**Collegium Witelona Uczelnia Państwowa w Legnicy**

Wydział Nauk Technicznych i Ekonomicznych

Kierunek: Informatyka



**Sebastian Waga nr. Indeksu 43894**

**Rok studiów: 1**

**Grupa: 2(2)**

**Rok akademicki: 2023-2024**

**Projekt z przedmiotu „Podstawy programowania”**

**Prosta gra w zgadywanie liczb w języku C++**

**Prowadzący przedmiot**

**mgr inż. Marcin Tracz**

**Legnica 2023**

**Spis treści**

Krótki opis programu……………………………………………………………………..……………..3

Zrzuty ekrany pokazujące działanie programu…………………………………………………….……4

Zrzuty ekrany kodu i jego działanie ………………………………………………….……………….7

Język programu : C++

Użyty program: Visual Studio Code

Działanie programu: Program wyświetla menu, w którym dostępne są 4 opcje:

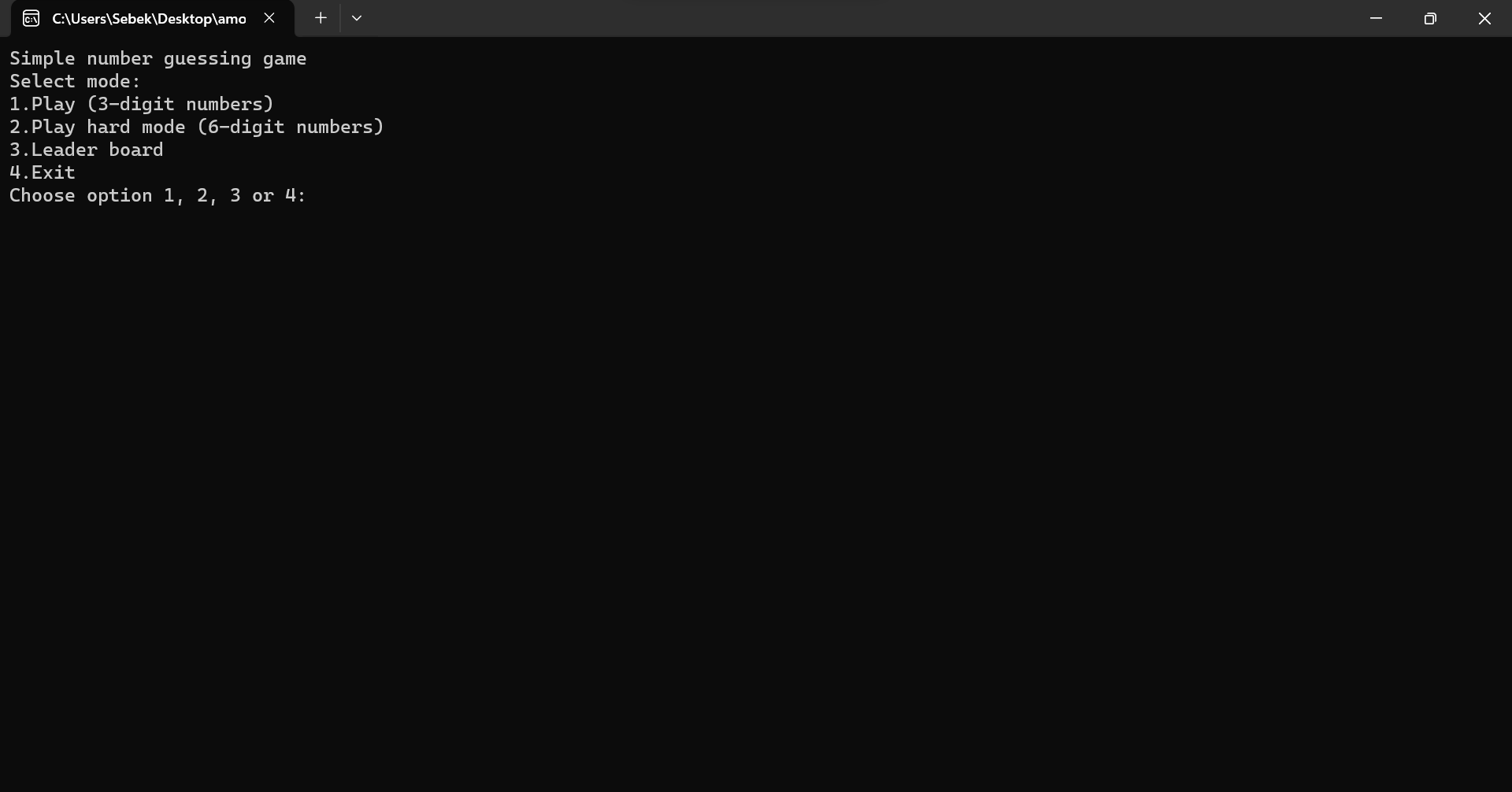
-Play (3-digit numbers) – program generuję liczbę trzycyfrową którą użytkownik musi odgadnąć wpisując trzy cyfry w konsole. Odgadnięte liczby wyświetlane są w kolorze zielonym, a nie poprawne w kolorze czerwonym. Po poprawnym odgadnięciu liczby program wyświetla gratulacje oraz prosi użytkownika o podanie nazwy użytkownika (składającej się z 5 cyfr lub liter). Po wpisaniu nazwy zapisywana jest ona razem z ilością prób oraz trybem gry w pliku „Leader\_board”.

-Play (6-digit numbers) – program generuję liczbę sześciocyfrową którą użytkownik musi odgadnąć wpisując sześć cyfry w konsole. Odgadnięte liczby wyświetlane są w kolorze zielonym, a nie poprawne w kolorze czerwonym. Po poprawnym odgadnięciu liczby program wyświetla gratulacje oraz prosi użytkownika o podanie nazwy użytkownika (składającej się z 5 cyfr lub liter). Po wpisaniu nazwy zapisywana jest ona razem z ilością prób oraz trybem gry w pliku „Leader\_board”.

