

Uniwersytet Przyrodniczo-Humanistyczny w Siedlcach

Kierunek Informatyka

QUIZ

Sebastian Rosochacki

Praca wykonana pod kierunkiem

dr Anny Wawrzyńczak-Szaban

Siedlce, 2020

Spis treści

Cel projektu	3
Funkcjonalności	3
Narzędzia	3
Znane błędy	3
Schemat ogólny	4
Struktura aplikacji	5
Struktura bazy danych.....	5
klucz_odpowiedzi.....	5
zadanie	5
odpowiedz	5
init_test	6
start_test	7
display_zadanie	8
maszyna_losujaca	9
load_pytania	10
Instrukcja użycia programu	11
Pierwsze uruchomienie.....	11
Tryb egzaminu	13
Kod programu.....	16
main.cpp.....	16
struct.h	17
test.cpp.....	18
base.cpp	21

Cel projektu

Program ma za zadanie wczytać pytania z pliku tekstowego (format dowolny), wyświetlić je w losowej kolejności (wraz z losową kolejnością odpowiedzi) oraz zebrać odpowiedzi użytkownika. Po zakończonym quizie, program wyświetla liczbę zdobytych punktów (również procentowo) oraz listę pytań, na które odpowiedziano źle (wraz z zaznaczeniem odpowiedzi poprawnej i tej, której udzielił użytkownik). Format graficzny dowolny. Pytania nie mogą się powtarzać.

Funkcjonalności

- Przeprowadzanie testu z losowymi lub przygotowanymi wcześniej pytaniami
- Sprawdzanie odpowiedzi po teście, wyświetlanie wyniku, także w postaci procentowej
- Dodawanie, usuwanie, edytowanie pytań
- Rozpoczęcie testu z przygotowaną wcześniej listą pytań

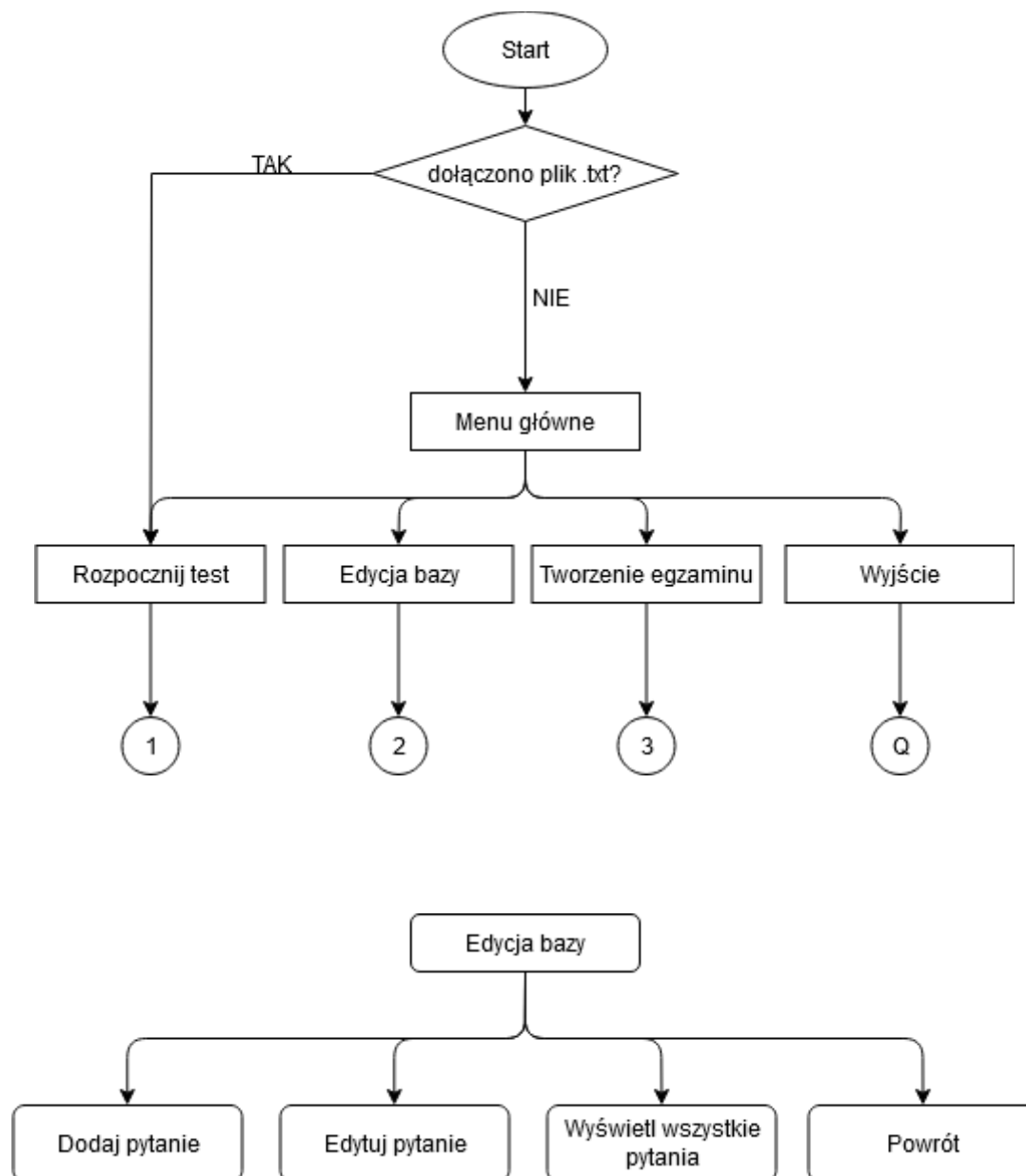
Narzędzia

Program został napisany w języku C++, wersja C++17. Wszystkie wymagane biblioteki są statycznie dołączone do pliku wykonywalnego - .exe jest w pełni samodzielny. Program został napisany dla systemu Windows, jednakże jego kompilacja pod system Linux lub macOS jest możliwa z użyciem przygotowanego pliku Makefile. Edycje kodu dokonywałem w CLion IDE. W projekcie został również Git, do wersjonowania i zarządzania kodem, jak i CMake, do kompilowania kodu oraz generacji projektów CodeBlocks i plików Makefile, aby program mógł być skompilowany na każdą platformę.

Znane błędy

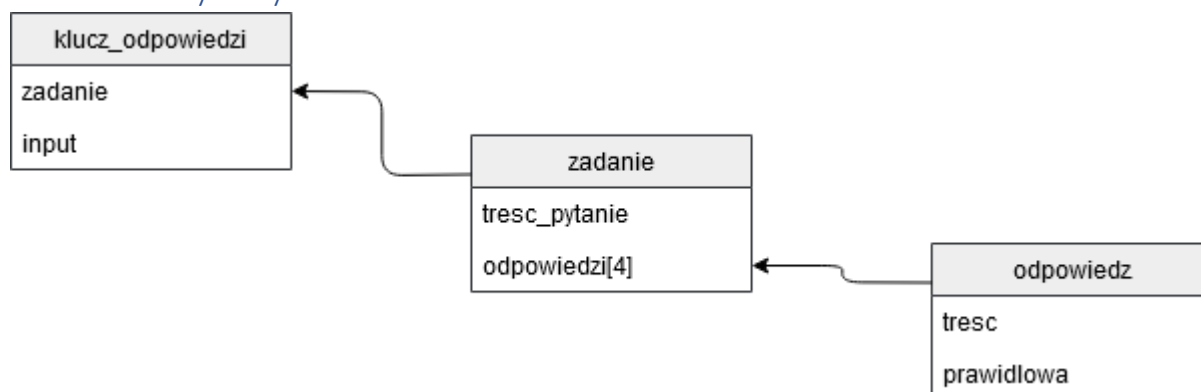
- Program nie będzie możliwy do skompilowania w obecnym oficjalnym wydaniu MinGW. Jest to spowodowane nieaktualnymi wersjami bibliotek, w tym przypadku <algorithm>, co powoduje błędy kompilacji przy samym #include <algorithm>. Aby obejść ten problem, należy zmienić kompilator na inny lub pochodny do MinGW, który posiada nowsze wersje GCC – w moim przypadku był to MSYS2.
- Program nie obsługuje znaków polskich – wczytywanie takowych z pliku bazy skutkuje „krzakami” lub nieoczekiwanym zamknięciem programu.

