

Анотація

Курсова робота є прикладом програмної реалізації гри «Бики та корови».

Алгоритм був реалізований у вигляді консольної програми. Для реалізації алгоритму була обрана мова програмування C++. Інтерфейс у консольному вікні програми був виконаний з допомогою бібліотеки «windows.h» та елементів псевдографіки. Програма складається з наступних основних пунктів: декілька меню, одне з яких – робота із обліковими записами, інше – власне сама гра (на певному акаунті), контактна інформація щодо розробника, а також правила гри.

Продукт курсової роботи має розважальний характер. Гра призначена для людей будь якого віку та сприяє розвитку логічного мислення.

Annotation

The course work is an example of software implementation of the game "Bulls and Cows".

The algorithm was implemented as a console program. The C ++ programming language was chosen to implement the algorithm. The interface in the console window of the program was made using the library "windows.h" and pseudo-graphics elements. The program consists of the following main items: several menus, one of which - working with accounts, the other - the game itself (on a specific account), contact information about the developer, as well as the rules of the game.

The product of the course work is entertaining. The game was created for people of all ages and promotes the development of logical thinking.

ЗМІСТ

ЗАВДАННЯ	
ВСТУП.....	
РОЗДІЛ 1	4
1.1. Власне задача, інформація з джерел	4
1.2. Уточнена постановка задачі на розробку програмного забезпечення.....	5
РОЗДІЛ 2 Розробка програмного продукту.....	6
2.1. Метод розв’язку задачі	6
2.2. Алгоритм розв’язку задачі	9
РОЗДІЛ 3 Опис розробленого програмного продукту	17
3.1. Опис головних структур і змінних програми.....	17
3.2. Опис головних функцій програми.....	20
3.3. Опис інтерфейсу	21
3.4. Результати роботи програмного продукту	27
ВИСНОВКИ	
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	

ВСТУП

Актуальність: В наш час розвиток ігрова індустрія активно розвивається. Деякі продукти використовуються задля інтелектуального розвитку, деякі – для відпочинку та розваг, тощо. Мій продукт – приклад розважальної та дійсно дієвої для розумового розвитку гри.

Метою мого курсового проекту є створення програми, у якій має бути реалізований принцип відгадування чисел за тим чи іншим принципом. Робота має бути виконана у вигляді консольної програми. Інтерфейс гри бажано не нагромаджувати функціоналом, проте зробити приємним та захоплюючим.

Завдання: На основі курсу програмування та алгоритмічних мов, а також аналізу літературних джерел, було створено консольну програму «Бики та корови», досліджено основний принцип гри, використано елементи псевдографіки за допомогою бібліотеки «windows.h» для приємних ігрових відчуттів. За допомогою певних джерел, були досліджені та реалізовані правила гри у консольному вікні. Приємним бонусом є можливість керувати (створювати, авторизуватися, видаляти) обліковими записами із власними внутрішніми рейтингами.

Практичне значення одержаних результатів: За допомогою виконаного проекту можна покращити навички логічного мислення. Зокрема, максимально раціонально підходити до вирішення певної проблеми. До того ж, ще й з друзями, у яких, власне, можуть бути індивідуальні облікові записи.

Використане програмне забезпечення: При написанні курсової роботи було використане середовище розробки Dev-C++ та операційна система Windows 10, для підготовки та оформлення курсової роботи використано Microsoft Word 2013.

Структура роботи: Робота складається із вступу, трьох розділів, висновків, додатків та списку використаних джерел.

РОЗДІЛ 1

Постановка задачі

1.1. Власне задача, інформація з джерел

Під час розробки частини гри, у якій програма за певним алгоритмом повинна віднайти загадане людиною число, було знайдено невелику кількість алгоритмів, які зводилися до принципу сортування. Натомість було власноруч винайдено алгоритм роботи програми у такій ситуації. Він є трохи незвичним, якщо порівнювати із певними іншими варіантами, де пропонують аналізувати, наприклад, парність, потім подільність на 3, 5, тощо, а потім вже підбирати серед решти. Або ж підбирати без зайвих зусиль все підряд. Було винайдено компроміс, про який трохи далі. Слід зазначити, що правила гри теж певним чином відрізняються від базових - була змінена реалізація биків. Програма не просто повертає їхню кількість (кількість цифр, що співпали за положенням), вона буде поступово (відповідно до гри гравця) виводити відгадані цифри, що, на мою думку, є не менш цікавим варіантом, а корів (кількість співпадінь) було збережено відповідно до класичних правил.

Якщо порівнювати мету власної програми та класичну реалізацію, то тут відмінності відсутні – за надану кількість кроків потрібно відгадати чотиризначне число із унікальними цифрами. Якщо не встигаєте ви чи встигає програма, то, відповідно від рейтингу віднімається 1 очко (може бути тільки невід'ємним, за поразки 0 залишається незмінним). Якщо встигаєте ви або не встигає програма, то до рейтингу додається 1 очко.

Зауважимо, що не було знайдено джерел, де дана гра була б реалізована із можливим керуванням обліковими записами. Як правило, людині одразу пропонують зіграти. Інколи можна зустріти різні рівні складності. Оскільки реалізація різних рівнів складності – шаблонний код, то цю функцію не було взято до уваги.

1.2. Уточнена постановка задачі на розробку програмного забезпечення

Як вже згадувалось, проект полягає у створенні консольної програми, яка окрім власне ігрового процесу, має ще й початкове меню із керуванням обліковими записами. У цьому меню потрібно реалізувати створення акаунту, його видалення, вхід в існуючий та вихід із програми. Після вдалої авторизації програма викликає функцію ігрового(другого) меню, де вже має бути реалізований наступний принцип:

Комп'ютером або ж людиною задане чотиризначне число без повторень, яке протилежній стороні потрібно вгадати за надану кількість кроків. На кожному кроці комп'ютер має повертати (якщо відгадує людина) інформацію про кількість угаданих цифр, і кількість таких з них, що стоять на своєму місці (останнє поняття називається биками, що мають бути замінені на виведення відгаданих цифр замість зірочок на потрібному місці). Також повинен бути реалізований випадок, де задає число людина, а відгадує його програма за певним алгоритмом, що написаний власноруч.

Також потрібно реалізувати рейтингову систему, відповідно до якої будуть нараховуватися та зніматися бали за гру – за виграшу додається 1 очко, за програшу – 1 очко віднімається (реалізувати невід'ємність кількості очок).

Функція із реалізацією ігрового процесу повинна працювати із певним обліковим процесом, щоб коректно відображати поточний рейтинг на певному обліковому записі.

Під час переходу між меню потрібно реалізувати в тому числі і повернення у перше меню зі збереженням поточного рейтингу певного облікового запису.

Користувач повинен мати змогу ознайомитися із правилами гри, а також із даними засновника задля контактного зв'язку за потреби.

РОЗДІЛ 2

Розробка програмного продукту

2.1. Метод розв'язку задачі

Для написання даної програми була використана мова *C++ та бібліотека «windows.h»*.

Спочатку людині пропонують обрати вже існуючий акаунт, створити новий, видалити існуючий або ж вийти з гри взагалі.

Опишемо основний функціонал першого меню: (рис.1.1):

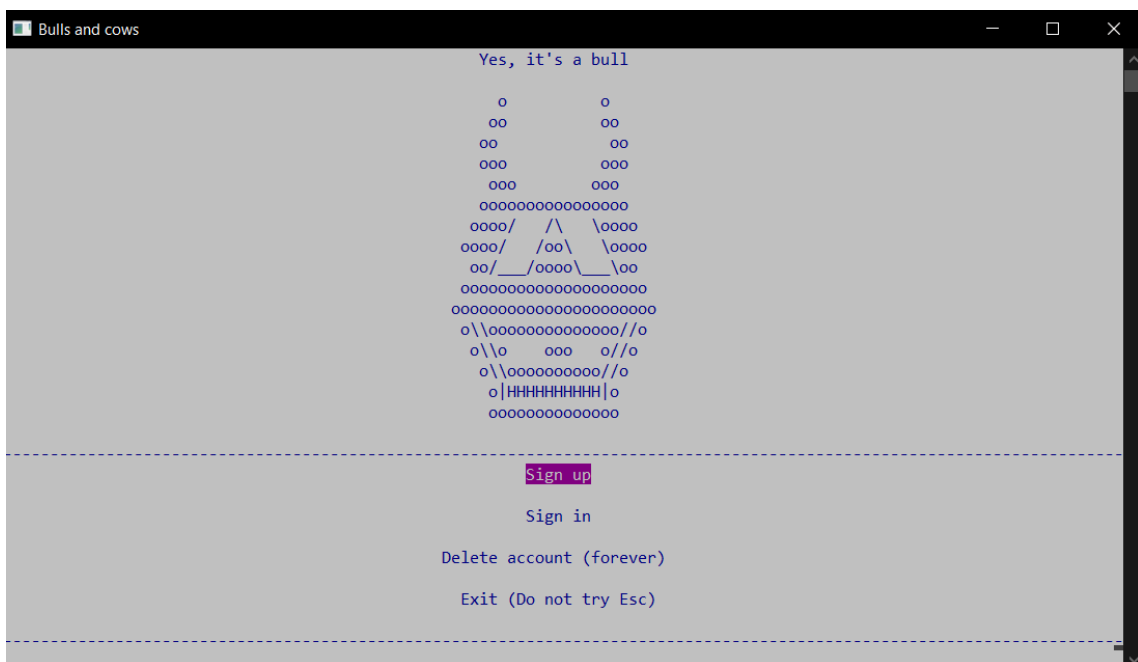


Рис.1.1. Перше меню

- 1) Максимальна кількість існуючих акаунтів – 5;
- 2) Гравець не може авторизуватися або видалити якийсь обліковий запис, якщо їхня кількість дорівнює 0;
- 3) Якщо ви обрали якийсь акаунт для авторизації, але забули до нього пароль, то ви можете вийти у меню, якщо введете “@”.
- 4) Якщо ви вирішили видалити якийсь акаунт, то після введення його ніку вам треба буде підтвердити свої дії. Так 0 вас поверне у меню, а 1 видалить акаунт, якщо він існує.
- 5) При виході із програми акаунти не зберігаються. При поверненні у

перше меню із другого, всі існуючі акаунти та їхній поточний рейтинг залишаються незмінними. Гравець матиме можливість авторизуватися знову.

