

Programowanie Komputerów

Projekt: Quiz

Autor: Korwin Bieniek
Semestr: drugi
Grupa: 3
Seksja: 1

Treść zadania

Napisz program przeprowadzający quiz na podstawie bazy pytań przechowywanej w pliku.

Po uruchomieniu program losuje zadana liczbę pytań z całej bazy i przeprowadza QUIZ.

Baza pytań jest skonstruowana według następującego schematu:

<treść pytania>

<nr poprawnej odpowiedzi>

<treść odp 1>

<treść odp 2>

<treść odp 3>

<treść odp 4>

Przykładowy plik z pytaniami ma postać:

Pytania.txt

Trzecia planeta od słońca jest:

1

Ziemia

Mars

Wenus

Księżyc

Wielka Brytania leży w:

3

Afryce

Ameryce Pd.

Europie

Azji

...

Program wyświetla wybrane pytania z bazy w losowej kolejności, podlicza liczbę poprawnych odpowiedzi

i zapisuje to do pliku z wynikami dla konkretnej bazy pytań wraz z godziną przeprowadzenia quizu i nickiem użytkownika.

Program uruchamiany jest z linii poleceń z następującymi przełącznikami:

-o plik z bazą pytań

-n nick uczestnika quizu

-q liczba zadawanych pytań

Analiza

rozwiązania

Głównym zadaniem programu jest przeprowadzenie quizu dla użytkownika, korzystając z podanego pliku tekstowego z pytaniami. Program rozpoczyna się menu głównym, i prostymi informacjami o formie przeprowadzenia quizu. Nie znamy długości pliku wejściowego, dlatego funkcja OpenQuestions czyta program linią po linii aż do końca pliku. Linijki są kolejno umieszczane w odpowiednich zmiennych struktury QUESTIONS. Następnie losuje X pytań wybranych przez użytkownika ze wszystkich pytań w pliku. Liczby są na bieżąco weryfikowane, żeby nie występowały powtórzenia w pytaniach.

Następnie funkcja wyświetla pytania i pobiera odpowiedzi od użytkownika. Funkcja checkAnswer weryfikuje odpowiedź, i tylko w przypadku dobrej odpowiedzi dodaje punkty, w innym przypadku ilość punktów się nie zmienia.

Na końcu, program zapisuje wynik quizu w formie NAZWAQUIZU_score.txt, gdzie zamieszcza nazwę użytkownika, punkty z quizu, oraz datę przeprowadzenia quizu.

Specyfikacja zewnętrzna

Program należy uruchomić z linii poleceń (cmd.exe) w następujący sposób;

<program.exe> -o <plik z pytaniami> -n <nazwa użytkownika> -q <ilość pytań dla użytkownika>

Można podać pełną ścieżkę do programu, aby uruchomić go z katalogu.

Po uruchomieniu, program wita nas krótkim komunikatem, wyświetla naszą nazwę użytkownika.

```
        Welcome BabaJaga
          to
        the Quiz Program

-----
- Press N to start a new game
- Press any other key to quit the program
-----
```

W tym momencie możemy nacisnąć znak 'n', aby rozpocząć grę, lub inny przycisk jeżeli chcemy zamknąć program.

Następnie program wyświetla nam ilość pytań na które chcemy odpowiedzieć, nazwę pliku z pytaniami, oraz podaje zasady liczenia punktów za quiz. Użytkownik naciska dowolny przycisk, aby kontynuować

```
You've chosen to answer 2 questions from the question base questions.txt
You can answer all the questions
Your score will be the amount of correctly answered questions

Press any key to continue
```

Kolejnym etapem jest przeprowadzenie quizu. Przed użytkownikiem pojawia się nr pytania rozdzielony / na ilość pytań w quizie, treść pytania, oraz cztery odpowiedzi.

```
Pytanie 1/2

Co w matematyce oznaczamy symbolem h?

A) szerokosc
B) wysokosc
C) jednostke
D) calke nieoznaczona
```

Użytkownik wybiera teraz jedną z odpowiedzi i wpisuje odpowiadającą jej literkę.

```
Pytanie 1/2
Co w matematyce oznaczamy symbolem h?
A) szerokosc
B) wysokosc
C) jednostke
D) calke nieoznaczona

Incorrect answer!
Press any key to continue to the next question
```

W przypadku błędnej odpowiedzi lub wpisania litery innej niż domyślne (a,b,c,d – tutaj dodatkowo pojawia się informacja iż X odpowiedzi, którą wpisał użytkownik nie istnieje), na ekranie pojawia się informacja, iż odpowiedź jest nieprawidłowa. Następnie użytkownik naciska dowolny przycisk co przenosi go do następnego pytania.