Schemat ogólny



Struktura aplikacji

Struktura bazy danych



Aplikacja wykorzystuje 3 tabele:

- klucz_odpowiedzi – zawiera odpowiedzi użytkownika na poszczególne zadania
- zadanie – zawiera informacje o pytaniu, tj. treść i odpowiedzi
- odpowiedz – zawiera treść odpowiedzi oraz czy jest prawidłowa

W kodzie programu struktury zostały zdefiniowane w pliku struct.h. Pytania są zapisywane w pliku .txt w formacie: pytanie;~odp1;odp2;odp3;odp4, gdzie tylda(~) oznacza odpowiedź prawidłową.

klucz_odpowiedzi

Tabela zawiera informacje o odpowiedzi użytkownika na zadanie. Zawiera w sobie tabelę *zadanie*.

Nazwa pola	Typ pola	Opis
zadanie	struct zadanie – tabela <i>zadanie</i>	Zadanie, na które odpowiedział użytkownik
input	int – liczba całkowita	Odpowiedz użytkownika na zadanie

zadanie

Tabela zawierająca szczegóły pytania. Zawiera w sobie tabelę *odpowiedz*.

Nazwa pola	Typ pola	Opis
tresc_pytanie	string – ciąg znaków	Treść pytania
odpowiedzi	struct odpowiedz[4] – tablica tabel <i>odpowiedz</i>	Odpowiedzi na pytanie

odpowiedz

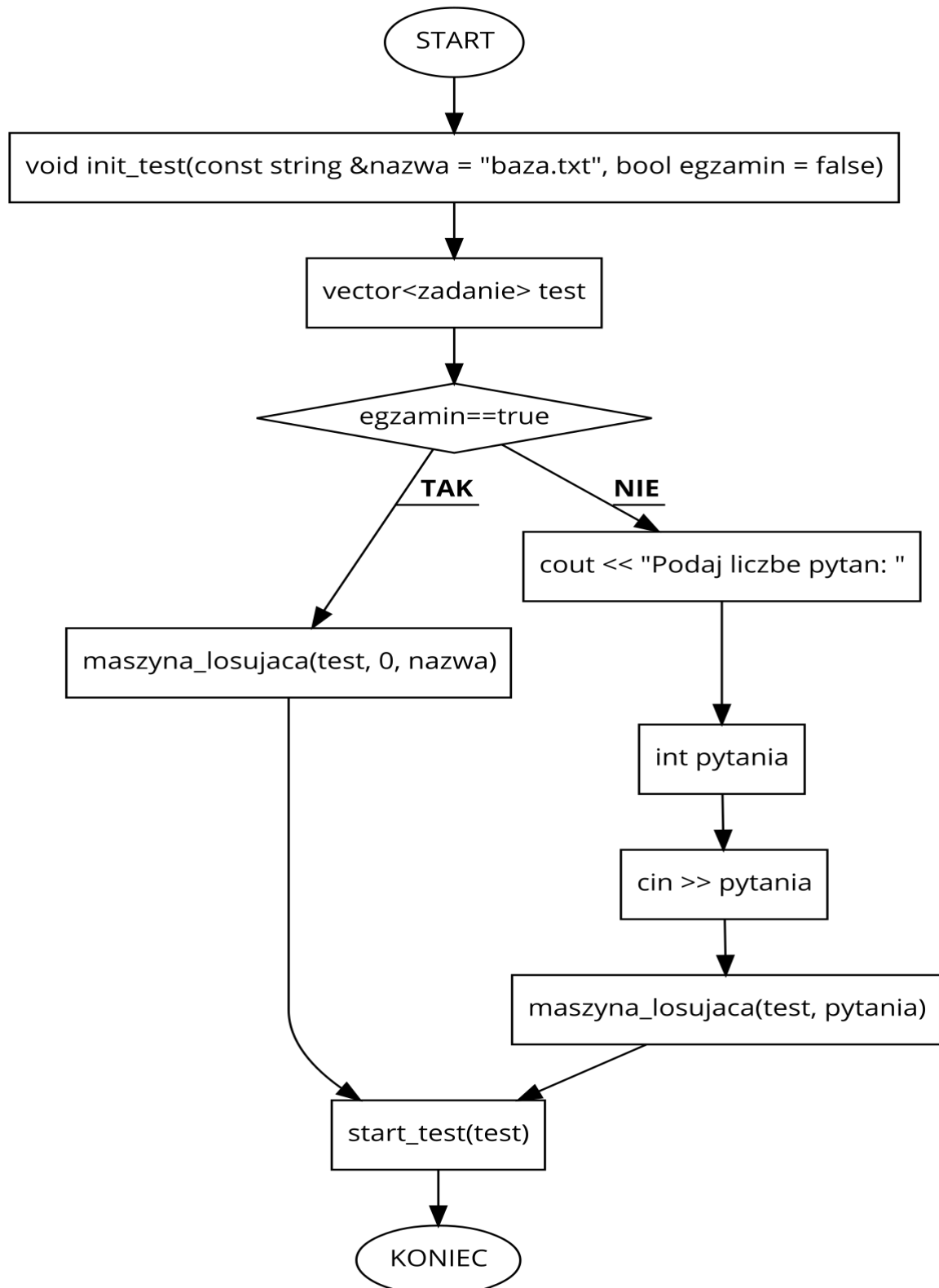
Tabela zawierająca odpowiedzi na pytanie.

Nazwa pola	Typ pola	Opis
tresc	string – ciąg znaków	Treść odpowiedzi
prawidlowa	bool – typ logiczny	Określa, czy odpowiedź jest prawidłowa

init_test

```
void init_test(const string &nazwa = "baza.txt", bool egzamin = false)
```