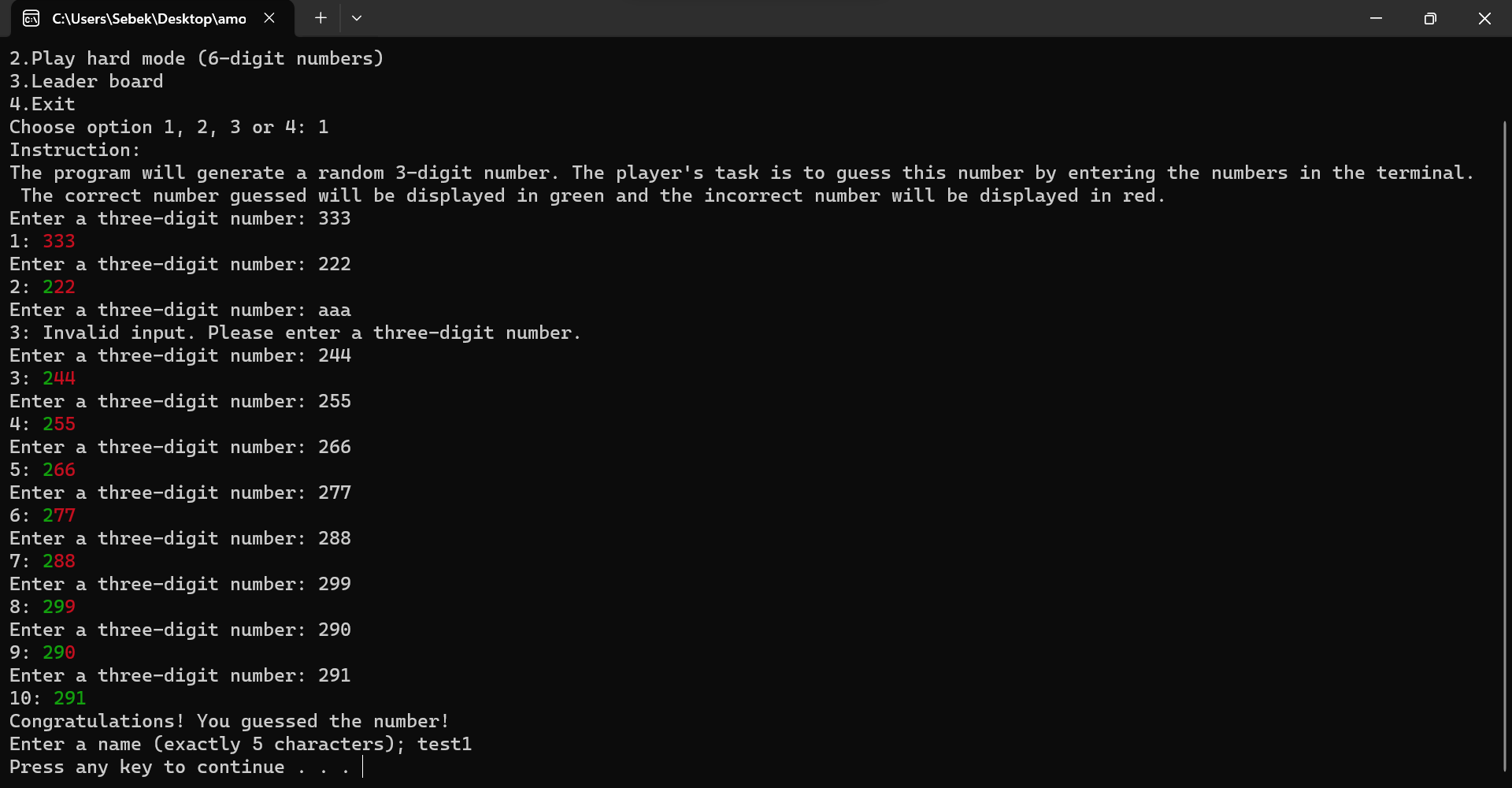
-Leader board - wyświetla zawartość pliku „Leader\_board”, jeśli istnieje i nie jest pusty.

-Exit – kończy działanie programu.

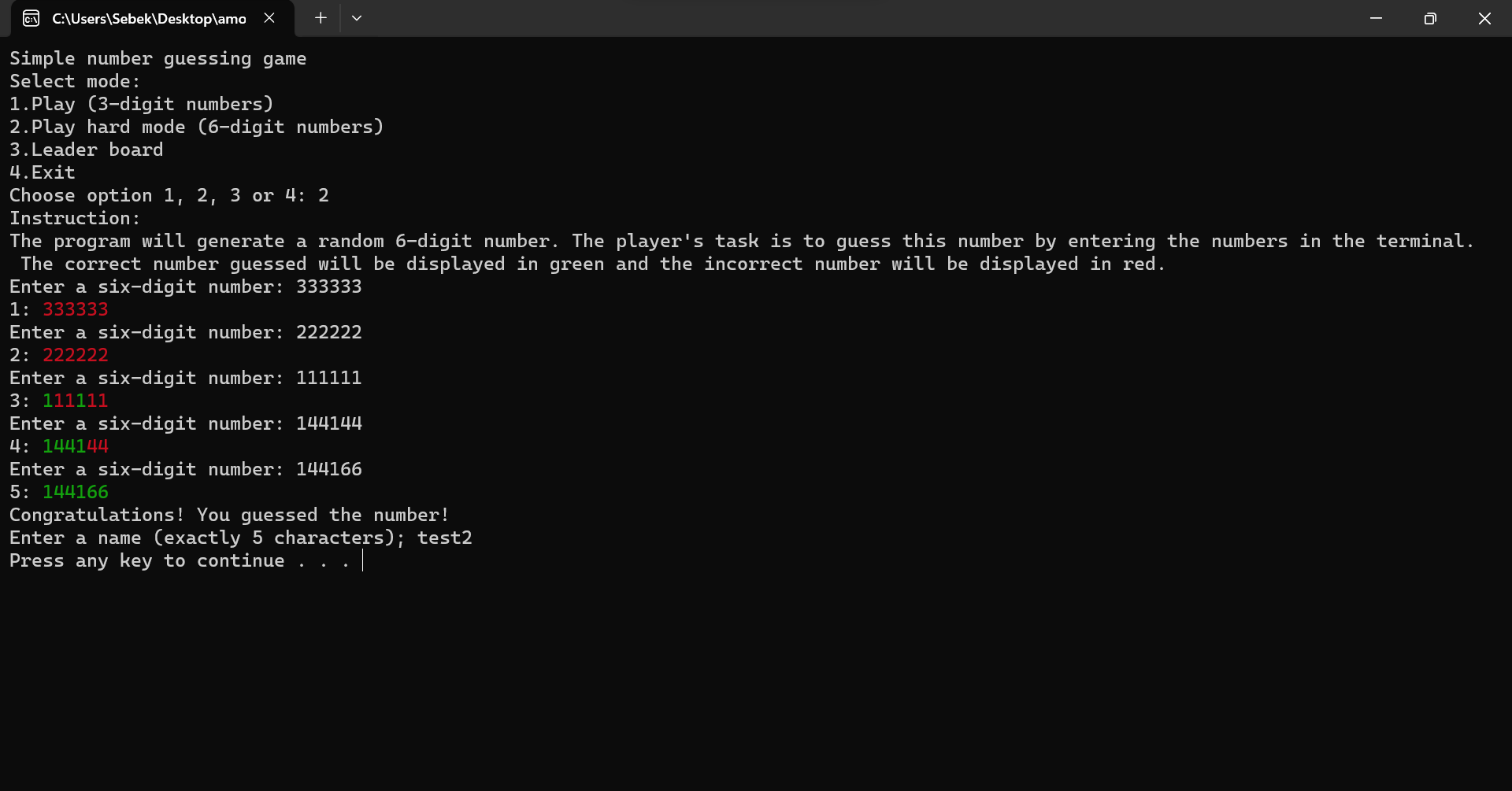
Menu główne:



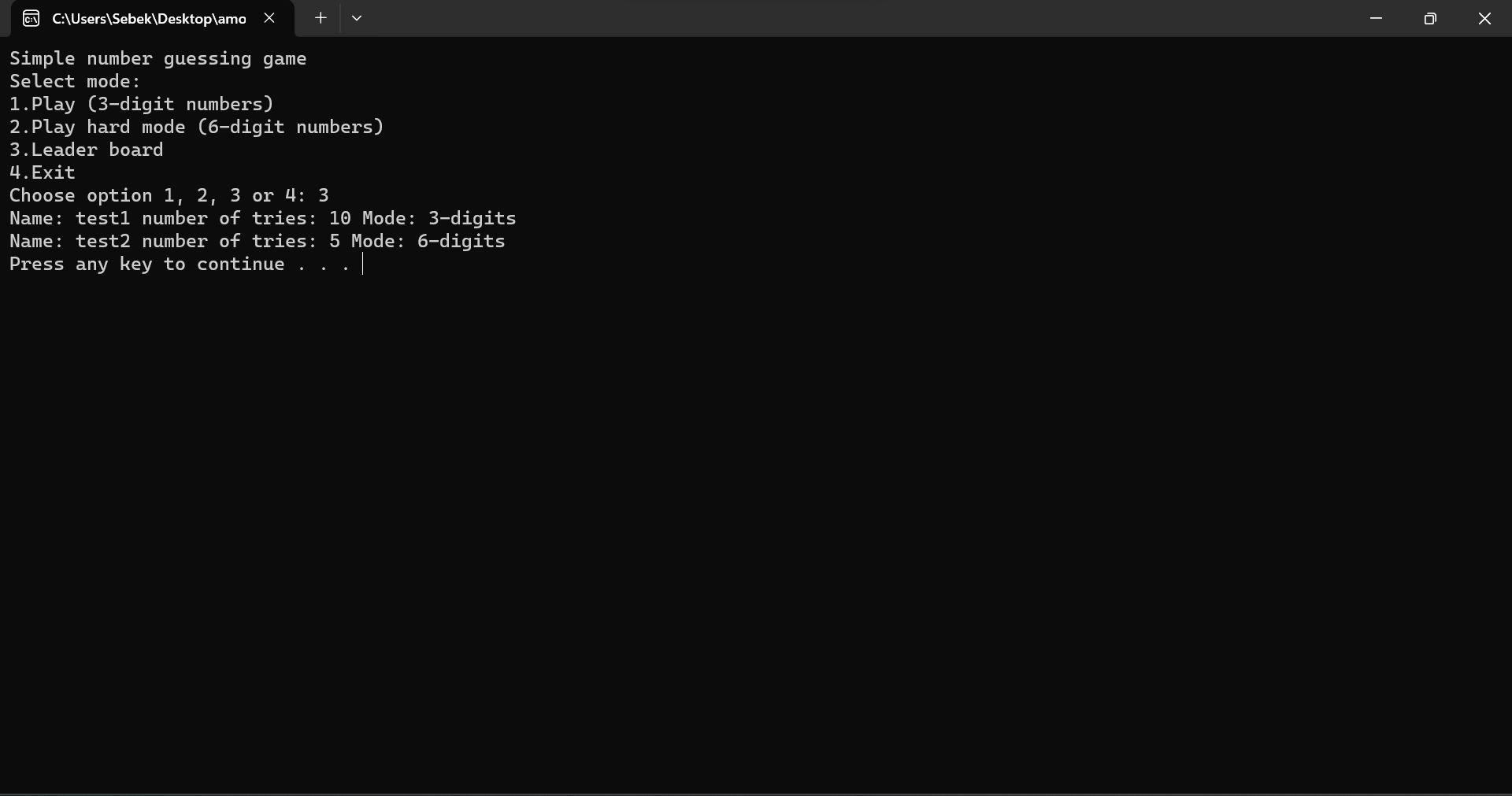
Rozgrywka w trybie 3-digit numbers. Po zakończeniu usuwany jest zawartość konsoli i ponownie wyświetla się menu.



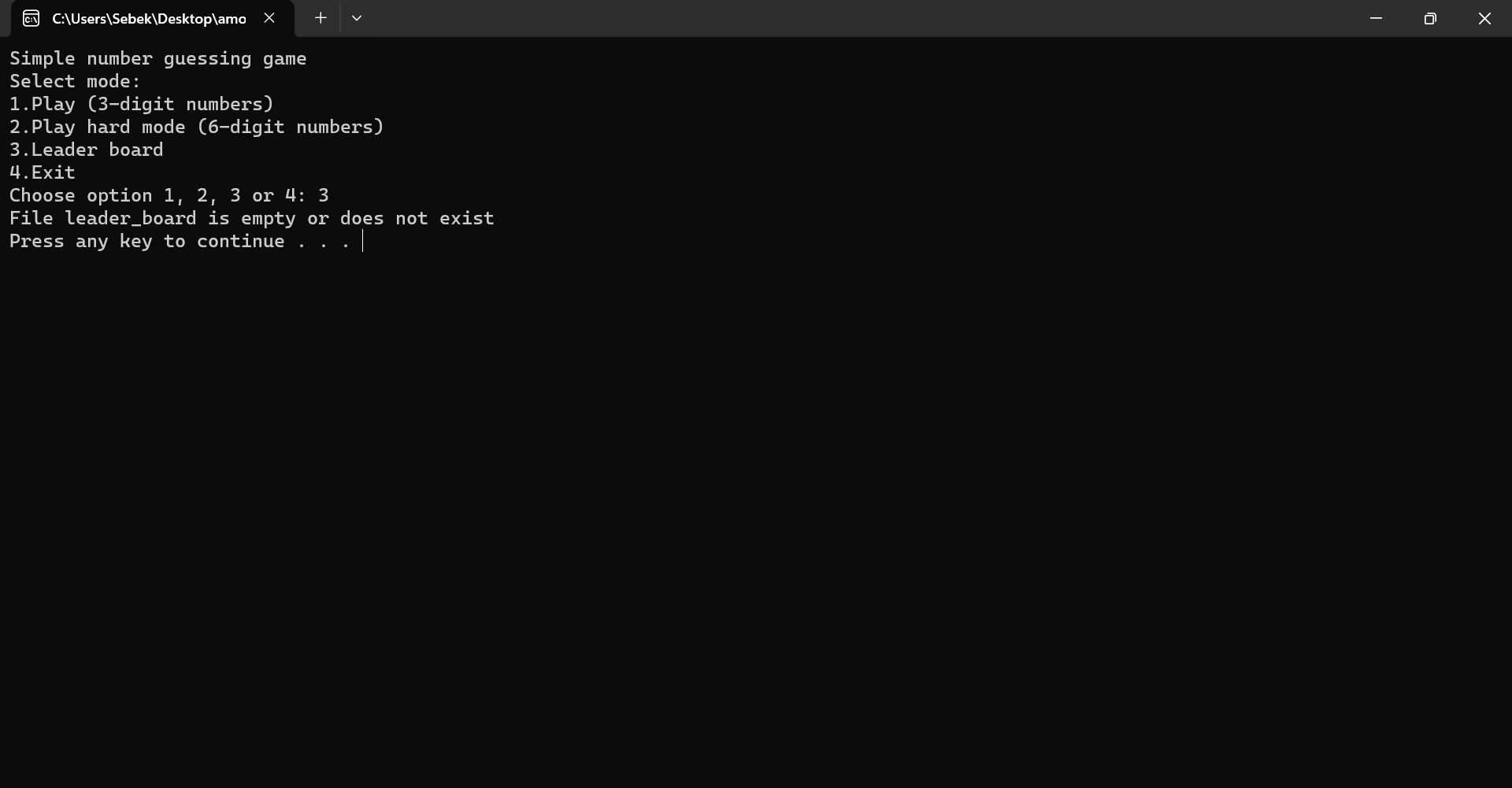
Rozgrywka w trybie 6-digit numbers. Po zakończeniu usuwany jest zawartość konsoli i ponownie wyświetla się menu.