W przypadku prawidłowej odpowiedzi, na ekranie pojawia się stosowna informacja i także użytkownik naciska dowolny przycisk aby przejść dalej.

```
Pytanie 2/2
Portret damy z gronostajem namalował:
A) Leonardo da Vinci
B) Adam kalota
C) Michal Aniol
D) Aniol Stroz

Correct answer!
Press any key to continue to the next question
```

Na końcu program wyświetla gratulacje dla użytkownika oraz liczbę punktów, które uzyskał. Dodatkowo powiadamia użytkownika, że wynik zostanie zapisany w pliku <plik_wejściowy>_score.txt.

```

      Congratulations! You have finished the quiz with the score: 1
      Now your score will be saved in questions_score.txt

Process returned 0 (0x0)   execution time : 1075.607 s
Press any key to continue.
```

Jeżeli użytkownik pierwszy raz przeprowadza quiz na danej bazie pytań, to na końcu pojawi się dodatkowa informacja o utworzeniu pliku na wynik

```

Cannot find a file questions2_score.txt - creating a new one
Process returned 0 (0x0)   execution time : 4.065 s
Press any key to continue.
```

W przypadku wyboru błędnego lub nieistniejącego pliku, program wyświetli komunikat „No such file” i zwróci kod błędu.

```
No such file
```

Dodatkowo w przypadku gdzie liczba pytań wybrana przez użytkownika nie będzie liczbą, zostanie wyświetlony komunikat. Następnie program zakończy działanie.

```
'ab' is not valid
```

Natomiast w przypadku, gdzie użytkownik poda więcej pytań niż zawiera się w pliku, zostanie wyświetlony następujący komunikat, i liczba pytań zostanie zmniejszona do maksymalnej liczby pytań w pliku. Użytkownik kontynuuje naciskając dowolny przycisk.

```

      The file has only 24 questions.
      Changing the number of questions to answer to max questions in the file.

      Press any key to continue_
```

Specyfikacja wewnętrzna

typedef struct questions

```
{  
    char code[200];  
    char correct[3];  
    char a[50];  
    char b[50];  
    char c[50];  
    char d[50];  
} QUESTIONS;
```

Struktura pytań, każda linijka w pliku tekstowym jest jednorazowo przyporządkowywana do odpowiedniej zmiennej w strukturze. Każda zmienna tej struktury jest tablicą z predefiniowaną wielkością, natomiast sama struktura jest alokowana dynamicznie.

Program posiada zabezpieczenia przed: złymi przełącznikami (wartości bazowe), błędnym rodzajem pliku oraz błędnym plikiem, brakiem wartości liczbowej w przełączniku -q (liczba pytań). Zabezpieczenia przed wyborem 0 pytań. Program weryfikuje odpowiedź wybraną przez użytkownika i zwraca informację, jeżeli ta jest poza domyślnymi wartościami (a,b,c,d). Dodatkowo istnieje zabezpieczenie przed zbyt małą lub zbyt dużą ilością argumentów, przed pustymi wartościami przełączników. Koryguje zbyt dużą ilość pytań wybraną przez użytkownika na maksymalną liczbę pytań w pliku. Sprawdza czy plik wyjściowy istnieje i jeżeli nie to go tworzy. Weryfikuje czy program się zapisał. Zwalnia zaalokowaną dynamicznie pamięć.

int OpenQuestions(int nrOfQ, char* filename);

Funkcja odpowiada za główną logikę programu. Na początku funkcja czyta pytania aż do końca pliku wejściowego. Następnie funkcja sprawdza czy użytkownik nie wybrał zbyt wielu pytań, w porównaniu do ilości pytań w pliku wejściowym. W funkcji odbywa się generowanie losowych X (wybrana ilość pytań przez użytkownika) pytań z całej puli z pliku wejściowego. Następnie pytania są wyświetlane, i funkcja pobiera od użytkownika odpowiedzi, które są sprawdzane w funkcji checkAnswer.

