// BibleCode.cpp : Defines the entry point for the console application.

//developed by MK

#include "stdafx.h"

#include <math.h>

#include <string>

#include <windows.h>

#include <iostream>

#include <fstream>

#include <sstream>

#include <fcntl.h> // for \_setmode

#include <io.h>

using std::ofstream; //using std::iostream;

using std::cout; using std::cerr;

using std::endl; using std::string;

using std::ifstream; using std::ostringstream;

using std::wstring;

//całkowita ilość liter w Torze 304805

static int litery = 650;

//punkt startowy poszukiwań

unsigned int punktStartowy = 0;

//co ile występują litery

unsigned int liczbaPrzeskokow = 0;

//max liczba przeskoków

unsigned int maxLiczbaPrzeskokow = 0;

//!szukana fraza początkowa //TD! #1

//wstring fraza = L"מתוכננת";

string fraza = "2021";

//n->co ile występuje

//ps->punkt startowy

unsigned int ps, n = 0;

//tora do przeszukania

string Tora = "";

//dwukrotne przepisane

string ToraFix = "";

//FUNKCJA wczytaj tekst

string readFileIntoString(const string& path)

{

ifstream input\_file(path);

if (!input\_file.is\_open())

{

cerr << "Could not open the file - '"

<< path << "'" << endl;

exit(EXIT\_FAILURE); //zabezpiecznie

}

return string((std::istreambuf\_iterator<char>(input\_file)), std::istreambuf\_iterator<char>());

}

//FUNKCJA przepisująca

void przygotowawcza(const string& path, const string& path2)

{

//pobierz text z pliku

ifstream input\_file(path);

string ToraRaw = string((std::istreambuf\_iterator<char>(input\_file)), std::istreambuf\_iterator<char>());

string ToraRed,ToraFin;

//upSizing stringow

ToraRed.resize(ToraRaw.length()\*2);

ToraFin.resize(ToraRaw.length()\*2);

//usuwanie i redagowanie

for (int i = 0; i < ToraRaw.length(); i++) //TD! #2

{

//usuwamy zbedne znaki

if (ToraRaw[i] == ',' || ToraRaw[i] == '.' || ToraRaw[i] == ' ')

{

//nadpisanie

for (int j = i; j < (ToraRaw.length()-1); j++)

ToraRaw[j] = ToraRaw[j + 1];

}

}

//przygotowanie pliku

ofstream ofs;

ofs.open(path2);

//przepisanie

for (int i = 0; i < (ToraRaw.length()\*2); i++) //dwukrotny

{

//wyjście poza rozmiar

if (i == ToraRaw.length())

n = 0;

//przepisanie

ToraFin[i] = ToraRaw[n];

n++;

}

//zapis

ofs << ToraFin;

ofs.close();

//ui

cout << endl << "zapisano pomyslnie" << endl;

cout << "Liczba znakow w Finalnej Torze: " << ToraFin.length() << endl;

}

//FUNKCJA startowa

int startowa(string szukanaFraza)

{

//czy znaleziono fraze?

bool znaleziono = false;

//wartość przeskoku

int przeskok = 0;

//litery:długość frazy => tyle razy można przeskoczyć

maxLiczbaPrzeskokow = floor((litery / (szukanaFraza.length())));

for (n = maxLiczbaPrzeskokow; n > 0; n--) //zakres poszukiwań

{

for (ps = 0; ps < litery; ps++) //zmiana pktStartowego

{

for (int i = 0; i < szukanaFraza.length(); i++)

{

//zabezpieczenie przed wyjsciem

przeskok = i\*n;

if (przeskok > litery)

przeskok -= (litery-1);

//Tora[punktStartowy+(przeskok)]

if (Tora[ps+(przeskok)] == fraza[i])

znaleziono = true;

else

{

znaleziono = false;

break;

}

}

if (znaleziono == true) break;

}

if (znaleziono == true) break;

}

//zapisz

liczbaPrzeskokow = n;

punktStartowy = ps;

//zwróć 1 jeśli znaleziono

if (znaleziono == true)

return 1;

else

return 0;

}

//główna FUNKCJA

int main()

{

//redakcja oryginalnej Tory

przygotowawcza("TorahRaw.txt", "TorahFin.txt");

//wczytaj tore do Stringa

Tora = readFileIntoString("Torah.txt");

//szukanie frazy, inicjalizacja

cout << "..szukanie frazy.." << endl << endl; //ui

if (startowa(fraza) != 1)

{

cout << "Nie znaleziono szukanej frazy: "+ fraza << endl; //ui

getchar(); getchar();

return 0; //zakończ

}

else

{

cout << "Fraza: " + fraza + " wystepuje w tekscie"<< endl; //ui

}

//właściwa kombinatoryka //TD! #3

//test wczytywania

cout << "test" << endl;

for (int i = 0; i < litery /\*fraza.length()\*/; i++)

cout << Tora[i];

cout << endl << "test" << endl;

getchar(); getchar();

return 0;

}