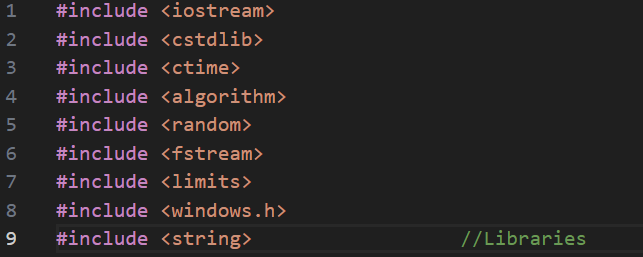
Wyświetlenie zawartości pliku „Leader\_board”



Powiadomienie w przypadku braku pliku lub braku zawartości pliku „Leader\_board”.

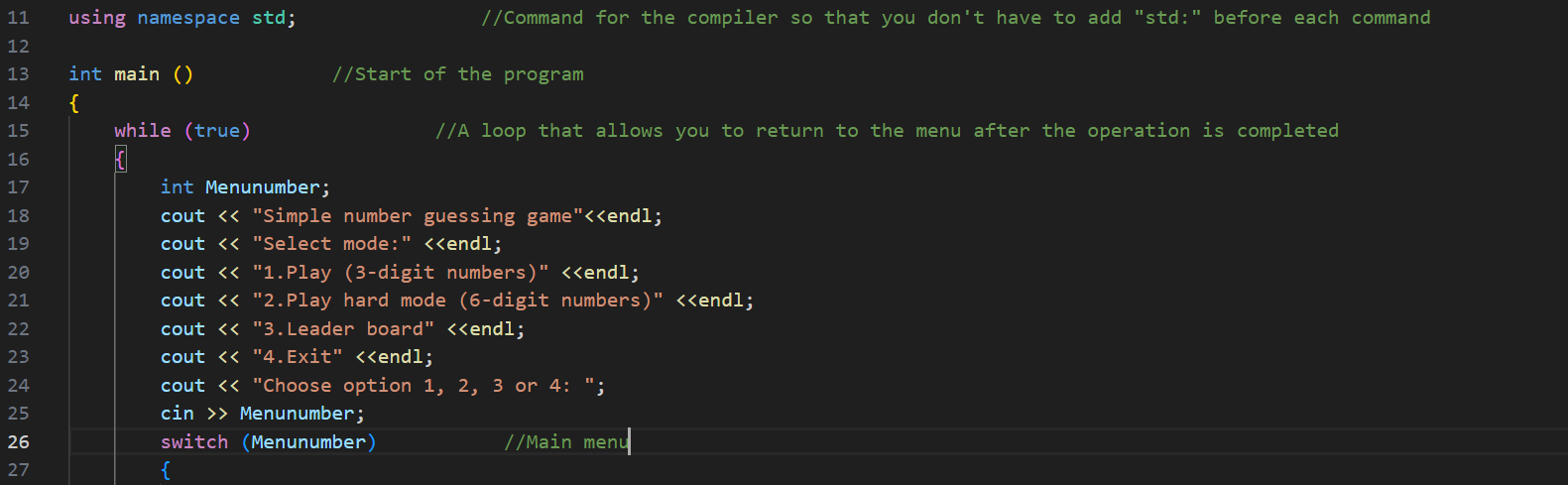


Biblioteki użyte w projekcie. Rozszerzają one możliwości języka umożliwiając użycie gotowych funkcji.

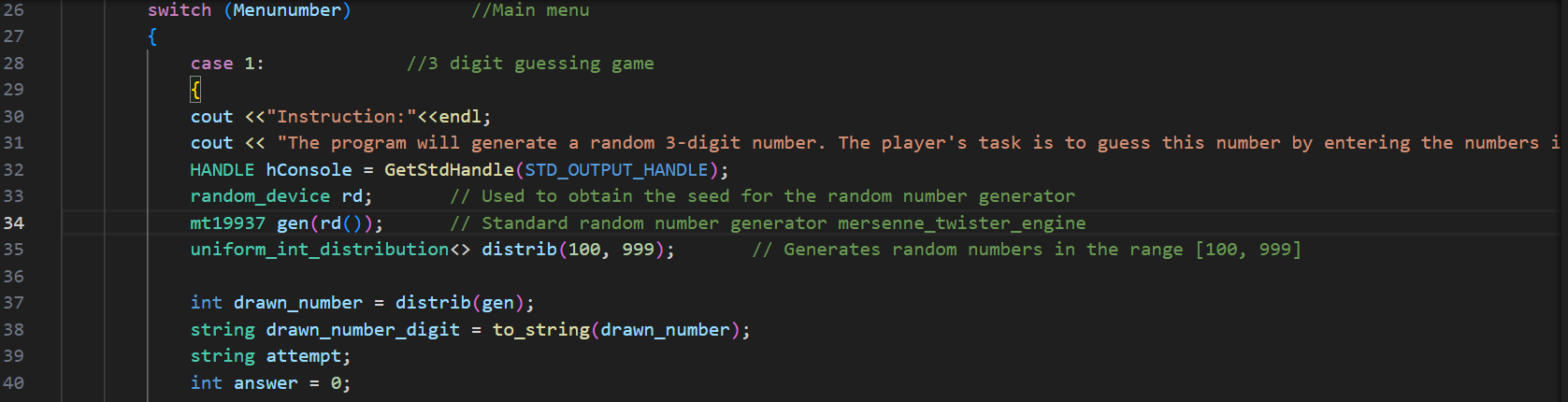


Dyrektywa „Using namespace std” umożliwia pisanie komend pokroju „cout” bez potrzeby dopisywania „std::cout”, ponieważ pokazuje ona kompilatorowi żeby w przypadku nie znalezienia danej nazwy szukał jej w przestrzeni „std”