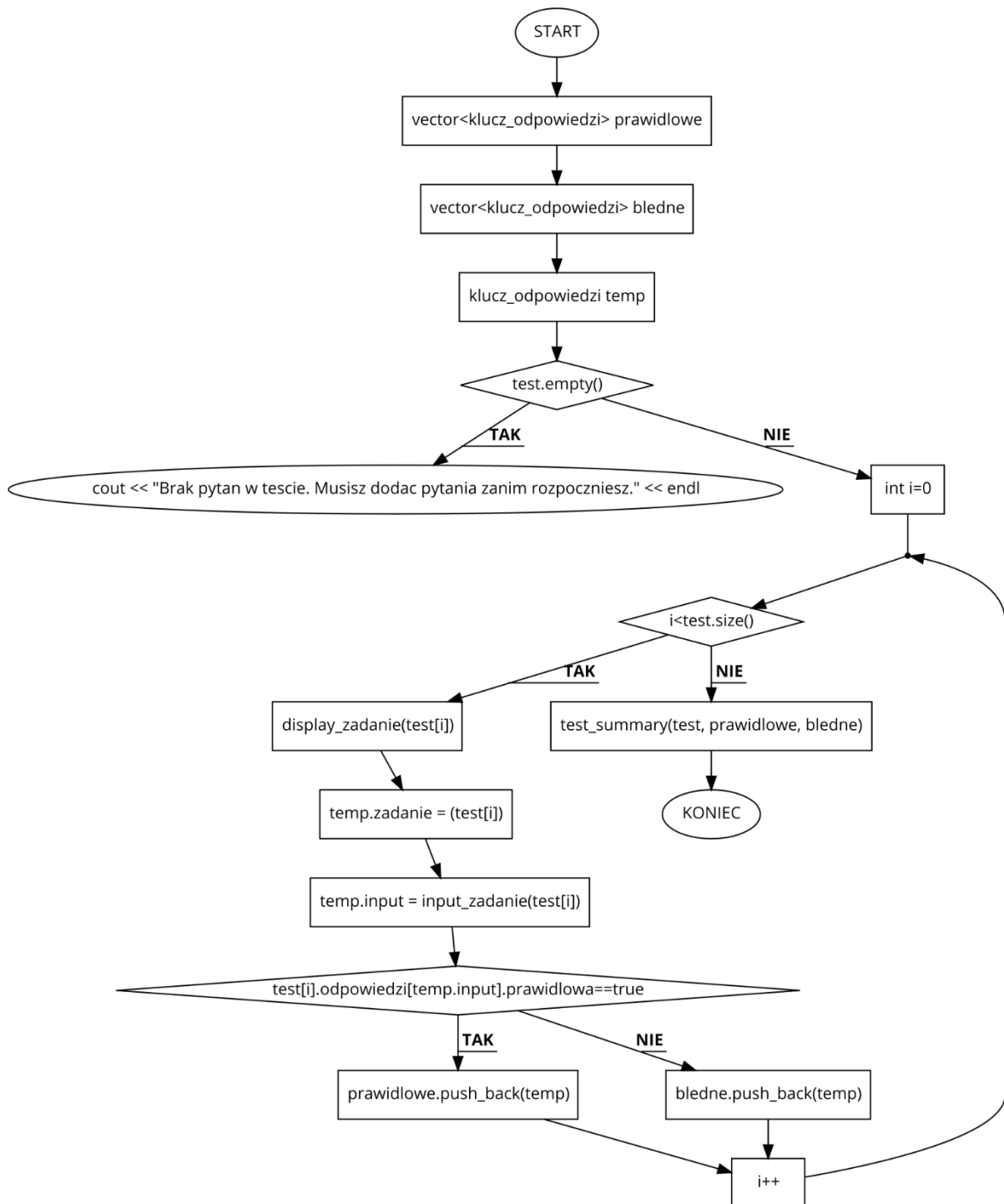
Funkcja przygotowująca test. Jako argumenty przyjmuje nazwę pliku z pytaniami, oraz czy jest on wczytywany jako egzamin.



start_test

```
void start_test(vector<zadanie> test)
```

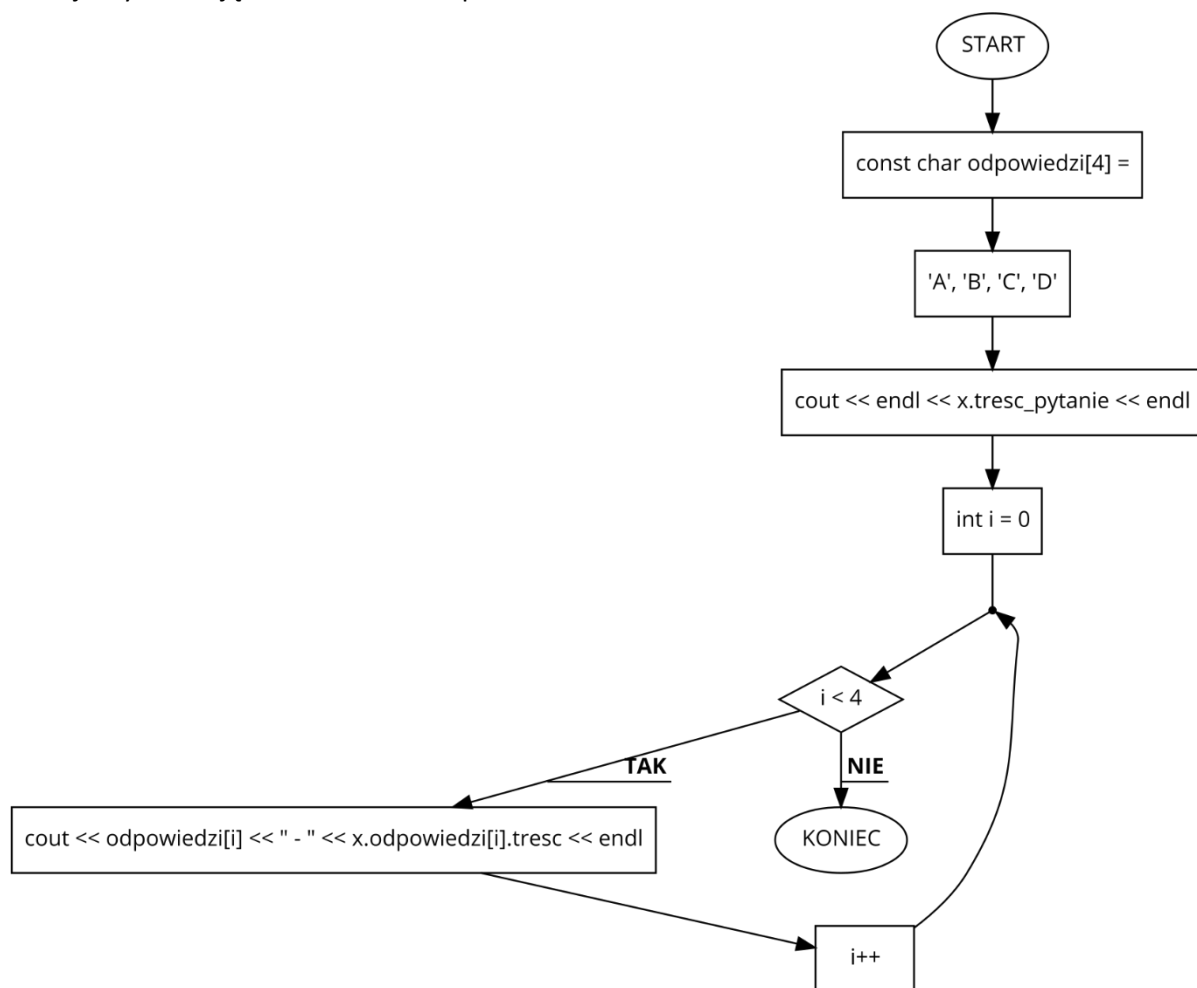
Funkcja odpowiedzialna za prowadzenie testu. Podaje po kolei pytania z wektora *test*, i segreguje je według odpowiedzi użytkownika do wektorów z odpowiedziami prawidłowymi i błędnymi.