void checkIfOpen(FILE* fp);

Funkcja sprawdza czy plik wejściowy istnieje. W przypadku jeżeli nie istnieje to zwraca 0.

int tooManyQuestions(int nrOfQ, int i);

Funkcja porównuje ilość pytań wybraną przez użytkownika z ilością pytań w pliku wejściowym. W przypadku kiedy użytkownik wybrał zbyt dużą ilość pytań, funkcja zwraca maksymalną liczbę pytań w pliku, w innym przypadku funkcja zwraca ilość pytań wybraną przez użytkownika.

int fRead(QUESTIONS questions[], int i, FILE* fp);

Funkcja pobiera linijki z pliku i przypisuje je do zmiennych w strukturze.

int checkAnswer(char answer, QUESTIONS questions[], int x, int base[]);

Funkcja konwertuje char correct (zmienna przechowująca poprawną odpowiedź) na liczbę, następnie porównuje wybór użytkownika z poprawną odpowiedzią, w przypadku poprawnej odpowiedzi zwraca 1, w przypadku niepoprawnej zwraca 0.

char* getTime();

Funkcja korzysta z biblioteki time.h. Funkcja zwraca dzień, miesiąc czas (HH:MM:SS) oraz rok w którym został skompilowany program.

```
void mainMenu(char* user);
```

Funkcja wyświetla instrukcje dla użytkownika w ekranie powitalnym. Funkcja także wita użytkownika wybranym przez niego nickiem.

```
void startGame(int nrOfQ, char* filename);
```

Funkcja wyświetla ilość pytań które wybrał użytkownik, oraz nazwę pliku wejściowego. Funkcja pokazuje instrukcje dotyczące liczenia punktów w quizie.

```
char* TxtDel(char* name);
```

Funkcja usuwa z nazwy pliku wejściowego ostatnie 4 znaki, które definiują typ pliku (eg. Test.txt). Funkcja zwraca nazwę pliku wejściowego bez końcówki.

```
void saveToFile(char* filename, int nrOfQ, char* user);
```

Funkcja zapisuje wynik użytkownika do pliku tekstowego w formie <nazwa pliku wejściowego>_score.txt. W pliku umieszczona jest data przeprowadzenia quizu, nazwa użytkownika oraz ilość poprawnych odpowiedzi.

```
int inputHandling(int opt, int *nrOfQ, int argc, char** argv, char** fileName, char** user);
```

Funkcja ta obejmuje prawidłowe wprowadzanie przełączników. W przypadku błędnego przełącznika, pokazywana jest informacja na ekranie startowym, natomiast zmienna ta nie jest brana pod uwagę, i używana jest wartość domyślna pominiętej zmiennej (dla -o nie ma wartości domyślnej, gdyż zakładamy że bez podania pliku, plik ten nie będzie istniał, dla -n jest to 'user', dla -q jest to 1)

Testowanie

Plik wejściowy (questions.txt), zawiera 24 pytania.

Test I:

Użytkownik wybrał nazwę użytkownika „User” oraz liczbę pytań do odpowiedzi 2.

Program wywołany w sposób ProjektQuiz.exe -o questions.txt -n User -q 2

Użytkownik udzielił niepoprawnej odpowiedzi na pierwsze pytanie oraz poprawnej odpowiedzi na pytanie drugie. Użytkownik kończy quiz z jednym punktem. Plik z punktacją był już wcześniej utworzony

```

                                     Congratulations! You have finished the quiz with the score: 1
                                     Now your score will be saved in questions_score.txt

Process returned 0 (0x0)   execution time : 18.230 s
Press any key to continue.
```

Wynik testu: Poprawne działanie programu

```

questions_score — Notatnik
Plik  Edycja  Format  Widok  Pomoc

Date: Fri May 22 07:41:59 2020
Player: User
Correct answers: 1
```

Test II:

Użytkownik wybrał nazwę użytkownika „5315312” oraz liczbę pytań do odpowiedzi 1786.
Program wywołany w sposób `ProjektQuiz.exe -o questions.txt -n 5315312 -q 1786`