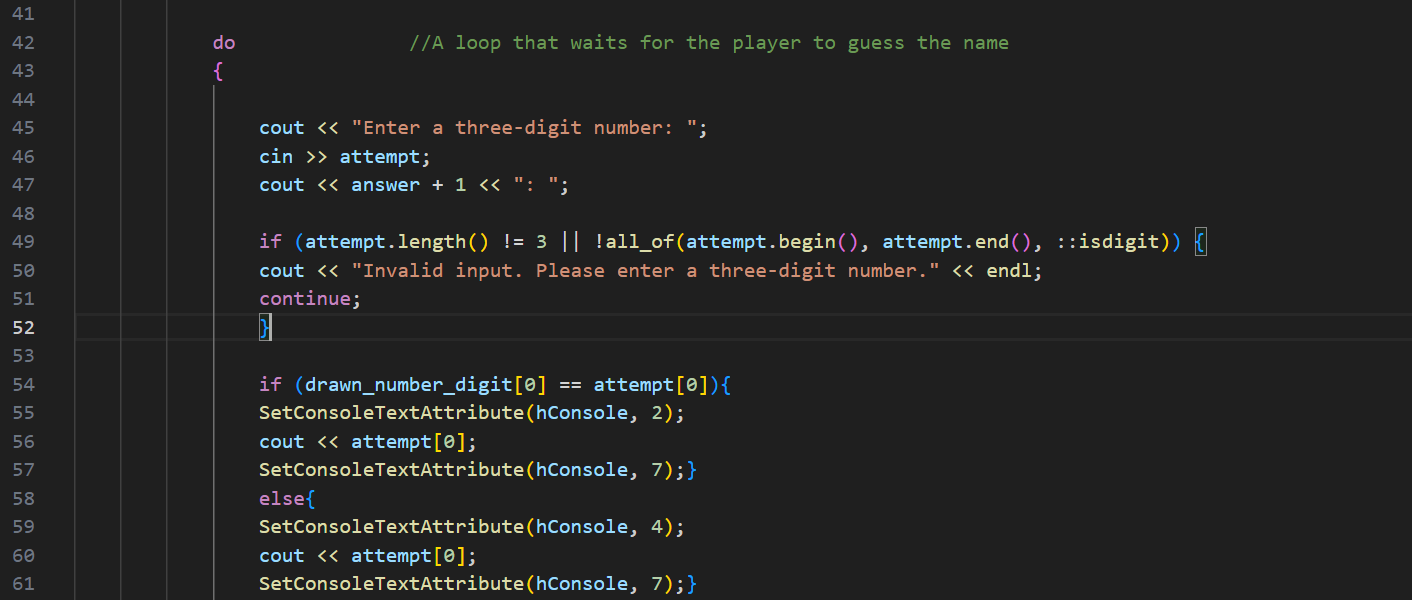


„Int main ()” rozpoczyna program. Całą gra znajduję się w pętli while (true) by umożliwić powrót do menu głównego po skończonej operacji oraz usuwanie wcześniej wpisanych komend. Poniżej dodana jest zmienna „Menunumber” której wartość odpowiada wybranej pozycji w menu. Od linijki 18 do 24 wyświetlane jest menu. Linijka 26 prosi użytkownika o wpisanie numeru menu a funkcja „switch” umożliwia wybór pozycji z menu poprzez przypisanie zmiennych „Menunumber” do różnych przypadków czyli do „case”.

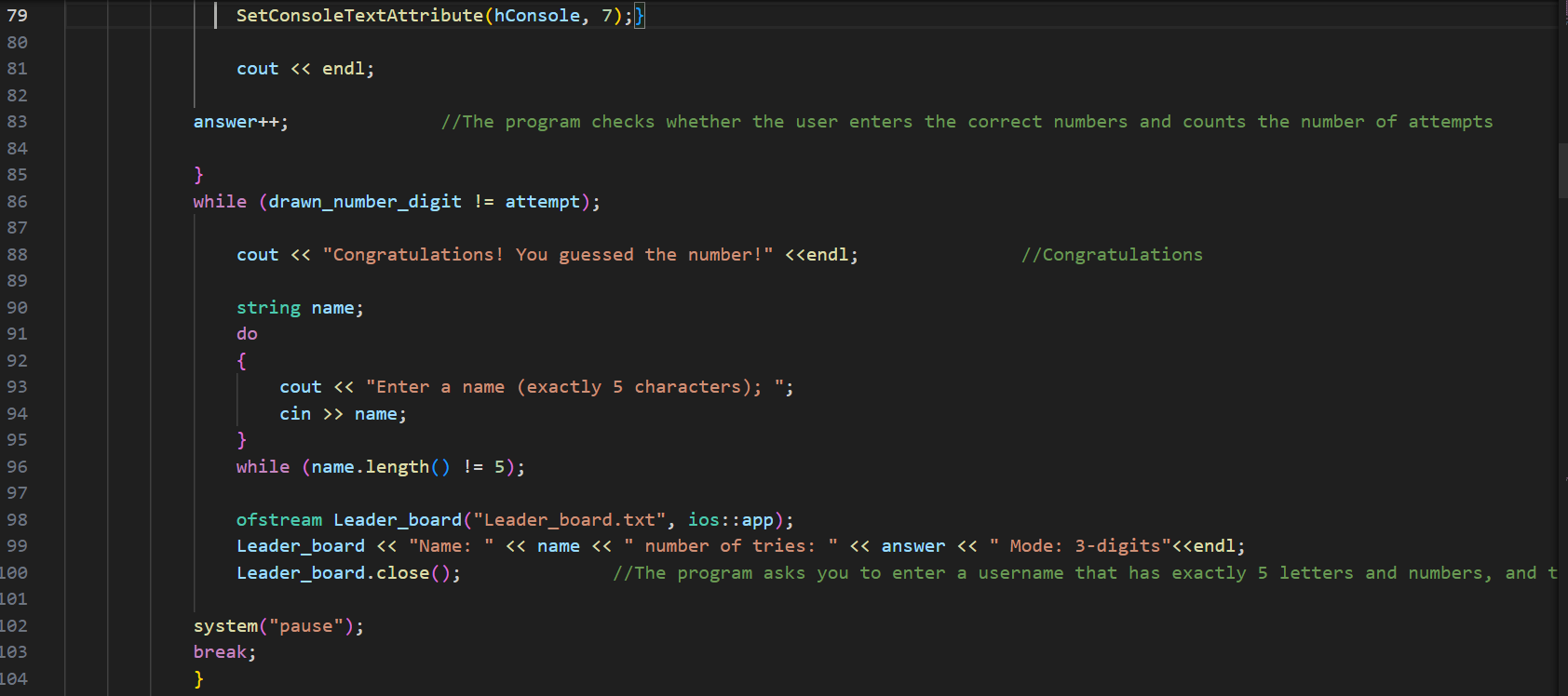
„case 1:” jest to prosta gra w zgadnięcie liczby trzycyfrowej. Program najpierw wyświetla instrukcje jak grać. Następnie za pomocą „Handle hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE) tworzymy „uchwyt” do urządzenia wyjściowego czyli konsoli i przechowujemy go zmiennej o nazwie hConsole. Będzie to potrzebne do zmiany koloru tekstu. W linijkach od 33 do 35 program najpierw generuję losowe „ziarno” i generuje liczbę z zakresu od 100 do 999. Następne program za pomocą "int drawn\_number = distrib(gen)" generuje liczbę za pomocą generatora liczb losowych „gen” i dystrybucji „distrib”. Wynik jest przechowywany w zmiennej „drawn\_number”. "string drawn\_number\_digit = to\_string(drawn\_number)" konwertuje liczbę "drawn\_number" na ciąg znaków (string) za pomocą funkcji "to\_string". Wynik jest przechowywany w zmiennej "drawn\_number\_digit". Komenda"string attempt" deklaruje pusty ciąg znaków o nazwie "attempt". A „int answer = 0” deklaruje zmienną „answer” typu „int” i inicjalizuje ją wartością 0.