Була реалізована гра в обидва боки – гравець може відгадати число, загадане комп'ютером і, натомість, комп'ютер може відгадувати число, загадане гравцем.

У другому меню передбачений наступний функціонал (рис.1.2):



Рис.1.2.Друге (ігрове) меню

- 1) Загадати число комп'ютеру;
- 2) Відгадати число комп'ютера;
- 3) Дізнатися правила гри;
- 4) Дізнатися контактні дані засновника;
- 5) Повернутися до першого меню(зі збереженням поточних даних облікового запису).

Зауважимо, що усі малюнки із літер “o” виконані окремо в текстових файлах, тому програма їх просто зчитує.

Перший пункт – справа удачі. Було створено доволі незвичний алгоритм: для перемоги комп'ютера достатньо та необхідно хоча б 1 раз відгадати перші 3 цифри чотиризначного числа. Кількість спроб

дорівнює 200. 200 разів комп'ютер хаотично намагається підібрати правильні цифри. Якщо не встигає – гравець виграє – і 1 очко додається до рейтингу. Якщо програма встигла хоча б 1 раз – гравець програє та, відповідно, віднімається від рейтингу 1 очко (якщо він є додатнім, інакше 0 залишається незмінним). Розглянемо комбінаторний зміст алгоритму:

A – імовірність програти, якщо вибрати довільне тризначне число. Звідси оскільки нас цікавлять тризначні числа з унікальними цифрами, то $9 * 9 * 8 = 648$ – це кількість можливих чисел. Нас цікавить імовірність вгадати 1 з них – те, яке ввів користувач.

Зазначимо, що $A = \frac{647}{648}$, а тепер знайдемо імовірність A' програти

всі 200 разів (спроб): $A' = A^{200} = \left(\frac{647}{648}\right)^{200} \approx 0.7343$. Зрозуміло, що

імовірність відгадати хоча б 1 раз потрібне число $= 1 - A' = 0.2657$.

Звідси маємо імовірність виграшу комп'ютера, що дорівнює 0.2657

А тому імовірність програшу людини становить ≈ 0.2657 .

Тепер щодо принципу роботи частини програми, що відповідає за гру навпаки – за гру, де комп'ютер загадує чотиризначне число, а гравець намагається його відгадати.

На кожному кроці повертається кількість корів (кількість співпадінь між цифрами, що були введені та переліком цифр, що міститься у числі, тобто порядок розташування значення не має) та замість зірочок повертаються відгадані цифри – це зроблено замість повернення биків. На мою думку, так навіть цікавіше, бо по функціонали поняття биків та корів дуже схожі, а таким чином програма має певне різноманіття.

Якщо за 10 кроків гравець встигає відгадати, то людина виграє та нараховується 1 очко до рейтингу, інакше – зчитується із назвою “Defeat” і 1 очко віднімається від рейтингу (від'ємним рейтинг не може бути).

2.2. Алгоритм розв'язку задачі

Структурна блок-схема алгоритму роботи програми зображена на рисунку 2.1.

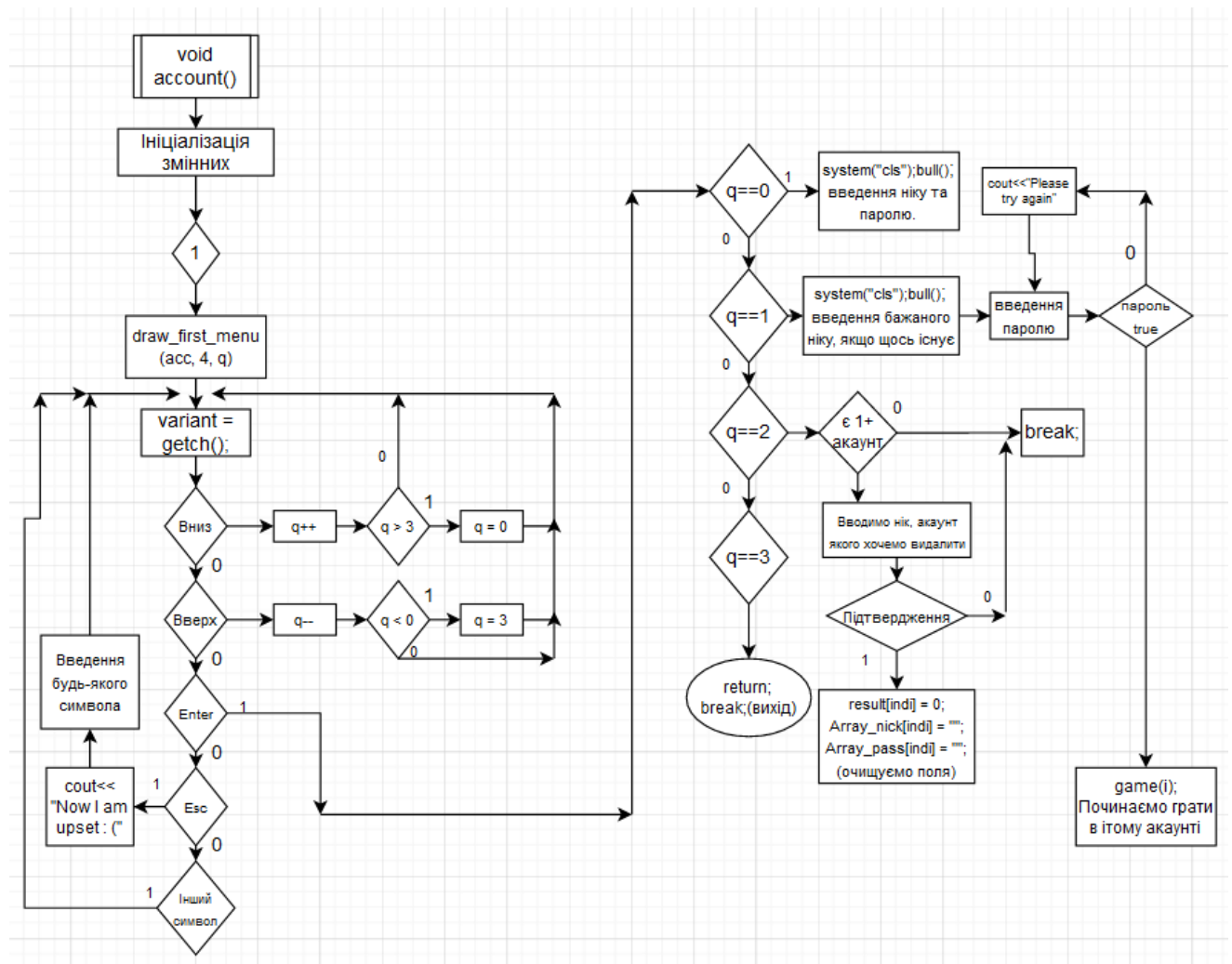


Рис.2.1 Структурна блок-схема алгоритму роботи програми
void account() – виклик функції у main.

Суть цієї функції полягає у роботі із обліковими записами (акаунтами).

Після ініціалізації змінних та необхідних рядків для ніку та паролю, відбувається вхід у нескінченний цикл. Вийти із нього можливо тільки перейшовши за останнім пунктом (першого) меню, де за допомогою операторів return та break обірветься цикл та програма закінчує свої дії. При натисканні Esc гравець лише отримає наступне повідомлення "Now I am upset(" та після натискання будь-якої клавіші повернеться до першого меню.

Якщо обрати варіант "Sign up", то спочатку відбудеться перевірка, чи кількість акаунтів не перевищила 5. Якщо так, то доведеться видалити хоча б 1

із них (інакше створити додаткові неможливо) за допомогою пункту "Delete account (forever)", про нього трохи пізніше. Якщо ні, то гравцю запропонують ввести нік та пароль - Рис.2.2.

Enter your nickname

3

Enter your password(do not enter only @)

3

Remember your data and do not give it to anyone:

Your nickname: 3

Your password: 3

Для продовження натисніть будь-яку клавішу . . .

Рис.2.2. Реалізація "Sign up".