Nazwa użytkownika może być dowolna, więc także może być liczbą. Natomiast liczba pytań wybrana przez użytkownika jest większa niż liczba pytań w pliku. Program skorygował błąd i zmniejszył liczbę pytań dla użytkownika:

```
The file has only 24 questions.  
Changing the number of questions to answer to max questions in the file.  
  
Press any key to continue.
```

Dalej został przeprowadzony quiz, użytkownik otrzymał 8 punktów.

```
Date: Fri May 22 07:49:30 2020  
Player: 5315312  
Correct answers: 8
```

Wynik testu: Poprawne działanie programu

Test III:

Użytkownik wybrał nazwę użytkownika „xXDemonySlayerXx” oraz liczbę pytań do odpowiedzi 17a86.
Program wywołany w sposób `ProjektQuiz.exe -o questions.txt -n xXDemonySlayerXx -q 17a86`

Program zakończył swoje działanie, gdyż przełącznik `-q` nie otrzymał liczby

```
'17a86' is not valid  
  
Process returned 0 (0x0)   execution time : 0.019 s  
Press any key to continue.
```

Wynik testu: Błędne działanie programu, program nie otrzymał liczby w przełączniku `-q`.

Test IV

Użytkownik wybrał nazwę użytkownika „zzz” oraz liczbę pytań do odpowiedzi 0.
Program wywołany w sposób `ProjektQuiz.exe -o questions.txt -n zzz -q 0`

0 pytań wybranych przez użytkownika jest wyłapywane w funkcji `main`, zwraca komunikat z błędem i kończy program.

```
Bład: Wybrano 0 pytan  
  
Process returned 0 (0x0)   execution time : 0.171 s  
Press any key to continue.  
-
```

Wynik testu: Błędne działanie programu, program otrzymał 0 pytań od użytkownika, i zakończył pracę wyświetlając komunikat o błędzie.

Test V

Użytkownik wybrał nazwę użytkownika „HKL78//]],[,[],m[]]” oraz liczbę pytań do odpowiedzi 12.
Program wywołany w sposób ProjektQuiz.exe -o questions.txt -n HKL78//]],[,[],m[]]. -q 12

Wynik testu: Poprawne działanie programu, ponieważ nie ma ograniczeń co do wyboru nazwy użytkownika.

Test VI

Użytkownik wybrał plik wejściowy questions, nazwę użytkownika „HKL78//]],[,[],m[]]” oraz liczbę pytań do odpowiedzi 12.

Program wywołany w sposób ProjektQuiz.exe -o questions -n HKL78//]],[,[],m[]]. -q 12

Program zwraca błąd, gdyż nie znalazł odpowiedniego pliku.

```
No such file
Process returned 0 (0x0)   execution time : 2.293 s
Press any key to continue.
```

Wynik testu: Błędne działanie programu, gdyż program nie znalazł odpowiedniego pliku.

Test VII

Użytkownik wybrał plik wejściowy questions.txt, zamiast nazwy użytkownika umieścił przełącznik -a oraz zamiast liczby pytań umieścił przełącznik -x

Program wywołany w sposób ProjektQuiz.exe -o questions -a hello -x 13

Program kompiluje się normalnie używając wartości bazowych zmiennych.

```
C:\Users\Korwin\Desktop\University\Programowanie\Projekt Quiz\Projekt\ProjektQuiz\bin\Release\ProjektQuiz.exe: invalid option -- a
C:\Users\Korwin\Desktop\University\Programowanie\Projekt Quiz\Projekt\ProjektQuiz\bin\Release\ProjektQuiz.exe: invalid option -- x

Welcome user
to
the Quiz Program

-----
- Press N to start a new game
- Press any other key to quit the program
-----
```

Wynik testu: Poprawne działanie programu, przełącznik -q domyślnie wynosi 1, natomiast przełącznik -n domyślnie wynosi 'user'.

Desktop\University\Programowanie\Projekt Quiz\Projekt\ProjektQuiz\bin\Release\ProjektQuiz.exe" -o questions.txt -a hello -x 13

```
You've chosen to answer 1 questions from the question base questions.txt
You can answer all the questions
Your score will be the amount of correctly answered questions

Press any key to continue_
```