display_zadanie

```
void display_zadanie(const zadanie &x)
```

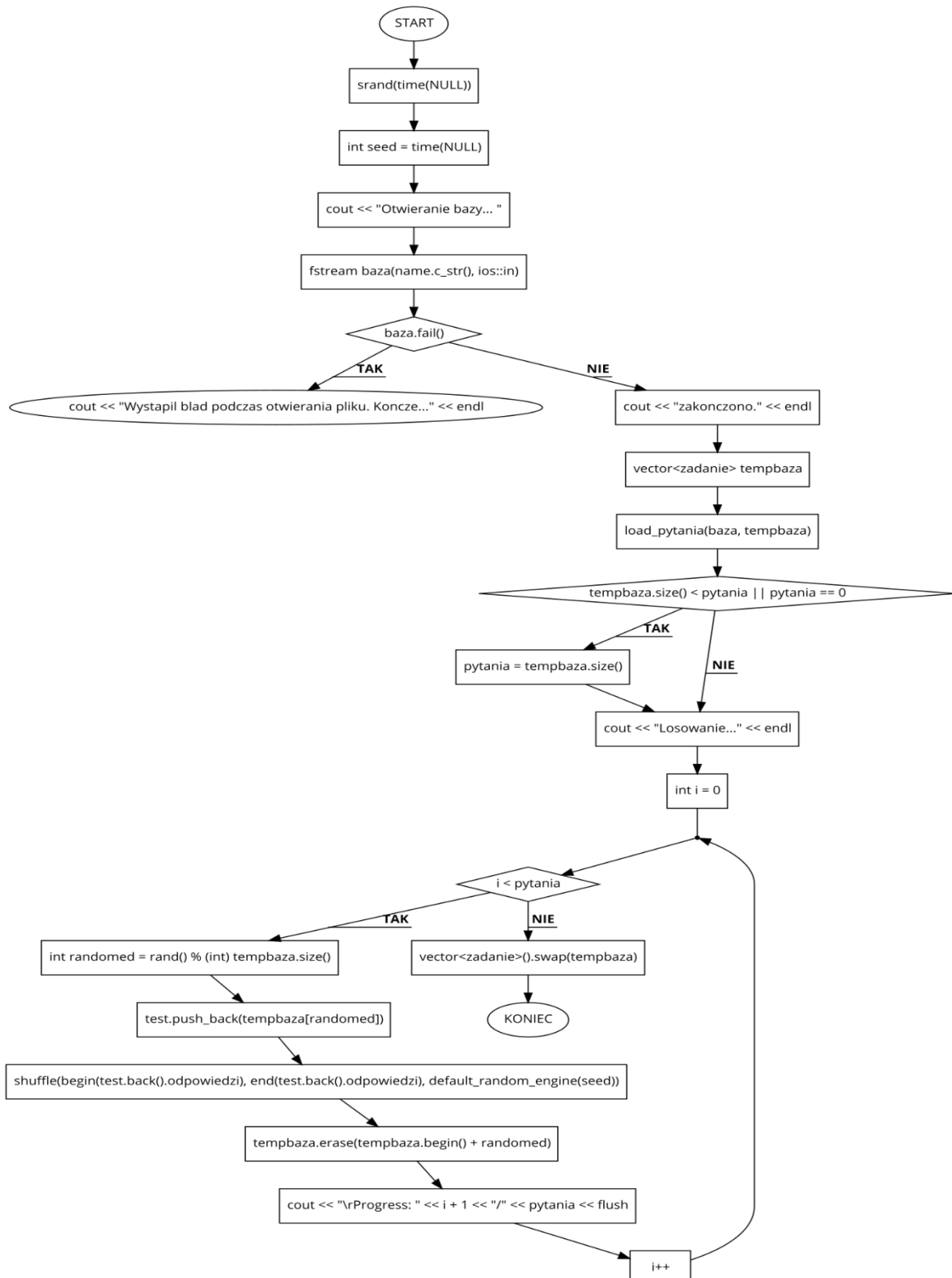
Funkcja wyświetlająca zadanie oraz odpowiedzi.



maszyna_losujaca

```
void maszyna_losujaca(vector<zadanie> &test, int pytania, const string &name)
```

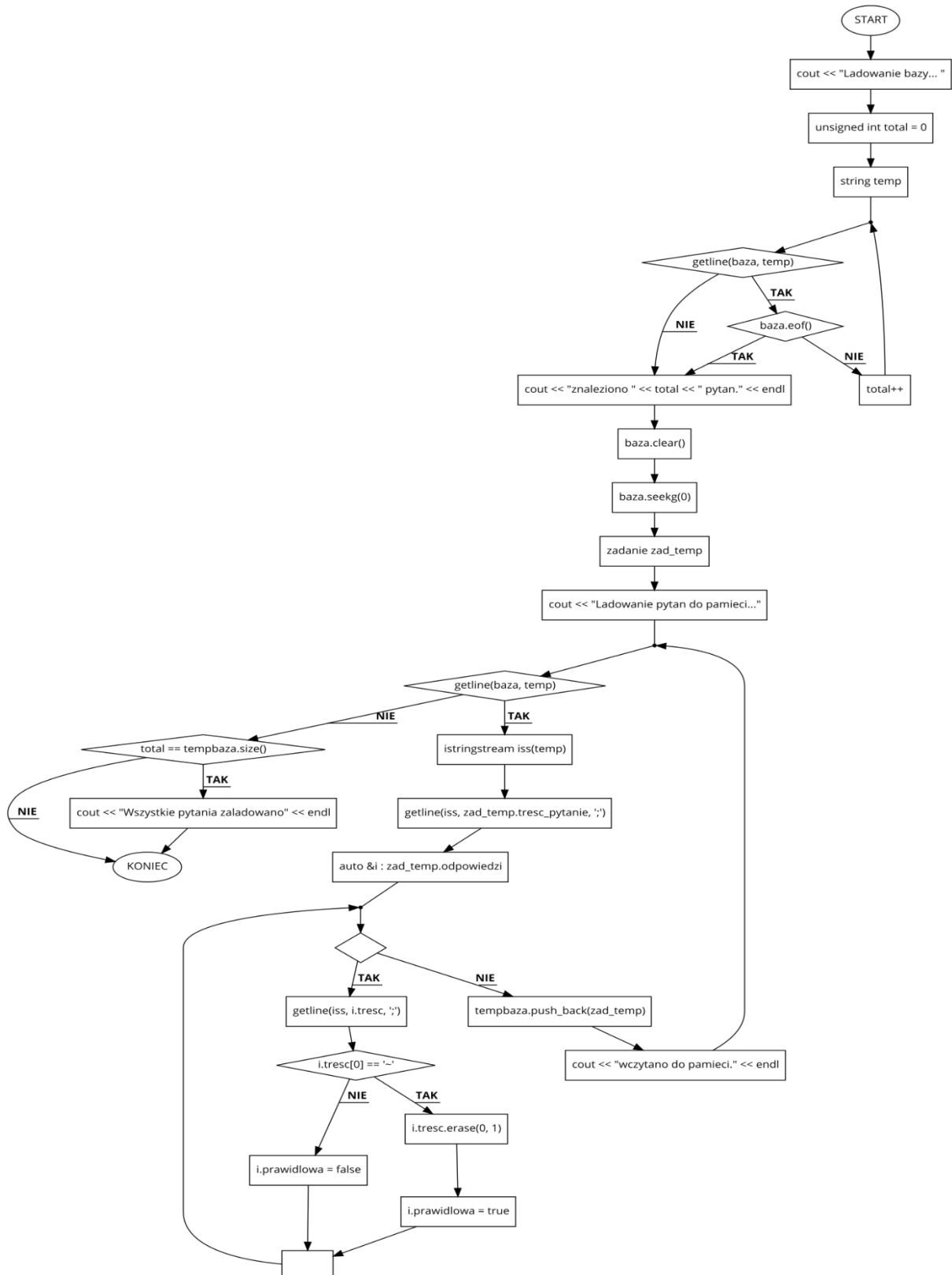
Funkcja losująca pytania z bazy danych do testu. Jako argumenty przyjmuje wektor dla pytań, liczbę pytań oraz nazwę pliku z którego pytania będą losowane.



load_pyrania

```
void load_pyrania(fstream &baza, vector<zadanie> &tempbaza)
```