Відбудеться перевірка, чи не існує вже такого ж самого ніку (це необхідно для авторизації, бо там людина вводить бажаний нік). Якщо такий нік вже існує, то треба буде вводити новий. Якщо такого не існує, то треба буде ввести пароль, що не має містити єдиний символ "@" (декілька таких символів вже можна(це зроблено для того, щоб при авторизації можна було виходити за допомогою введення цього символу, якщо забув пароль до цього акаунту)). Припустимо, що гравець зареєструвався та у програмі налічується хоча б 1 обліковий запис.

Тепер за допомогою "Sign in" гравець може увійти у бажаний акаунт та розпочати на ньому гру. (Якщо акаунтів немає, то відбудеться повернення у меню). Якщо вони є, то просто треба обрати бажаний нік та ввести до нього пароль - Рис.2.3.

Please log in or sign up (Enter nick):

1) 1

2) 2

3) 3

nickname -> 3

Please enter password: 3

Success!!!! Press any key to continue the game

Для продовження натисніть будь-яку клавішу . . .

Рис.2.3.Реалізація " Sign in ".

За допомогою “@” можна повернутися у меню. Якщо пароль правильний, то відбудеться перехід у друге меню, тобто викликається функція `void game(i)`, де `i` – це номер акаунту, у якому людина буде грати, це зроблено заради збереження рейтингу на кожному акаунті навіть при виході у перше меню. Збереження рейтингів реалізовано за допомогою глобальної ініціалізації масиву із 5 рейтингів. Якщо пароль не співпав, то доведеться вводити ще раз.

Остання нерозглянута вкладка меню – вкладка "Delete account (forever)", функціонал якої вже був частково розкритий, адже усі функції між собою пов'язані - Рис.2.4.

```
1) 1
2) 2
3) 3
Please enter a nick:
1
-----
Are you sure you want to delete this item? Yes -> 1; No -> 0
-----
1
-----
Well done!
-----
Для продолжения нажмите любую клавишу . . .
```

Рис.2.4. Реалізація згоди у вкладці "Delete account (forever)".

Додаємо інформацію. Якщо гравець вводить нік, якого не існує у списку, то необхідно буде зробити введення ніку ще раз. Якщо нік співпав, то гравцю нададуть підтвердження дії. При введенні 1 акаунт видаляється (очищуються його дані, а під час виведення акаунтів, у відповідних для цього пунктах меню, виводяться тільки непорожні облікові записи) та відбувається повернення у перше меню. При введенні 0 нічого не видаляється, відбувається побажання приємного дня та повернення у перше меню зі збереженням акаунтів та їхніх даних - Рис.2.5.

```
2) 2
3) 3
Please enter a nick:
3
-----
Are you sure you want to delete this item? Yes -> 1; No -> 0
-----
0
-----
Have a nice day!
-----
Для продолжения нажмите любую клавишу . . .
```

Рис.2.5. Реалізація відмови у вкладці "Delete account (forever)".

Нижче наведемо реалізацію функції `void game(i)`. Це, власне, і є сама гра - Рис.2.6.

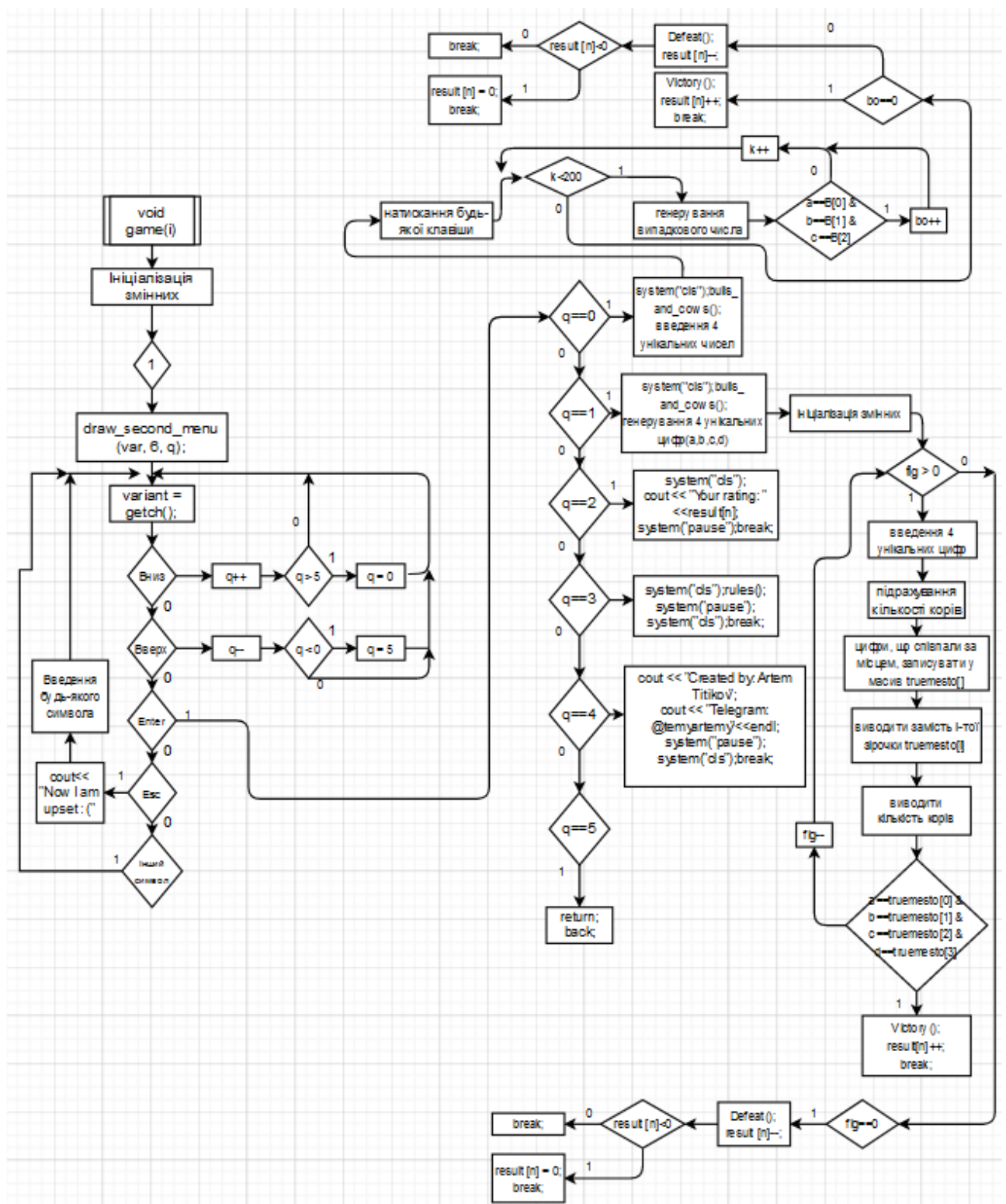


Рис. 2.6. Структурна блок-схема алгоритму роботи функції `void game(i)`.

`void game(i)` – виклик функції для *i*-го непорожнього елемента масиву із обліковими записами.

Після ініціалізації необхідних змінних відбувається вхід у нескінченний цикл, вихід із якого можливий лише за переходу за вкладкою меню "Exit (Do not try Esc)".

Гравець може обрати будь-який пункт із 6:

- 1) "Your chyslo". У цій вкладці спочатку відбувається зчитування файлу bulls_and_cows.txt. Далі іде введення першої цифри, яка має бути від 1 до 9. Потім вводиться друга, яка має бути від 0 до 9 та при цьому не має дорівнювати першій. Далі гравцю треба ввести третю, яка не співпадає ні з другою, ні з першою, до того ж, вона має бути від 0 до 9. Закінчується введення числа четвертою цифрою, що не має співпадати із жодною минулою цифрою, а також яка має бути від 0 до 9. За будь-якого порушення правил введення числа, спрацьовує перевірка та відбувається повторне введення цифри. Для наочності введене число виводиться на екран. За натискання будь-якої клавіші починається відгадування програмою уведеного гравцем числа. Всього надається 200 спроб. Для програшу гравця програмі достатньо підібрати хоча б 1 раз правильно перші 3 цифри уведеного числа. У цьому і полягає незвичність алгоритму – таким чином зберігається баланс між тривалістю підбору числа та імовірністю виграшу машини. Якщо достатню умову було виконано принаймні 1 раз, то гравець програє і зчитується файл "Defeat.txt", а також від рейтингу віднімається 1 очко (рейтинг може бути лише невід'ємним, тому від'ємні значення замінюються на 0). Якщо достатня умова не виконалася жодного разу, то гравець виграє, зчитується файл під назвою "Victory.txt", до рейтингу акаунту додається 1 очко. Натисканням будь-якої клавіші відбудеться повернення у друге (ігрове) меню - Рис.2.7 та Рис.2.8.