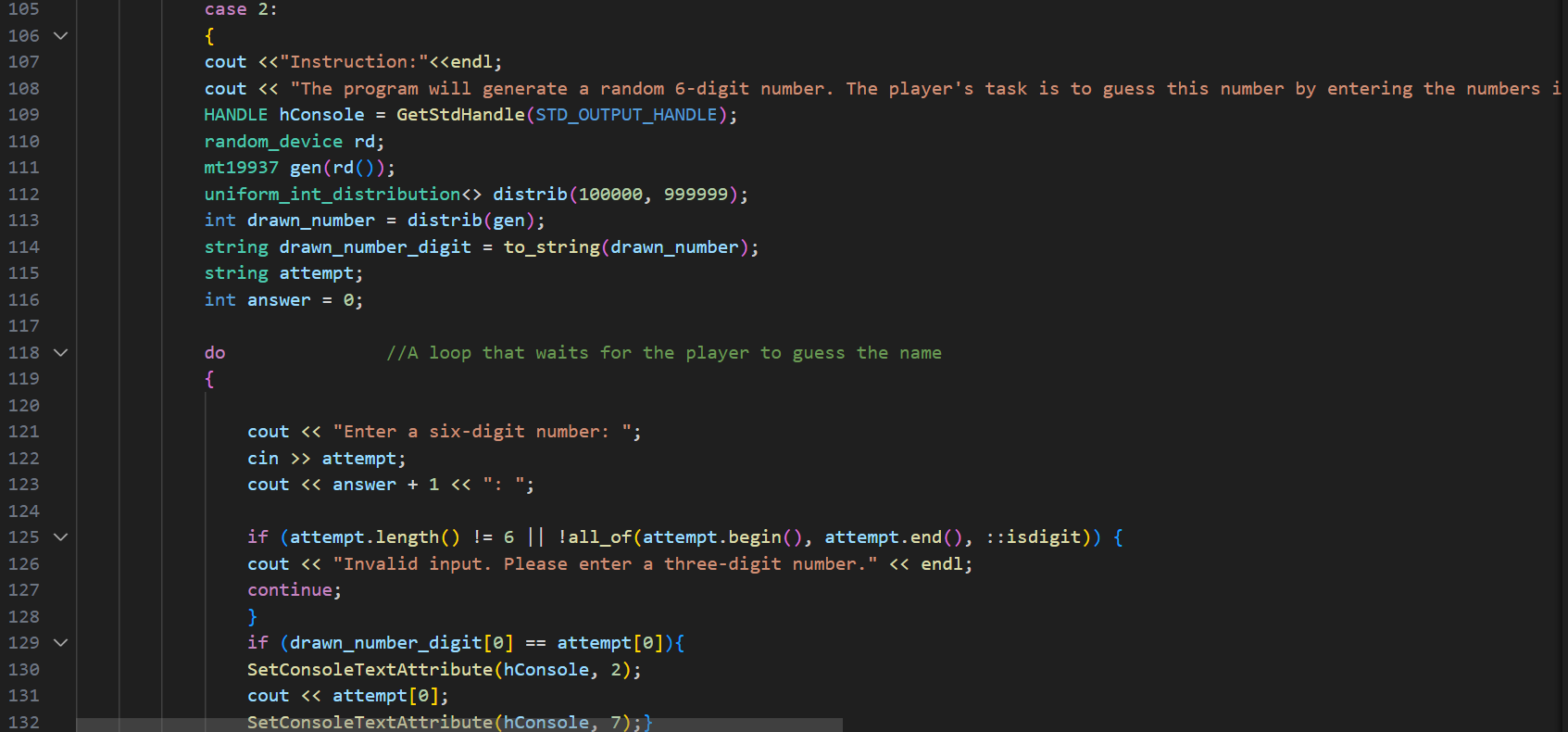
Całość następującego kodu zamknięta jest w pętli „do while” by gra zapętlała się do momentu poprawnie odgadniętej liczby. Następnie prosi użytkownika o podanie trzycyfrowej liczby oraz odlicza przy podaniu liczby ilość odpowiedzi których udzielił użytkownik. Następnie program sprawdza czy dopowiedz ma dokładnie trzy znaki i czy wszystkie znaki są cyframi, jeśli nie program pokazuje wiadomość o niepoprawnym zapisie. Program od linijki 54 do 79 sprawdza każdą cyfrę, czy jest identyczna z wygenerowaną liczbą losową. Jeśli tak liczba wyświetlona jest w kolorze zielonym, jeśli nie to cyfra wyświetlona jest w kolorze czerwonym.