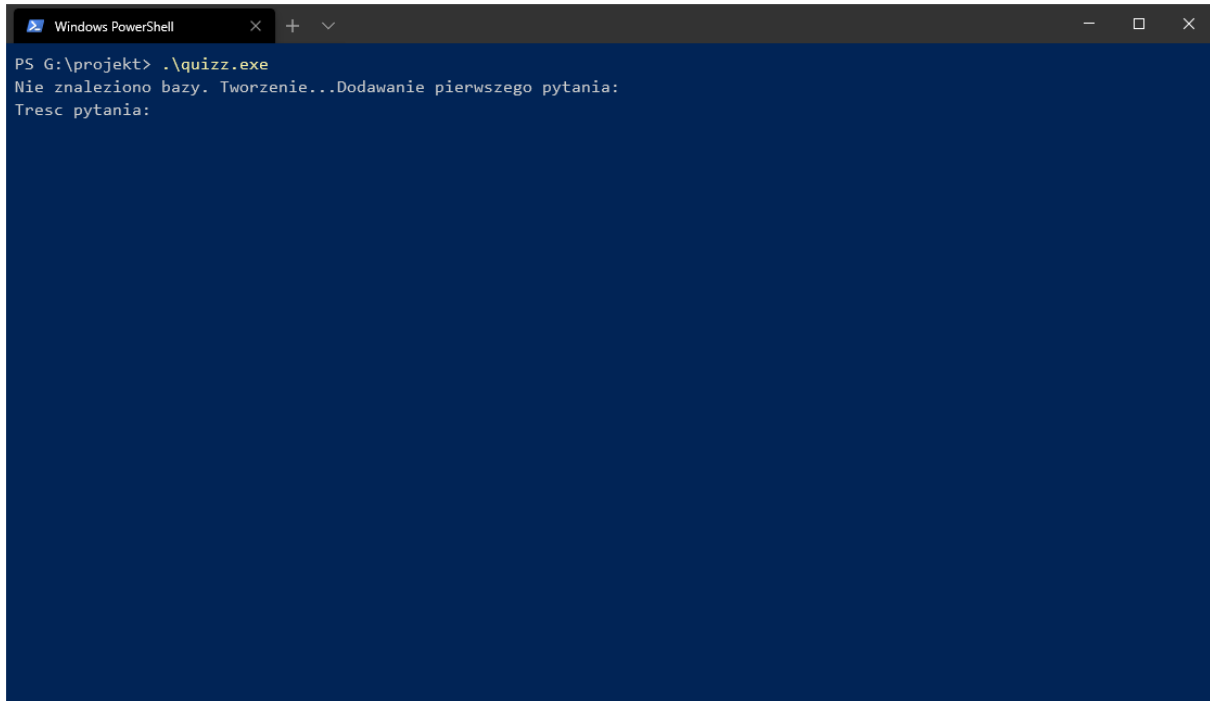
Funkcja zajmująca się ładowaniem pytań z pliku do wektora z pytaniami. Jako argumenty przyjmuje *fstream* z dostępem do pliku, oraz wektor w który mają być załadowane pytania.



Instrukcja użycia programu

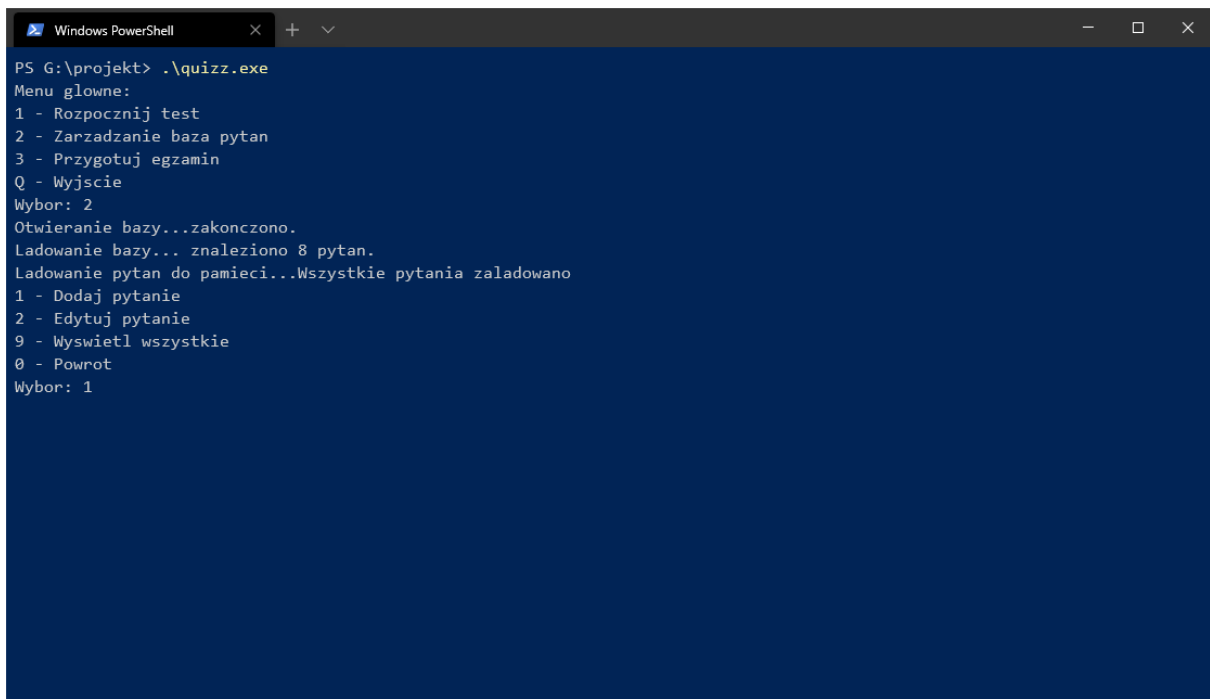
Program działa w trybie tekstowym, z poziomu wiersza poleceń. Można go uruchomić poprzez otwarcie pliku wykonywalnego .exe, lub z poziomu konsoli.

Pierwsze uruchomienie



```
Windows PowerShell
PS G:\projekt> .\quizz.exe
Nie znaleziono bazy. Tworzenie...Dodawanie pierwszego pytania:
Tresc pytania:
```

Przy uruchomieniu, program sprawdza czy plik bazy pytań istnieje. Jeżeli takowy nie istnieje, zostaje utworzony pusty plik, a następnie użytkownik zostaje poproszony o dodanie pierwszego pytania, a potem przeniesiony do menu głównego.



```
Windows PowerShell
PS G:\projekt> .\quizz.exe
Menu glowne:
1 - Rozpocznij test
2 - Zarzadzanie baza pytan
3 - Przygotuj egzamin
Q - Wyjscie
Wybor: 2
Otwieranie bazy...zakonczone.
Ladowanie bazy... znaleziono 8 pytan.
Ladowanie pytan do pamieci...Wszystkie pytania zaladowano
1 - Dodaj pytanie
2 - Edytuj pytanie
9 - Wyszwietl wszystkie
0 - Powrot
Wybor: 1
```

Zalecane jest, by dodać więcej pytań z oczywistych względów. Można to zrobić wpisując 2 na ekranie głównym – menu zarządzania bazą danych – a następnie 1 – by dodać nowe pytanie.

```
Windows PowerShell
PS G:\projekt> .\quizz.exe
Menu glowne:
1 - Rozpoczniej test
2 - Zarzadzanie baza pytan
3 - Przygotuj egzamin
Q - Wyjście
Wybor: 1
Podaj liczbe pytan: 0
```

Test rozpoczynamy, wpisując 1 na ekranie głównym. Zostaniemy poproszeni o podanie liczby pytań. Wpisanie liczby bardzo wysokiej lub 0, zostanie zinterpretowane jako wybranie wszystkich pytań.

```
Na ktorej z rzek wzniesiono Tame Asuanska?
A - Na Dunaju
B - Na Mekongu
C - Na Woldze
D - Na Nilu
Odpowiedz:
```

Każde pytanie zawiera cztery odpowiedzi – A, B, C i D. Prawidłowa jest tylko jedna z nich.

```
Wynik: 4/8=50%
1 - Poprawne odpowiedzi
2 - Bledne odpowiedzi
0 - Zakoncz test
Wybor:
```

Po odpowiedzeniu na wszystkie pytania, program oblicza i prezentuje nam nasz wynik. Stąd możemy sprawdzić, na które pytania odpowiedzieliśmy dobrze, a na które źle.

```
1. Na ktorej z rzek wzniesiono Tame Asuanska?
2. Ciemnobrazowe zytnie pieczywo sprzedawane w cienkich kromkach to:
3. Zgodnie z nazwa, Corn Flakes powstaja z:
4. Który przymiotnik stopniuje się regularnie?
Pytanie do sprawdzenia(0 by wyjsc): |
```

Lista odpowiedzi jest prezentowana w kolejności pojawienia się w teście. Użytkownik może wybrać dowolne z pytań poprzez wpisanie odpowiadającej liczby, lub powrócić wpisując 0.

```
Zgodnie z nazwa, Corn Flakes powstaja z:
A - Ryzu
B - Owsa
C - Jeczmienia
D - Kukurydzy
Twoja odpowiedz: Ryzu
Prawidlowa odpowiedz: Kukurydzy
```

Prezentowane jest pytanie, możliwe odpowiedzi, odpowiedź prawidłowa oraz odpowiedź użytkownika.