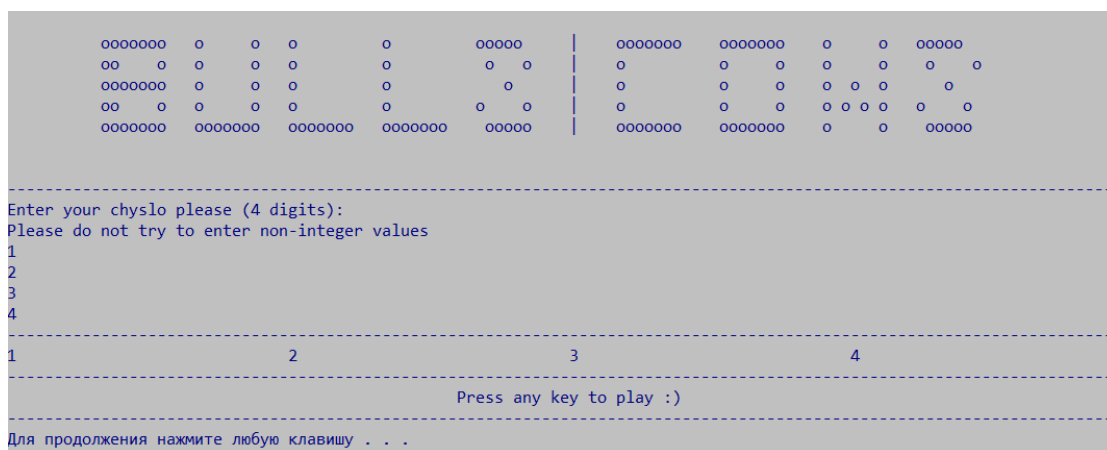


Рис.2.7. Реалізація пункту "Your chyslo".

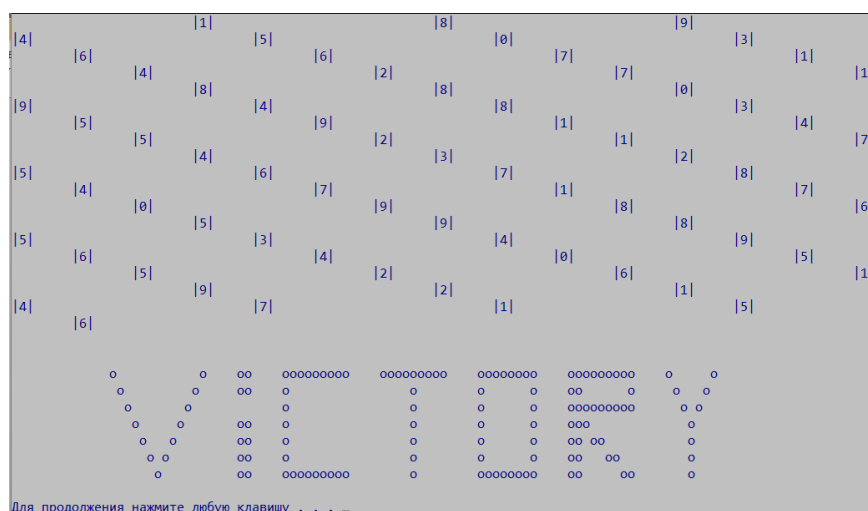


Рис.2.8. Реалізація пункту "Your chyslo".

- 2) "Computer's chyslo". У цій вкладці спочатку відбувається зчитування файлу bulls_and_cows.txt. Далі програма генерує по чергово кожну цифру із 4 відповідно до відомих вже правил. Потім відбувається ініціалізація необхідних для подальшої роботи програми змінних – і запускається цикл `while(f > 0){...}`, що буде допомагати рахувати решту спроб та знову вводити нове число, доки не закінчатся надані 10 спроб або гравець не відгадає загадане програмою число. На кожній ітерації виводиться кількість корів та виводяться відгадані вже цифри, що, на мою думку, більш цікаво, аніж аналогічна до корів реалізація биків. Якщо протягом 10 спроб гравець відгадав загадане число, то зчитується файл під назвою "Victory.txt", а також додається 1 очко до поточного рейтингу. Якщо людина не встигла за надані 10 кроків відгадати число, то, відповідно, зчитується файл

“Defeat.txt”, а також від рейтингу буде віднято 1 очко, якщо той більший за 0 (інакше так 0 і залишиться). Натисканням будь-якої клавіші відбудеться повернення у друге (ігрове) меню - Рис.2.9, Рис.2.10, Рис.2.11.

```

0000000 0 0 0 0 0 00000 | 0000000 0000000 0 0 00000
00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0000000 0 0 0 0 0 0 0 0 0 0 0 0 0
00 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0000000 0000000 0000000 0000000 00000 | 0000000 0000000 0 0 00000

* * * *
The number of attempts -> 10
Please, enter the first digit:

```

Рис.2.9. Реалізація пункту "Computer's chyslo".

```

* * * *
The number of attempts -> 10

Please, enter the first digit:
1

Please, enter the second digit:
2

Please, enter the third digit:
3

Please, enter the fourth digit:
4

* * * * 4
Cows -> 2

The number of attempts -> 9

Please, enter the first digit:

```

Рис.2.10. Реалізація пункту "Computer's chyslo".

```

Please, enter the first digit:
9

Please, enter the second digit:
6

Please, enter the third digit:
1

Please, enter the fourth digit:
4

9 6 1 4

Cows -> 4

0 0 00 00000000 00000000 00000000 00000000 0 0
0 0 00 0 0 0 0 00 0 0 0
0 0 00 0 0 0 0 00000000 0 0
0 0 00 0 0 0 0 0 0 0
0 0 00 0 0 0 0 0 0 00 00
0 0 00 0 0 0 0 0 0 00 00
0 0 00000000 0 00000000 00 00 0

Для продолжения нажмите любую клавишу . . .

```

Рис.2.11. Реалізація пункту "Computer's chyslo".

- 3) "Rating". У цій вкладці відбувається виведення поточного рейтингу на екран. Натисканням будь-якої клавіші відбудеться повернення у друге (ігрове) меню - Рис.2.12.

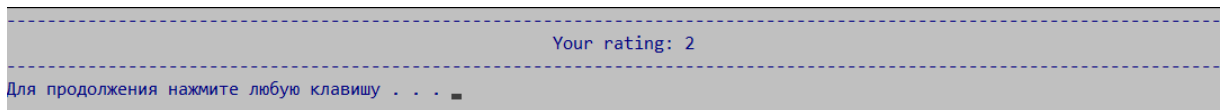


Рис.2.12. Реалізація пункту "Rating".

- 4) "Rules for game". У цій вкладці зчитується файл під назвою "rules.txt", у якому наведені правила гри. Натисканням будь-якої клавіші відбудеться повернення у друге (ігрове) меню - Рис.2.13.

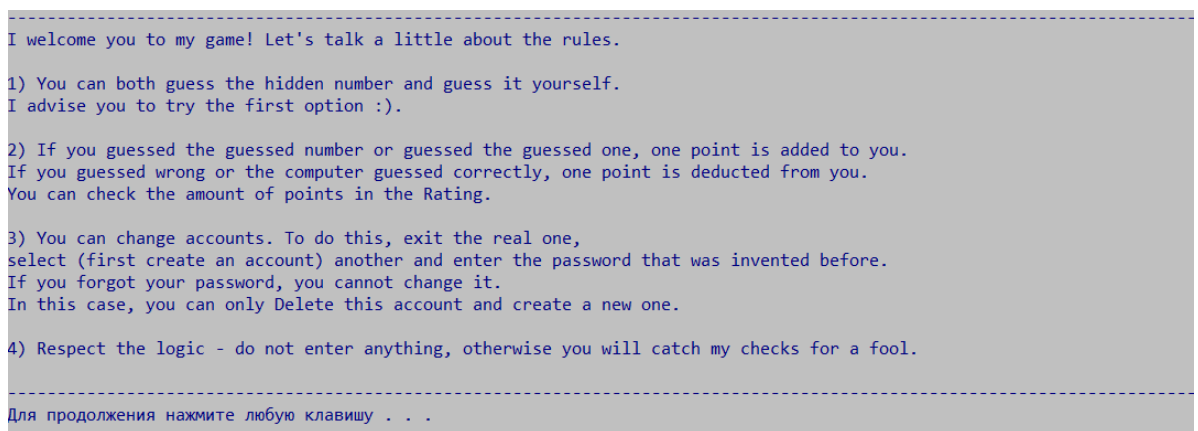


Рис.2.13. Реалізація пункту "Rules for game".

- 5) "Owner". Відбувається наступне: `cout << "Created by: Artem Titikov"<<endl;` `cout << "Telegram: @temyartemy"<<endl;` Після цього натисканням будь-якої клавіші відбудеться повернення у друге (ігрове) меню - Рис.2.14.

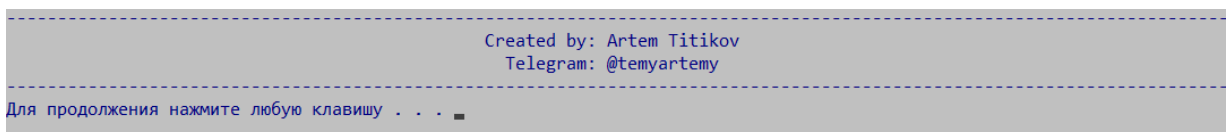


Рис.2.14. Реалізація пункту " Owner ".