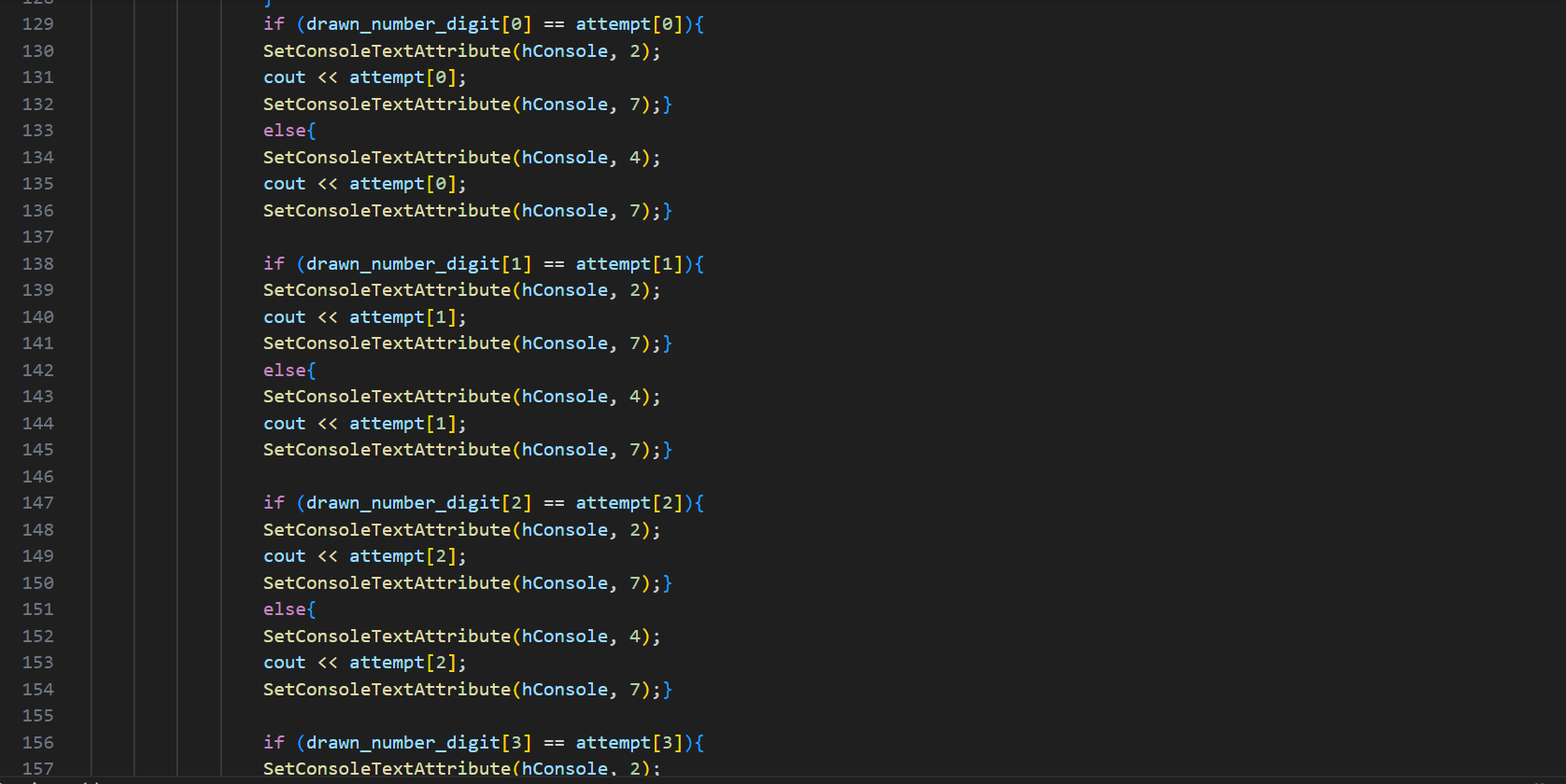


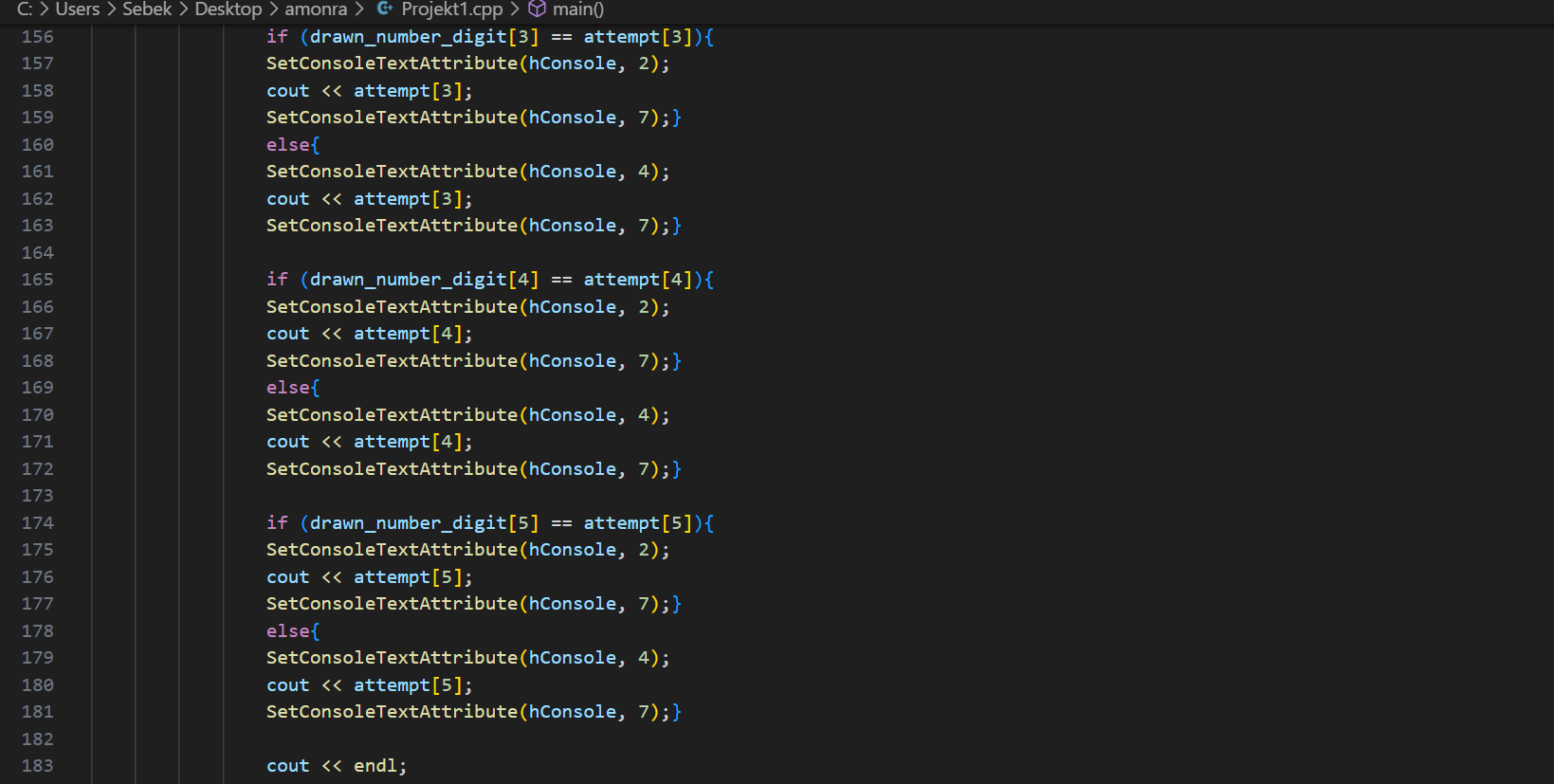
Gra kończy się w momencie odgadnięcia liczby przez użytkownika. Program wyświetla wtedy gratulacje oraz prosi o podanie nazwy której długość nie może przekroczyć limitu 5 znaków. Następnie program zapisuje nazwę, ilość prób zgadywania oraz tryb gry w pliku „Leader\_board.txt”. W przypadku braku takiego pliku zostaje on utworzony.