Tryb egzaminu

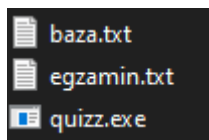
Program wspiera także tryb, gdzie musi poprowadzić test z przygotowanej wcześniej bazy pytań. Można ją przygotować z poziomu programu.

```
Menu glowne:
1 - Rozpoczniej test
2 - Zarzadzanie baza pytan
3 - Przygotuj egzamin
Q - Wyjście
Wybor: 3
Ladowanie bazy... znaleziono 8 pytan.
Ladowanie pytan do pamieci...Wszystkie pytania zaladowano
1. Ktorey klawisz jest zwykly w lewym gornym rogu klawiatury QWERTY?
2. Ciemnobrazowe zytnie pieczywo sprzedawane w cienkich kromkach to:
3. Jakiej blotki nie ma w pelnej talii?
4. Ktorey przymiotnik stopniuje sie regularnie?
5. Zgodnie z nazwa, Corn Flakes powstaja z:
6. Spora dawke naturalnej witaminy D zapewni nam:
7. Na ktorej z rzek wznesiono Tame Asuanska?
8. Obywatele ktorego kraju w 2016 roku zadecydowali w referendum o opuszczeniu Unii Europejskiej?
Podaj nazwe pliku: egzamin.
```

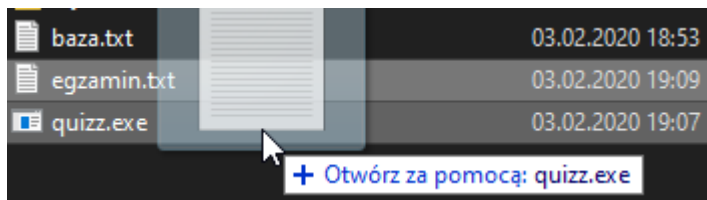
Aby to zrobić, należy wpisać 3 w menu głównym. Program załaduje wszystkie pytania z bazy danych, a następnie poprosi o nazwę pliku z testem. Do nazwy wczytanej przez użytkownika program automatycznie doda rozszerzenie .txt.

```
Wpisz nr pytania ktore chcesz dodac do egzaminu (0 aby wyjsc)8
Wpisz nr pytania ktore chcesz dodac do egzaminu (0 aby wyjsc)3
Wpisz nr pytania ktore chcesz dodac do egzaminu (0 aby wyjsc)5
Wpisz nr pytania ktore chcesz dodac do egzaminu (0 aby wyjsc)2
Wpisz nr pytania ktore chcesz dodac do egzaminu (0 aby wyjsc)0
Zapisywanie...
```

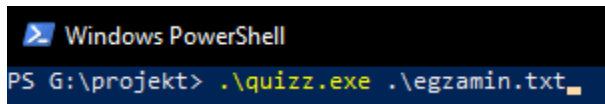
Program będzie prosił o wybór pytań, które chcemy dodać do egzaminu. Aby zakończyć wybieranie, należy wpisać 0, po czym plik zostanie zapisany w katalogu z programem.



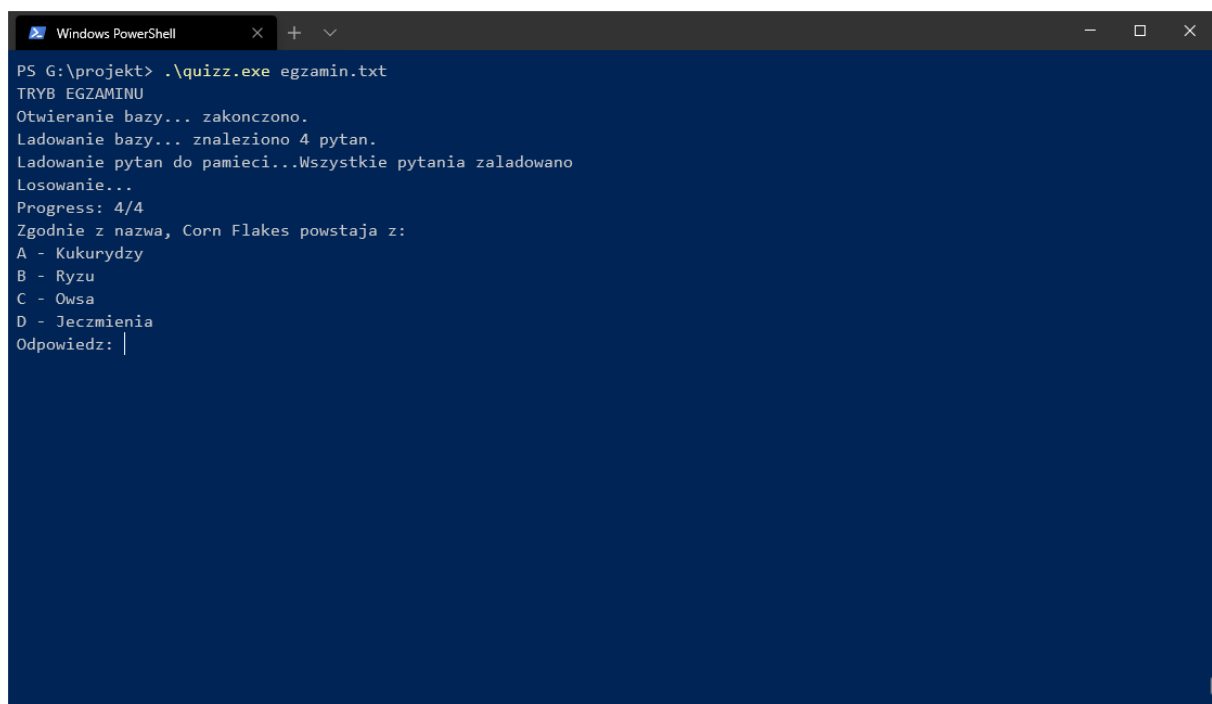
Egzamin można uruchomić na dwa sposoby:



- poprzez przeciągnięcie pliku z egzaminem na plik .exe programu



- poprzez dopisanie do nazwy programu nazwy pliku z egzaminem w konsoli



Program od razu rozpocznie test.

```
Windows PowerShell
Odpowiedz: a

Jakiej blotki nie ma w pelnej talii?
A - Jedynki
B - Czworki
C - Trojki
D - Dwojki
Odpowiedz: a

Ciemnobrazowe zytnie pieczywo sprzedawane w cienkich kromkach to:
A - Chalka
B - Bagietka
C - Pumpernikiel
D - Ciabatta
Odpowiedz: c

Obywatele ktorego kraju w 2016 roku zadecydowali w referendum o opuszczeniu Unii Europejskiej?
A - Portugalii
B - Szwecji
C - Wielkiej Brytanii
D - Grecji
Odpowiedz: c

Wynik: 4/4=100%
1 - Poprawne odpowiedzi
2 - Bledne odpowiedzi
0 - Zakoncz test
Wybor: 0
PS G:\projekt>
```

Po zakończeniu testu, można sprawdzić jego wyniki. Po wyjściu z testu program kończy pracę.