- 6) "Exit (Do not try Esc)". Ця вкладка дозволяє повернутися у перше меню зі збереженням поточних даних певного облікового запису на глобальному рівні, що дозволить у будь-який момент часу авторизуватися знову та продовжити гру (за винятком закриття програми).

РОЗДІЛ 3

Опис розробленого програмного продукту

3.1. Опис головних структур і змінних програми

Опис головних структур і змінних програми наведено в таблиці 3.1.

Таблиця 3.1. Опис головних змінних програми

№	Змінна	Тип змінної	Призначення змінної
1	handle	HANDLE	Дескриптор консольного вікна, який використовується для виведення на екран першого та другого меню.
2	csbuf	CONSOLE_SCREEN_BUFFER_INFO	Доступ до консольного вікна.
3	variant	int	З її допомогою відбувається переміщення по меню і вибір певних пунктів.

№	Змінна	Тип змінної	Призначення змінної
4	q	int	В залежності від нього відкривається певний пункт меню.
5	indi	int	З її допомогою фіксується потрібний елемент масиву (індекс), який ми хочемо видалити (очистити).
6	sure	int	Підтвердження чи спростування щодо видалення.

№	Змінна	Тип змінної	Призначення змінної
7	bo	int	Індикатор, за допомогою якого ми помічаємо, чи підбрала програма число, задане гравцем.
8	flg	int	Кількість спроб (кроків), що залишилися у гравця для відгадування слова.
9	sovpadenie	int	Кількість корів.
10	result[n]	int	Глобально ініціалізований масив рейтингів для кожного акаунта.
11	Array_nick[5]	string	Масив для нікнеймів
12	Array_pass[5]	string	Масив для паролів.

3.2. Опис головних функцій програми

Опис головних функцій програми наведено в таблиці 3.2.

Таблиця 3.2. Опис головних функцій програми

№	Синтаксис	Опис
1	<code>int main()</code>	Функція, що запускає увесь ігровий процес – функцію <code>void account()</code> .
2	<code>void account()</code>	Функція, що керує обліковими записами та після авторизації викликає <code>void game(i)</code> для гри на <i>i</i> -тому акаунті.
3	<code>void game(int)</code>	Функція, що керує ігровим процесом на акаунті, який приймає як параметр. Дозволяє повернутися у до минулої функції - <code>void account()</code> .
4	<code>void draw_first_menu(char* [], int, int);</code>	Малює перше меню, що передбачає взаємодію із обліковими записами.
5	<code>void draw_second_menu(char* [], int, int);</code>	Малює друге меню, що передбачає ігровий процес.

3.3. Опис інтерфейсу

На початку роботи програми у консольному вікні відображається головне меню, яке містить в собі 4 пункти, а назва вікна містить назву гри. Керування у меню реалізоване за допомогою клавіш KeyUp та KeyDown, вибір – за допомогою клавіші Enter. Меню реалізоване за допомогою псевдографіки, як показано на рисунку 3.1.



Рис. 3.1. Інтерфейс головного меню.

Тут за умовчанням вказується на перший пункт – “Sign up”. Якщо натиснути Enter, то очиститься консоль, зчитується файл “bull.txt”, виконається певна алгоритмічна процедура та відбудеться певне штрихування за шириною консольного вікна та отримаємо наступний результат (Рис.3.2):



Рис.3.2. Інтерфейс пункту “Sign up” після введення певних даних.

Після натискання будь-якої клавіші, відбудеться очищення екрану та повернення до першого меню. Далі можемо обрати наступний пункт меню – “Sign in”, при переході до якого знову очиститься екран та зчитується файл “bull.txt” (Рис.3.3, Рис.3.4.):

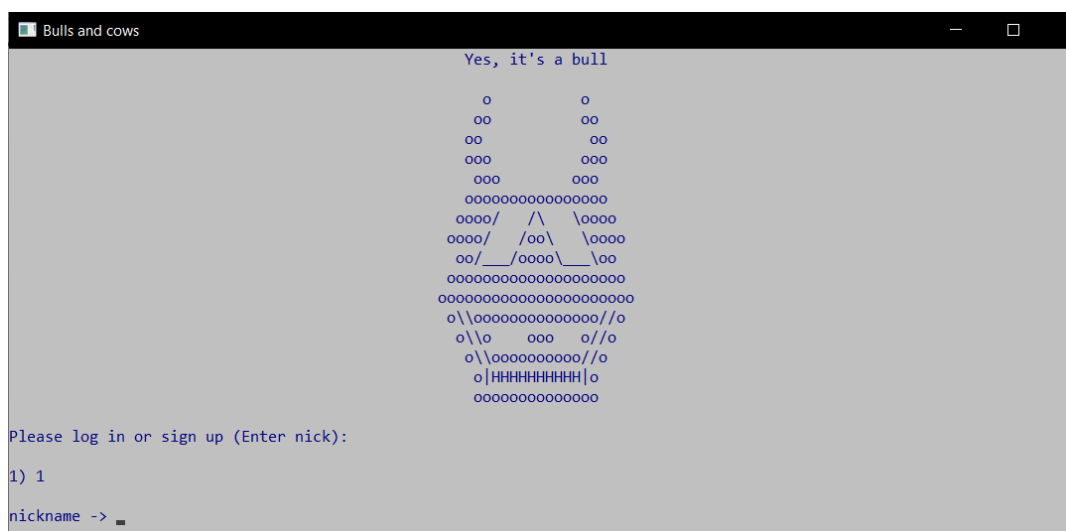


Рис.3.3. Інтерфейс пункту “Sign in” перед введенням даних.



Рис.3.4. Інтерфейс пункту “Sign in” після введення даних.

Зауважимо, що за натискання будь-якої клавіші тепер (після вдалої авторизації) відбудеться перехід до другого (ігрового) меню – викликається функція `game(i)`, у внутрішній частині якої викликається функція `void draw_second_menu(char* [], int, int)`, після чого маємо наступну ситуацію (Рис.3.5.):

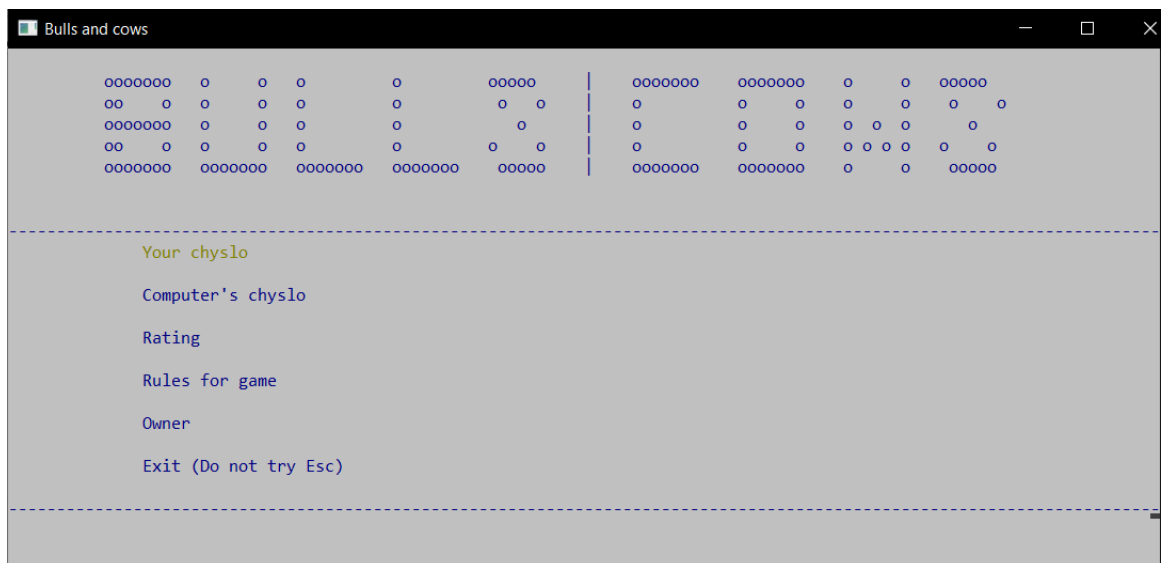


Рис.3.5. Інтерфейс другого меню (ігрового).

Тут функціонал схожий із першим меню, адже різняться вони лише зовнішнім виглядом, начинка ж доволі схожа. За вибору першого ж пункту, очистимо екран, зчитуємо “`bulls_and_cows`” та введемо бажане чотиризначне число і натиснемо будь-яку клавішу: (Рис.3.6., Рис.3.7., Рис.3.8.):

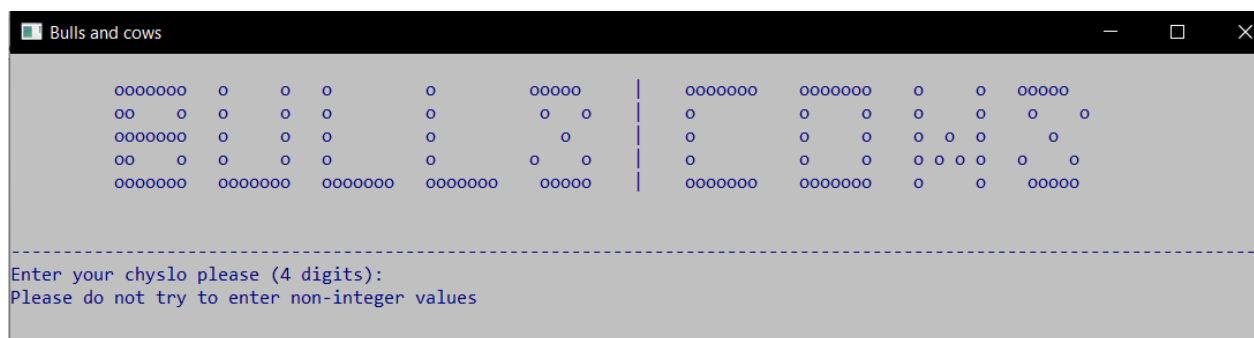


Рис.3.6. Інтерфейс пункту “Your chyslo” після його відкриття.