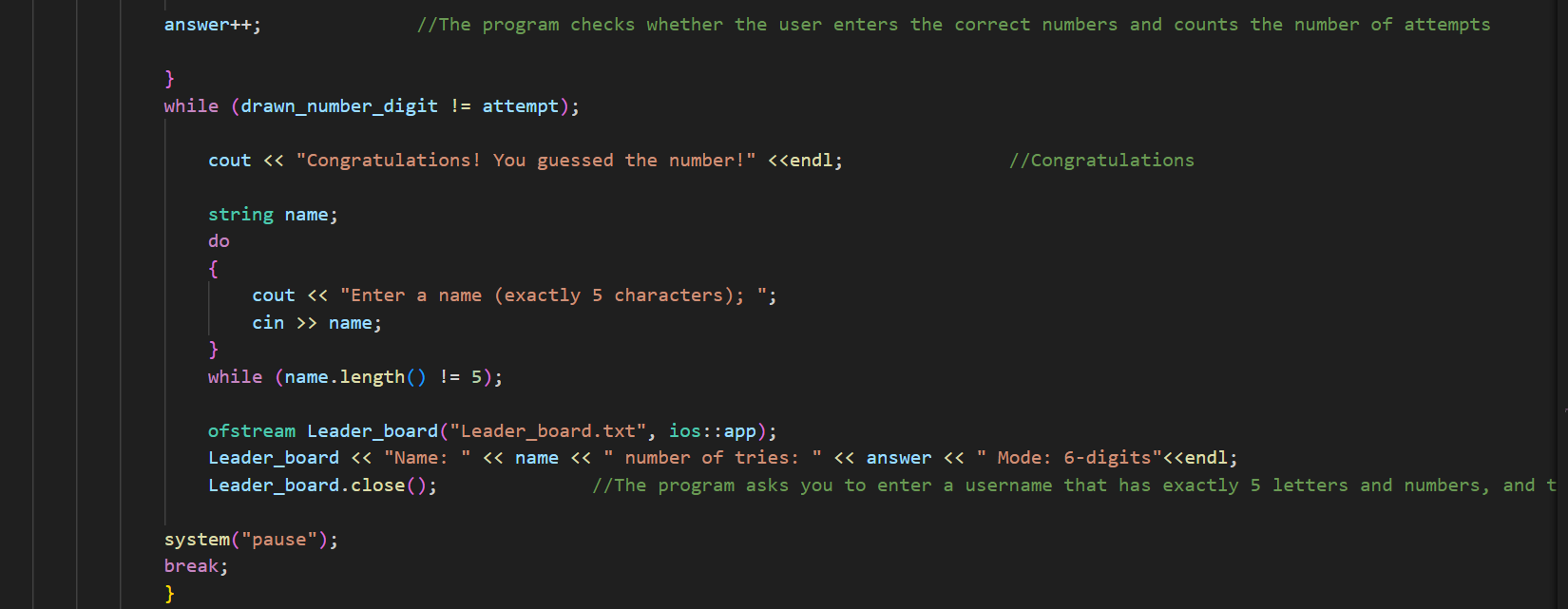


„case 2” zrobiony został w identyczny sposób, zostały tylko zmienione niektóre parametry programu.

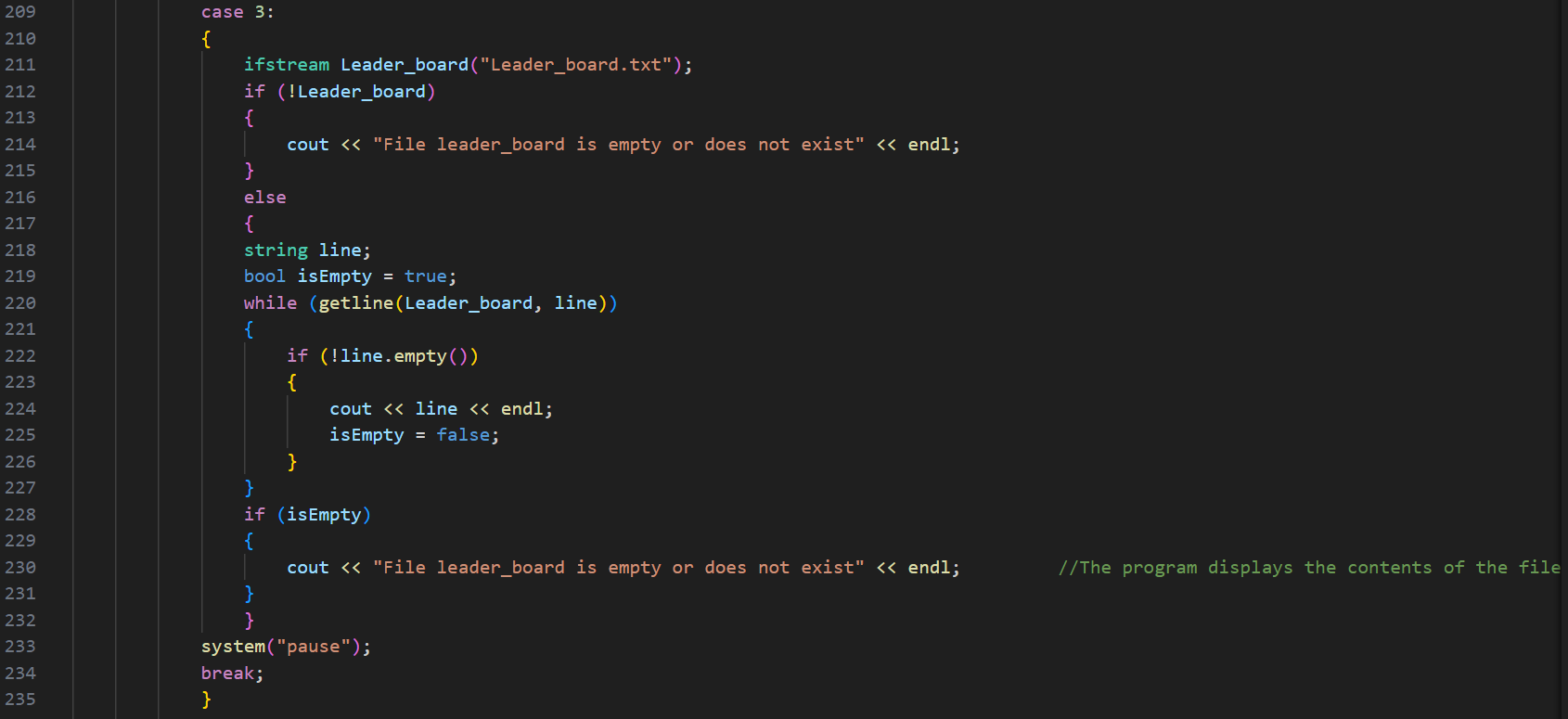




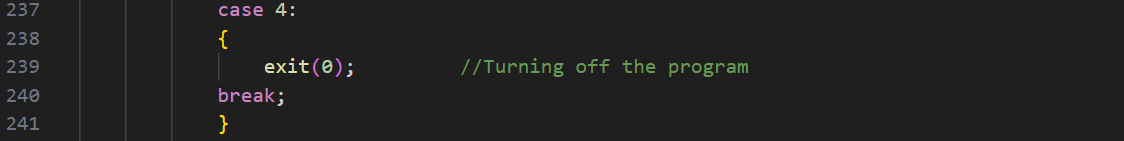




„case 3” rozpoczyna się komendą "ifstream Leader\_board("Leader\_board.txt")" która tworzy obiekt strumienia wejściowego o nazwie "Leader\_board", który otwiera plik o nazwie "Leader\_board.txt". Następnie program komendą "if (!Leader\_board)" sprawdza, czy plik "Leader\_board.txt" został poprawnie otwarty. Jeśli nie, wyświetla komunikat, że plik jest pusty lub nie istnieje. Jeśli plik został poprawnie otwarty wykonywana jest komenda „else”. Następnie"string line; bool isEmpty = true" deklaruje pusty ciąg znaków "line" i zmienną logiczną "isEmpty", która jest ustawiona na "true". W 220 linijce komenda "while (getline(Leader\_board, line))" czyta plik linia po linii. Jeśli linia nie jest pusta, wyświetla ją na ekranie i ustawia "isEmpty" na "false". Jeśli po przeczytaniu całego pliku "isEmpty" nadal jest "true", oznacza to, że plik był pusty. Wtedy wyświetla komunikat, że plik jest pusty lub nie istnieje.



„case4” kończy działanie programu



„default” wykonuje się gdy zostanie użyty w menu inny znak niż od 1 do 4. W takim przypadku program najpierw sprawdza za pomocą "if (cin.fail() || Menunumber < 1 || Menunumber > 4)" czy wystąpił problem podczas próby wczytania wartości do zmiennej "Menunumber" lub czy wartość "Menunumber" jest mniejsza od 1 lub większa od 4. Następnie czyści błąd dla polecenia „cin” i wyświetla napis „Wrong input”. Następnie czyści dane w terminalu.