Kod programu

main.cpp

```
#include "struct.h"
#include "test.cpp"
#include "base.cpp"

using namespace std;

int main(int argc, char *argv[]) {
    if (argc == 1) {
        if (!filesystem::exists("baza.txt")) {
            cout << "Nie znaleziono bazy. Tworzenie...";
            fstream plik;
            plik.open("baza.txt", ios::out);
            cout<<"Dodawanie pierwszego pytania:<<endl;
            add_question(plik);
            plik.close();
            cout << "zakonczono." << endl;
        }
        main_menu();
        return 0;
    } else if (argc == 2) {
        cout << "TRYB EGZAMINU" << endl;
        init_test(argv[1], true);
        return 0;
    } else {
        cout << "too much launch arguments." << endl;
        return 1;
    }
}

void main_menu() {
    char choice;
    while (true) {
        cout << "Menu glowne:" << endl
            << "1 - Rozpoczniej test" << endl
            << "2 - Zarzadzanie baza pytan" << endl
            << "3 - Przygotuj egzamin" << endl
            << "Q - Wyjście" << endl
            << "Wybor: ";
        cin >> choice;
        switch (choice) {
            case '1':
                init_test();
                break;
            case '2':
                base_menu();
                break;
            case '3':
                edit_exam();
                break;
            case 'q':
            case 'Q':
                return;
            default:
                cout << "Nieprawidlowy wybor.";
                break;
        }
    }
}
```


struct.h

```
//  
// Created by sabre on 30.12.2019.  
//  
  
#ifndef QUIZZ_STRUCT_H  
#define QUIZZ_STRUCT_H  
  
#include <iostream>  
#include <fstream>  
#include <vector>  
#include <sstream>  
#include <cstdlib>  
#include <iomanip>  
#include <filesystem>  
#include <random>  
  
using namespace std;  
  
struct odpowiedz {  
    string tresc;  
    bool prawdziwa=false;  
};  
  
struct zadanie {  
    string tresc_pytanie;  
    array<odpowiedz, 4> odpowiedzi;  
};  
  
struct klucz_odpowiedzi {  
    zadanie zadanie;  
    int input{};  
};  
  
// Declarations  
void main_menu();  
void base_menu();  
void edit_exam();  
void start_test(vector<zadanie> test);  
void maszyna_losujaca(vector<zadanie> &test, int pytania = 0, const  
string &name = "baza.txt");  
void input_tresc(zadanie &x);  
void input_odpowiedz(zadanie &x);  
bool input_prawidlowa(zadanie &x);  
zadanie add_question(fstream &baza);  
void display_zadanie(const zadanie &x);  
void input_odpowiedz(zadanie &x, int i);  
int input_zadanie(zadanie &x);  
int score_zadanie(zadanie &x, char odpowiedz);  
void test_summary(const vector<zadanie> &test, const  
vector<klucz_odpowiedzi> &prawidlowe,  
                const vector<klucz_odpowiedzi> &bledne);  
void zadanie_summary(const vector<klucz_odpowiedzi> &rozwiązane);  
void load_pytanias(fstream &baza, vector<zadanie> &tempbaza);  
void edit_baza(vector<zadanie> temp);  
void write_question(fstream &baza, const zadanie &temp);  
void edit_menu(vector<zadanie> &temp, int pytanie);
```

```
#endif //QUIZZ_STRUCT_H
```

test.cpp

```
#include "struct.h"

using namespace std;

void init_test(const string &nazwa = "baza.txt", bool egzamin = false) {
    vector<zadanie> test;
    if (egzamin) {
        maszyna_losujaca(test, 0, nazwa);
    } else {
        cout << "Podaj liczbe pytan: ";
        int pytania;
        cin >> pytania;
        maszyna_losujaca(test, pytania);
    }
    start_test(test);
}

void start_test(vector<zadanie> test) {
    vector<klucz_odpowiedzi> prawidlowe;
    vector<klucz_odpowiedzi> bledne;
    klucz_odpowiedzi temp = {};
    if (test.empty()) {
        cout << "Brak pytan w tescie. Musisz dodac pytania zanim
rozpoczniesz." << endl;
        return;
    }
    for (auto &i:test) {
        display_zadanie(i);
        temp.zadanie = i;
        temp.input = input_zadanie(i);
        if (i.odpowiedzi[temp.input].prawidlowa)
            prawidlowe.push_back(temp);
        else
            bledne.push_back(temp);
    }
    test_summary(test, prawidlowe, bledne);
}

void test_summary(const vector<zadanie> &test, const
vector<klucz_odpowiedzi> &prawidlowe,
const vector<klucz_odpowiedzi> &bledne) {
    char choice;
    cout << "Wynik: " << prawidlowe.size() << "/" << test.size() << "="
<< setprecision(4)
        << ((float) prawidlowe.size() * 100.0 / test.size()) << "%" <<
endl;
```

```

while (true) {
    cout << "1 - Poprawne odpowiedzi" << endl
        << "2 - Bledne odpowiedzi" << endl
        << "0 - Zakoncz test" << endl;
    cout << "Wybor: ";
    cin >> choice;
    switch (choice) {
        case '1':
            zadanie_summary(prawidlowe);
            break;
        case '2':
            zadanie_summary(bledne);
            break;
        case '0':
            return;
        default:
            cout << "Nieprawidlowy wybor." << endl;
            break;
    }
}

void zadanie_summary(const vector<klucz_odpowiedzi> &rozwiązane) {
    int wybor = 0;
    do {
        for (unsigned int i = 0; i < rozwiązanie.size(); i++) {
            cout << i + 1 << ". " << rozwiązanie[i].zadanie.tresc_pytanie
<< endl;
        }
        cout << "Pytanie do sprawdzenia(0 by wyjść): ";
        cin >> wybor;
        if (wybor == 0)
            continue;
        else if (wybor > rozwiązanie.size())
            cout << "Nieprawidlowy wybor" << endl;
        else {
            display_zadanie(rozwiązane[wybor - 1].zadanie);
            cout << "Twoja odpowiedź: " << rozwiązanie[wybor -
1].zadanie.odpowiedzi[rozwiązane[wybor - 1].input].tresc
<< endl;
            cout << "Prawidlowa odpowiedź: ";
            for (auto &i:rozwiązane[wybor - 1].zadanie.odpowiedzi) {
                if (i.prawidlowa) {
                    cout << i.tresc << endl;
                    break;
                }
            }
        }
    } while (wybor != 0);
}

#pragma clang diagnostic push
#pragma ide diagnostic ignored "modernize-use-nullptr"
#pragma ide diagnostic ignored "cert-msc50-cpp"
#pragma ide diagnostic ignored "cert-msc32-c"

```

```

void maszyna_losujaca(vector<zadanie> &test, int pytania, const string
&name) {
    srand(time(NULL));
    int seed = time(NULL);
    cout << "Otwieranie bazy... ";
    fstream baza(name.c_str(), ios::in);
    if (baza.fail()) {
        cout << "Wystapil blad podczas otwierania pliku. Koncze..." <<
endl;
        return;
    }
    cout << "zakonczone." << endl;
    vector<zadanie> tempbaza;
    load_pytnia(baza, tempbaza);
    if (tempbaza.size() < pytania || pytania == 0) {
        pytania = tempbaza.size();
    }
    cout << "Losowanie..." << endl;
    for (int i = 0; i < pytania; i++) {
        int randomed = rand() % (int) tempbaza.size();
        test.push_back(tempbaza[randomed]);
        shuffle(begin(test.back().odpowiedzi),
end(test.back().odpowiedzi), default_random_engine(seed));
        tempbaza.erase(tempbaza.begin() + randomed);
        cout << "\rProgress: " << i + 1 << "/" << pytania << flush;
    }
    vector<zadanie>().swap(tempbaza);
}

void display_zadanie(const zadanie &x) {
    const char odpowiedzi[4] = {'A', 'B', 'C', 'D'};
    cout << endl << x.tresc_pytnie << endl;
    for (int i = 0; i < 4; i++) {
        cout << odpowiedzi[i] << " - " << x.odpowiedzi[i].tresc << endl;
    }
}

int input_zadanie(zadanie &x) {
    char odpowiedz = 0;
    cout << "Odpowiedz: ";
    cin >> odpowiedz;
    return score_zadanie(x, odpowiedz);
}