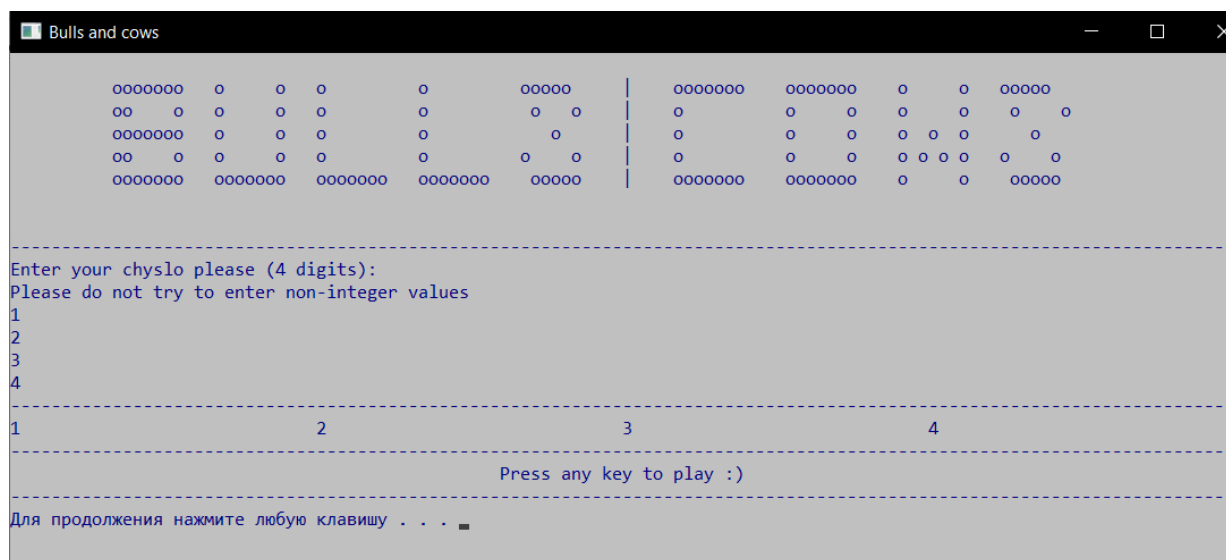


Рис.3.7. Інтерфейс пункту “Your chyslo” після введення числа.

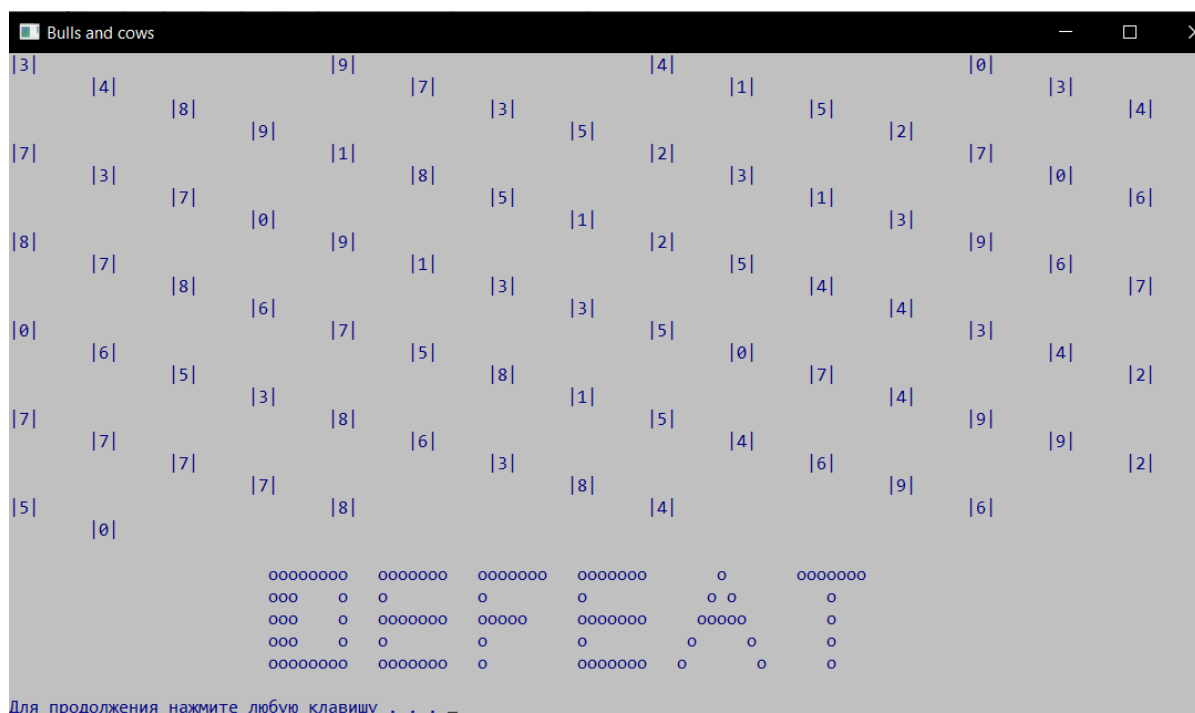


Рис.3.8. Інтерфейс пункту “Your chyslo” після гри із комп’ютером.

Інтерфейс пункту “Computer’s chyslo” вже був ідеально показаний на рисунках 2.9, 2.10, 2.11. Після гри так само натискається будь-яка клавіша і відбувається повернення до другого меню. Те саме стосується інших вкладок цього меню. У першому меню ще залишилися такі вкладки як “Delete account (forever)” та “Exit (Do not try Esc) “. Продемонструємо інтерфейс першого, якщо акаунтів не існує (Рис.3.9):



Рис.3.9. Інтерфейс “Delete account (forever)” за відсутності акаунтів.

Тепер покажемо цей же пункт за наявності хоча б 1 акаунта (Рис.3.10):



Рис.3.10. Інтерфейс “Delete account (forever)” за згоди на видалення.

Тепер продемонструємо вигляд пункту “Delete account (forever)”, якщо відмовитися від видалення та залишити усі данні усіх акаунтів незмінними (Рис.3.11.):

```

      o      o
     oo     oo
    oo      oo
   ooo     ooo
  ooo      ooo
 oooooooooooooooooo
oooo/  /\  \oooo
oooo/  /oo\ \oooo
oo/_/_/oooo\_oo
oooooooooooooooooooo
oooooooooooooooooooo
o\\oooooooooooo//o
o\\o   ooo  o//o
o\\oooooooooooo//o
o|oooooooooooo//o
o|NNNNNNNNN|o
ooooooooooooooo

1) 1
Please enter a nick:
1
-----
Are you sure you want to delete this item? Yes -> 1; No -> 0
0
-----
Have a nice day!
-----
Для продовження натисніть будь-яку клавішу . . .

```

Рис.3.11. Інтерфейс “Delete account (forever)” за відмови на видалення. Залишилось навести інтерфейс останнього пункту першого меню – пункту під назвою "Exit (Do not try Esc)"(Рис.3.12.):

```

      o      o
     oo     oo
    oo      oo
   ooo     ooo
  ooo      ooo
 oooooooooooooooooo
oooo/  /\  \oooo
oooo/  /oo\ \oooo
oo/_/_/oooo\_oo
oooooooooooooooooooo
oooooooooooooooooooo
o\\oooooooooooo//o
o\\o   ooo  o//o
o\\oooooooooooo//o
o|oooooooooooo//o
o|NNNNNNNNN|o
ooooooooooooooo

-----
Sign up
Sign in
Delete account (forever)
Exit (Do not try Esc)
-----
Process exited after 784.3 seconds with return value 0
Для продовження натисніть будь-яку клавішу . . .

```

Рис.3.12. Інтерфейс пункту "Exit (Do not try Esc)".

3.4. Результати роботи програмного продукту

Результатів завершення програмного продукту може бути надзвичайна кількість, адже рейтинг гнучкий, старі акаунти можна видаляти, а нові створювати, проте ми обмежимося чотирма випадками, у яких:

- 1) Гравець відгадав загадане комп'ютером число;
- 2) Гравець не відгадав загадане комп'ютером число;
- 3) Комп'ютер відгадав загадане гравцем число;
- 4) Комп'ютер не відгадав загадане гравцем число.

