <string> vs;

char userChoose;

double analisis\_result;

int totalSymbolCount;

int part\_size;

int part\_count;

double totalVovel;

double counters;

double syllable;

double eps;

void LoadWord(string path); // загрузка текста в словарь

void LoadSymbol(string path); // загрузка текста в словарь

void GetPartSize(); // определение размера порции

void Counting(); // заполнение таблицы

void CountByFormula(double \*countInPart);

void MiddleVal(double \*countInPart);

void GetSymbolFromUser();

double GetEpsilon();

void CountSyllableWithLetter();

int CountVovelInWord(string word);

bool IsVovel(char symbol);

void Check(string word);

public:

Analysis();

void Analys(string path); // анализ текста

void Report(); // отчет

};

#endif // ANALYSIS\_H

#include "analysis.h"

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <iomanip>

#include <cmath>

#include <conio.h>

string vovel = "аоиуяюэыёй";

#define VOVEL\_COUNT 10

using namespace std;

Analysis::Analysis()

{

totalSymbolCount=0;

part\_size =0;

analisis\_result=0;

part\_size=0;

part\_count=0;

totalVovel=0;

counters=0;

syllable=0;

eps =0;

}

// анализ текста

void Analysis::Analys(string path){

LoadSymbol(path);

LoadWord(path);

GetPartSize();

Counting();

CountSyllableWithLetter();

}

// загрузка текста

void Analysis:: LoadSymbol(string path){

std::ifstream text;

text.open(path);

if(!text.is\_open()){

cout << "file couldn`t open " << std::endl;

exit(1);

}

char c;

while(!text.eof()){

c = text.get();

symbols.push\_back(c);

if(c != ' '){

totalSymbolCount++;

}

}

text.close();

}

// загрузка текста

void Analysis:: LoadWord(string path){

std::ifstream text;

text.open(path);

if(!text.is\_open()){

cout << "file couldn`t open " << std::endl;

exit(1);

}

std::string temp;

while(!text.eof()){

std::getline(text, temp,' ');

dictionary.push\_back(temp);

}

text.close();

}

void Analysis::GetSymbolFromUser(){

cout << " Введите символ : " << endl;

userChoose = getch();

cout << " Введенный символ : " << userChoose << endl;

eps = GetEpsilon();

}

double Analysis::GetEpsilon(){

double temp =0;

cout << " Введите относительность : ";

cin >> temp;

return temp;

}

// определение размера порции

void Analysis::GetPartSize(){

if (totalSymbolCount !=0 ){

if (totalSymbolCount > 0 && totalSymbolCount <= 1000){

part\_size = 100;

} else if (totalSymbolCount > 1000 && totalSymbolCount <= 10000){

part\_size = 200;

} else if (totalSymbolCount > 10000 && totalSymbolCount <= 50000){

part\_size = 300;

} else if (totalSymbolCount > 50000 && totalSymbolCount <=100000){

part\_size = 500;

} else if (totalSymbolCount > 100000 ){

part\_size = 1000;

}

part\_count = (totalSymbolCount/part\_size);

if (part\_count \* part\_size < totalSymbolCount)

part\_count++;

}

}

// заполнение таблицы

void Analysis::Counting(){

GetSymbolFromUser();

double countInPart[part\_count];

double iter =0;

for(int i=0; i <part\_count; i++){

iter =0;

for(int j = i\*part\_size; j < part\_size\*(i+1); j++){

if(j < totalSymbolCount){

if(symbols[j] == userChoose)

iter ++;

}

}

countInPart[i] = iter;

}

CountByFormula(countInPart);

MiddleVal(countInPart);

}

// подсчет по формуле

void Analysis::CountByFormula(double \*countInPart){

double temp;

double iter = 0;

for(int i=0; i < part\_count; i++){

iter +=countInPart[i];

temp = iter /((i+1)\* part\_size);

countInPart[i] = temp;

}

}

// среднее значение

void Analysis::MiddleVal(double \*countInPart){

double temp=0;

for (int i= 0; i < part\_count-1; i++){

if (abs(countInPart[i+1]-countInPart[i]) > eps\*10){

cout << "в файле находится большое скопление символа\' " <<userChoose

<< " \' в порции " << i+1 << "приблизительно в районе " << (i+1)\*part\_size << endl;

exit(1);

}

}

for (int i= part\_count-1; i>part\_count-6; i--){

cout << " "<<fixed << setprecision(5) << countInPart[i]<< endl;

if (abs(countInPart[i]-countInPart[i-1])< 0.01)

temp +=countInPart[i];

}

analisis\_result = temp/5;

}

// проверка слов и подсчет слогов

void Analysis::Check(string word){

int iter =0; // счетчик

string str; // хранение одного слова

int vovel\_count = CountVovelInWord(word);

totalVovel += vovel\_count;

if(vovel\_count != 0){ // если количество гласных не равно нулю

for(int i=0; i < int( word.length()); i++){ // пройтись по слову

if(IsVovel(word[i])){ // находим гласную

str+=word[i]; // конкатенируем ее

vs.push\_back(str);

iter++;

str.clear();

}else{

str += word[i];

}

}

string temp = vs.back();

temp += str;

vs.erase(vs.end());

vs.push\_back(temp);

for (int i=0; i < vovel\_count; i++){

str =vs.back();

vs.erase(vs.end());

if (str[0] == userChoose){

counters++;

}

}

vs.clear();

}

}

// подсчет гласных в слове

int Analysis::CountVovelInWord(string word){

int temp =0; // временная переменная

for (int i =0; i< int(word.size()); i++){ // пройтись по всему слову

if (IsVovel(word[i])){ // если слово гласное

temp++; // увеличить счетчик

}

}

return temp;

}

// проверка является ли буква гласной

bool Analysis::IsVovel(char symbol){

for (int i =0; i < VOVEL\_COUNT; i++){ // пройтись по всем гласным

if(symbol == vovel[i]) // если символ находися в массиве гласных

return true; // вернуть true

}

return false;

}

// подсчет вероятности

void Analysis::CountSyllableWithLetter(){

for(int i =0; i< int(dictionary.size()); i++){

Check(dictionary[i]);

}

syllable = counters/totalVovel;

}

// отчет

void Analysis::Report(){

cout << fixed << setprecision(5)

<< " Результат анализа : " << analisis\_result << endl;

cout << " Размер порции : " << part\_size << endl;

cout << " Количество порций : " << part\_count << endl;

cout << " Количество символов : " << totalSymbolCount << endl;

cout << " Вероятность в слоге : " << syllable << endl;

cout << " Общее количество слогов : " << totalVovel << endl;

cout << " Кол. слогов по условию : " << counters << endl;

}

#include <iostream>

#include "analysis.h"

using namespace std;

int main()

{

string path = "D:\\1.txt";

Analysis \*object= new Analysis();

object->Analys(path);

object->Report();

return 0;

}