```

```

int score_zadanie(zadanie &x, char odpowiedz) {
    int wynik = 4;
    switch (odpowiedz) {
        case 'a':
        case 'A':
            wynik = 0;
            break;
        case 'b':
        case 'B':
            wynik = 1;
            break;
        case 'c':
        case 'C':
            wynik = 2;
            break;
        case 'd':
        case 'D':
            wynik = 3;
            break;
        default:
            break;
    }
    return wynik;
}

```

base.cpp

```

#include "struct.h"

using namespace std;

void base_menu() {
    cout << "Otwieranie bazy...";
    fstream baza("baza.txt", ios::in | ios::out);
    if (baza.fail()) {
        cout << "Wystapil blad podczas otwierania pliku. Koncze..." <<
endl;
        return;
    }
    cout << "zakonczono." << endl;
    vector<zadanie> temp;
    load_pytan(a(baza, temp);
    while (cout << "1 - Dodaj pytanie" << endl
        << "2 - Edytuj pytanie" << endl
        << "9 - Wyszwietl wszystkie " << endl
        << "0 - Powrot" << endl
        << "Wybor: ") {
        baza.clear();
        baza.seekg(0);
        int wybor;
        cin >> wybor;
        switch (wybor) {
            case 1:
                temp.push_back(add_question(baza));
                break;
            case 2:
                edit_baza(temp);
                break;

```

```

        case 9:
            for (int i = 0; i < temp.size(); i++) {
                cout << i + 1 << ". " << temp[i].tresc_pytanie <<
endl;

                }
                cout << endl;
                break;
            case 0: {
                cout << "Zamykanie bazy...";
                baza.close();
                fstream bufor;
                bufor.open("temp.txt", ios::out);
                for (auto &i:temp)
                    write_question(bufor, i);
                bufor.close();
                remove("baza.txt");
                rename("temp.txt", "baza.txt");
                cout << "zamknieto. Powrot..." << endl;
                return;
            }
            default:
                cout << "Nieprawidlowy wybor." << endl;
                break;
        }
    }
}

void edit_menu(vector<zadanie> &temp, int pytanie) {
    display_zadanie(temp[pytanie]);
    do {
        cout << "1 - edytuj tresc pytania" << endl
            << "2 - edytuj tresc odpowiedzi" << endl
            << "3 - edytuj prawidlowa odpowiedz" << endl
            << "4 - usun pytanie" << endl
            << "0 - powrot" << endl;
        cout << "Wybor: ";
        cin >> pytanie;
        switch (pytanie) {
            case 1:
                input_tresc(temp[pytanie]);
                break;
            case 2:
                int odpowiedz;
                cout << "Ktora odpowiedz chcesz edytowac? (1-4)";
                cin >> odpowiedz;
                if (odpowiedz <= 4 && odpowiedz >= 1)
                    input_odpowiedz(temp[pytanie], (odpowiedz - 1));
                else cout << "Nieprawidlowy wybor. Powrot..." << endl;
                break;
            case 3:
                input_prawidlowa(temp[pytanie]);
                break;
            case 4:
                temp.erase(temp.begin() + pytanie);
                return;
        }
    } while (pytanie != 0);
}

```

```

        case 0: {
            zadanie x = temp[pytanie];
            temp.erase(temp.begin() + pytanie);
            temp.insert(temp.begin() + pytanie, x);
            return;
        }
        default:
            cout << "Nieprawidlowy wybor." << endl;
            break;
    }
} while (pytanie != 0);
}

zadanie add_question(fstream &baza) {
    zadanie temp;
    input_tresc(temp);
    input_odpowiedz(temp);
    input_prawidlowa(temp);
    cout << "Pytanie dodane." << endl;
    write_question(baza, temp);
    return temp;
}

void write_question(fstream &baza, const zadanie &temp) {
    baza.seekp(0, ios::end);
    baza << temp.tresc_pytanie << ";";
    for (const auto &i : temp.odpowiedzi) {
        if (i.prawidlowa)
            baza << "~";
        baza << i.tresc << ";";
    }
    baza << endl;
}

void input_tresc(zadanie &x) {
    cout << "Tresc pytania: ";
    cin.ignore(1000, '\n');
    getline(cin, x.tresc_pytanie);
}

void input_odpowiedz(zadanie &x) {
    for (int i = 0; i < 4; i++) {
        cout << "Odpowiedz " << i + 1 << ": ";
        getline(cin, x.odpowiedzi[i].tresc);
    }
}

void input_odpowiedz(zadanie &x, int i = 0) {
    cout << "Odpowiedz " << i + 1 << ": ";
    getline(cin, x.odpowiedzi[i].tresc);
}

```

```

bool input_prawidlowa(zadanie &x) {
    int prawidlowa;
    cout << "Prawidlowa odpowiedz (1,2,3,4): 1`";
    while (cin >> prawidlowa) {
        if (prawidlowa > 0 && prawidlowa < 5) {
            x.odpowiedzi[prawidlowa - 1].prawidlowa = true;
            return true;
        } else cout << "Nieprawidlowy znak - dozwolone tylko 1, 2, 3, 4:
";
    }
    return false;
}

void load_pytnia(fstream &baza, vector<zadanie> &tempbaza) {
    cout << "Ladowanie bazy... ";
    unsigned int total = 0;
    string temp;
    while (getline(baza, temp)) {
        if (baza.eof())
            break;
        total++;
    }
    cout << "znaleziono " << total << " pytan." << endl;
    baza.clear();
    baza.seekg(0);
    zadanie zad_temp;
    cout << "Ladowanie pytan do pamieci...";
    while (getline(baza, temp)) {
        istringstream iss(temp);
        getline(iss, zad_temp.tresc pytanie, ';');
        for (auto &i : zad_temp.odpowiedzi) {
            getline(iss, i.tresc, ';');
            if (i.tresc[0] == '~') {
                i.tresc.erase(0, 1);
                i.prawidlowa = true;
            } else i.prawidlowa = false;
        }
        tempbaza.push_back(zad_temp);
    }
    if (total == tempbaza.size())
        cout << "Wszystkie pytania zaladowano" << endl;
}

void edit_baza(vector<zadanie> temp) {
    int pytanie = 0;
    do {
        cout << "Wybierz pytanie(0 by wyjsc): ";
        cin >> pytanie;
        if (pytanie == 0) {
            return;
        }
        pytanie -= 1;
        if (pytanie >= temp.size())
            cout << "Nieprawidlowy wybor." << endl;
    } while (pytanie >= temp.size());
    edit_menu(temp, pytanie);
}

```



```

void edit_exam() {
    fstream baza;
    baza.open("baza.txt", ios::in);
    vector<zadanie> temp;
    load_pytonia(baza, temp);
    baza.close();
    int pytanie = 0;
    for (int i = 0; i < temp.size(); i++) {
        cout << i + 1 << ". " << temp[i].tresc_pytonie << endl;
    }
    cout << "Podaj nazwe pliku: ";
    string nazwa;
    cin.ignore(1000, '\n');
    getline(cin, nazwa);
    fstream plik;
    plik.open((nazwa+".txt").c_str(), ios::out);
    cout << endl;
    do {
        cout << "Wpisz nr pytania ktore chcesz dodac do egzaminu (0 aby
wyjsc)";
        cin >> pytanie;
        if (pytonie == 0)
            continue;
        else if (pytonie > temp.size())
            cout << "Nie ma takiego pytania w bazie." << endl;
        else
            write_question(plik, temp[pytonie - 1]);
    } while (pytonie != 0);
    cout << "Zapisywanie..." << endl;
    plik.close();
}

```