У першому випадку в останню чергу буде зчитаний файл під назвою “Victory” – і консольне вікно буде виглядати наступним чином (Рис.3.13):

```
Please, enter the first digit:
1
Please, enter the second digit:
9
Please, enter the third digit:
8
Please, enter the fourth digit:
5

-----
1                9                8                5
-----

Cows -> 4

      o      o  oo  oooooooooo  oooooooooo  oooooooooo  oooooooooo  o  o
      o      o  oo  o          o          o  o  oo  o          o  o
      o      o  oo  o          o          o  o  oo  oooooooooo  o o
      o      o  oo  o          o          o  o  oo  o          o
      o      o  oo  o          o          o  o  oo  oo  oo  o
      o      o  oo  o          o          o  o  oo  oo  oo  o
      o      o  oo  oooooooooo  o          oooooooooo  oo  oo  o

Для продолжения нажмите любую клавишу . . .
```

Рис.3.13. Інтерфейс вікна у разі відгадування гравцем числа програми.

У другому випадку в останню буде зчитаний файл під назвою “Defeat” – і консольне вікно буде виглядати так - Рис.3.14:

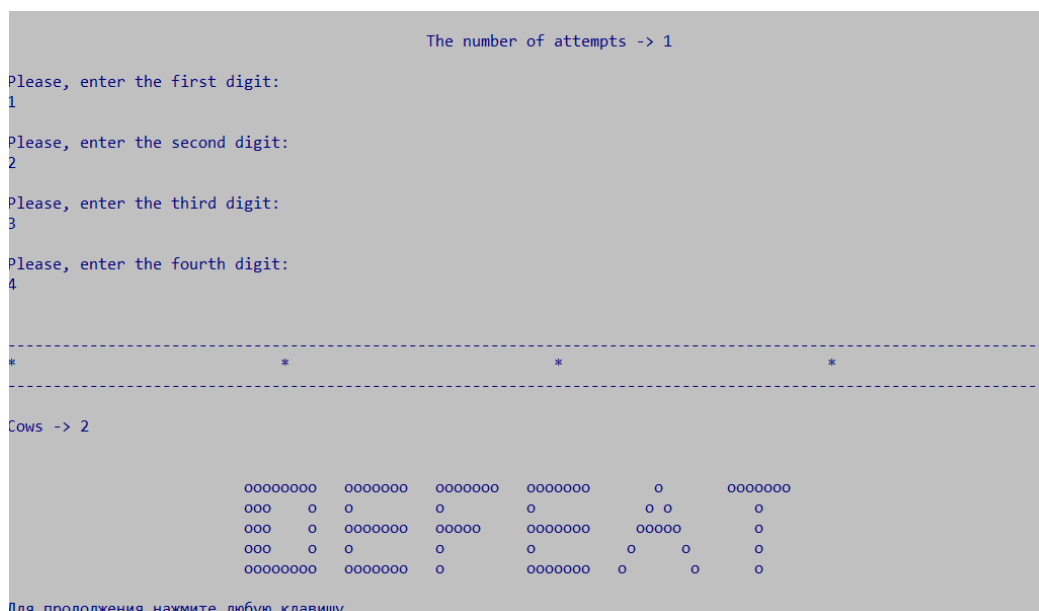


Рис.3.14. Інтерфейс вікна у разі не відгадування гравцем числа програми.

У третьому випадку в останню чергу буде зчитаний файл під назвою “Defeat” – і консольне вікно буде виглядати так - Рис.3.15:

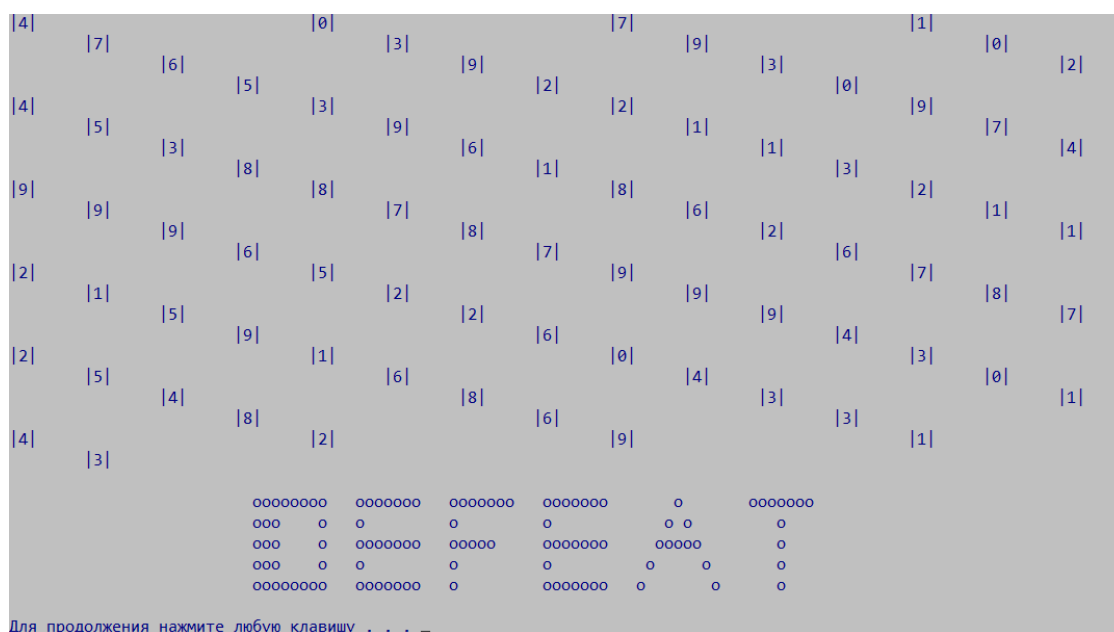


Рис.3.15. Інтерфейс вікна у разі відгадування програмою числа гравця.

У четвертому випадку, відповідно, матимемо - Рис.3.16:

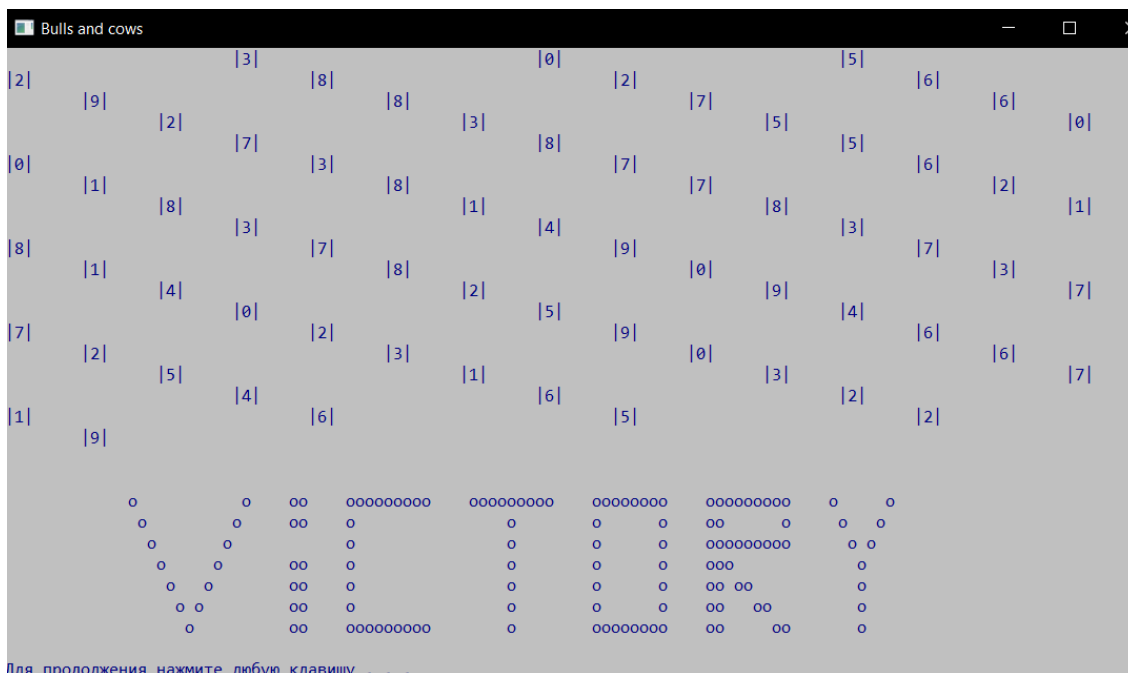


Рис.3.16. Інтерфейс вікна у разі не відгадування програмою числа гравця. Бонусом маємо наступне повідомлення в разі натискання Esc, після чого нас поверне до першого або ж другого меню – в залежності від того, в якому із них перебував гравець на момент натискання цієї клавіші - Рис.3.17:

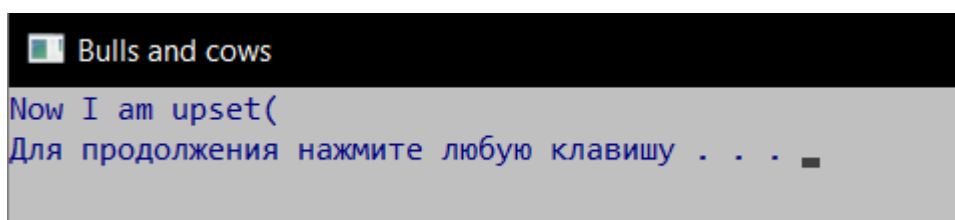


Рис.3.17. Інтерфейс вікна у разі натискання Esc.

ВИСНОВКИ

Під час розробки даного програмного продукту була створена консольна програма під назвою “Бики та корови”. Умови завдання було виконано навіть із вдосконаленням базових правил задля більшої цікавості. Зауважимо, що програма не є вимогливою до технічних характеристик комп'ютера.

За допомогою мови програмування C++ та базової бібліотеки windows.h вдалося реалізувати захоплюючу гру із доступним для розуміння інтерфейсом. Середовищем розробки було Dev-C++.

Перевагами продукту вважаю можливість керування обліковими записами та вдосконалені базові правила, що надають незвичності ігровому процесу.

Основним недоліком автор вважає відсутність можливості збереження облікових записів після завершення дії програми, а також неможливість змінити пароль до власного облікового запису.

Можливе подальше покращення інтерфейсу гри: зробити динамічні фігури биків та корів, зробити додатковий режим, де можна змагатися із друзями на час, тощо.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1) Bjarne Stroustrup Programmig: principles and practice using C++ - Second edition, 2014.
- 2) Посилання на джерело, яке було основним при розробці алгоритму - <https://www.e-olymp.com/uk/problems/4207>
- 3) Основне інтернет-джерело, яке було застосоване задля теоретичної підготовки до виконання курсового проекту - https://www.w3schools.com/cpp/cpp_intro.asp
- 4) Програмування та алгоритмічні мови. Програмування: Методичні вказівки до виконання курсового проекту для студентів галузі знань 12 «Інформаційні технології» спеціальність 122 «Комп'ютерні науки та інформаційні технології», 124 «Системний аналіз» / Уклад.: І.В.Назарчук, Г.Г.Швачко – К.НТУУ «КПІ», 2017 – 34с.
- 5) Ефективний та сучасний C++: 42 рекомендації по використанню C++11 та C++14. Скотт Майерс.
- 6) Інтернет-джерело, яке використовувалось задля розуміння принципів та правил гри: <https://sites.google.com/site/myhobi1995/gra-biki-i-korovi>
- 7) Герберт Шилдт. C++ для початківців. Крок за кроком.
- 8) Стівен Прата. Мова програмування C+. Лекції та вправи.

Додаток А

Текст програми

```
#include <string.h>
#include <windows.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <iostream>

using namespace std;

CONSOLE_SCREEN_BUFFER_INFO csbuf;
HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);

void clear_input()
{
    int a;
    while ((a = getchar()) != '\n' && a != EOF);
}

int result[5] = {0, 0, 0, 0, 0};

void bull();
void bulls_and_cows();
void Victory();
void Defeat();
void account();
void game(int);
void draw_first_menu(char* [], int, int);
void draw_second_menu(char* [], int, int);
void rules();

int main(){

    time_t t;
    srand((unsigned) time(&t));

    SetConsoleTitle("Bulls and cows");
```



```

SetConsoleTextAttribute(handle, BACKGROUND_BLUE | BACKGROUND_RED | BACKGROUND_GREEN | FOREGROUND_BLUE);

    account();
return 0;
}

void account(){

    int variant = 0;
    int q = 0, a = 0, indi;
    char r;

    string nick, sign_in, del;
    string password;
    int n;

    string Array_nick[5];
    string Array_pass[5];

    char *acc[4] = {"Sign up", "Sign in", "Delete account (forever)", "Exit (Do not try Esc)"};
    GetConsoleScreenBufferInfo(handle, &csbuf);

    while (1){

        cout << endl << endl;

        system("cls");
        draw_first_menu(acc, 4, q);

        variant = getch();

        switch (variant)
        {
            case 80:
            {
                q++;
                if (q > 3)q = 0;
                break;
            }
        }
    }
}

```

```

case 72:
{
    q--;
    if (q < 0)q = 3;
    break;
}

case 13:
{
    switch (q)
    {
        case 0:{

            system("cls");
            bull();

            if (Array_nick[0].size() == 0 | Array_nick[1].size() == 0 | Array_nick[2].size()
== 0 |
            Array_nick[3].size() == 0 | Array_nick[4].size() == 0){

                for (int i = 0; i < 5; i++){

                    if(Array_nick[i].size() == 0){
                        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
                        for (int k = 0; k < (csbuf.srWindow.Right - 20)/2; k++) cout
<< " ";

                        cout << "Enter your nickname"<<endl;
                        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";

                        cin >> Array_nick[i];

                        for (int k = 0; k < 5; k++){//проверка уникальности ников

                            if (i != k){

                                while (Array_nick[i] == Array_nick[k]){

                                    cout << "This name is taken : ( "<<endl;
                                    cin >> Array_nick[i];

                                }

                            }

                        }

                    }

                }

            }

        }

    }

}

```

```

        }
        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
        for (int k = 0; k < (csbuf.srWindow.Right - 40)/2; k++) cout
<< " ";

        cout << "Enter your password(do not enter only @)"<<endl;
        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
        cin >> Array_pass[i];

        if (Array_pass[i] == "@"){

            cout <<"I asked you not to do

this :("<<endl;

            cout << "Enter your password(do not

enter only @)"<<endl;

            cin >> Array_pass[i];

        }
        a = i;
        break;

    }

    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    cout << "Remember your data and do not give it to

anyone:"<<endl<<endl;

    cout << "Your nickname: " << Array_nick[a] << endl;
    cout << "Your password: " << Array_pass[a] << endl << endl;
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";

}

else{

    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    cout << "Sorry, but maximum can be only 5

accounts :( "<<endl<<endl;

    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";

}

    system("pause");
    break;

}

case 1:{

```

```

        bull();

        system("cls");

        if (Array_nick[0].size() == 0 & Array_nick[1].size() == 0 &
Array_nick[2].size() == 0

        & Array_nick[3].size() == 0 & Array_nick[4].size() == 0){

            for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
            for (int k = 0; k < (csbuf.srWindow.Right - 25)/2; k++) cout << " ";
            cout << "Please create an account!"<<endl<<endl;
            for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
        }

        else{ // тут будет весь остальной код

            cout << "Please log in or sign up (Enter nick):"<<endl<<endl;

            for (int i = 0; i < 5; i++){

                if(Array_nick[i].size() != 0){

                    cout << i+1 <<" " << Array_nick[i] << endl << endl;
                }
            }
            cout << "nickname -> ";
            cin >> sign_in;

            if (sign_in == Array_nick[0] ^ sign_in == Array_nick[1] ^ sign_in ==
Array_nick[2] ^ sign_in == Array_nick[3] ^ sign_in == Array_nick[4]){

                cout << "Please enter password: ";
                cin >> password;
                for (int i = 0; i < 5; i++){

                    if (sign_in == Array_nick[i]){//вхід в акаунт, чий
нік співпав із введеним

                        while (password != Array_pass[i]){

                            cout << "Please try again :
(Enter @ for exit)"<<endl;

```

```

        cin >> password;
        if (password == "@") break;
    }
    if (password == Array_pass[i]){
        for (int i = 0; i <=
            for (int k = 0; k <
                cout<< "Success!!!! Press any
                for (int i = 0; i <=
                    system("pause");
                    game(i);
            }
        }
    }
    else{
        clear_input();
        cout << "Please enter nick correctly : ( "<<endl<<endl;
    }
}
system("pause");
break;
}
case 2:{

    system("cls");
    bull();

    Array_nick[2].size() == 0
    if (Array_nick[0].size() == 0 & Array_nick[1].size() == 0 &
        & Array_nick[3].size() == 0 & Array_nick[4].size() == 0){

            for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
            for (int k = 0; k < (csbuf.srWindow.Right - 25)/2; k++) cout << " ";
            cout << "Please create an account!"<<endl<<endl;
            for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
        }
    else{

```

```

        for (int i = 0; i < 5; i++){

            if(Array_nick[i].size() != 0){

                cout << i+1 <<" " << Array_nick[i] << endl << endl;
                a = i;

            }

        }

        cout << "Please enter a nick:"<<endl;
        cin >> del;

        while (!(del == Array_nick[0] ^ del == Array_nick[1] ^ del ==
Array_nick[2] ^ del == Array_nick[3] ^ del == Array_nick[4])){//если не совпало ни одного

            clear_input();
            cout << "Please enter correctly : ( "<<endl<<endl;
            cin >> del;

        }

        for (int i = 0; i < 5; i++){

            if (Array_nick[i] == del) indi = i;

        }

        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
        for (int k = 0; k < (csbuf.srWindow.Right - 60)/2; k++) cout << " ";
        cout << "Are you sure you want to delete this item? Yes -> 1; No
-> 0" << endl;

        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
        int sure;
        for (int k = 0; k < (csbuf.srWindow.Right - 1)/2; k++) cout << " ";
        cin >> sure;
        if (sure == 1){

            result[indi] = 0;
            Array_nick[indi] = "";
            Array_pass[indi] = "";
            for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
            for (int k = 0; k < (csbuf.srWindow.Right - 10)/2; k++) cout
<< " ";

            cout << "Well done!"<<endl;

```

```

        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    }else {

        clear_input();
        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
        for (int k = 0; k < (csbuf.srWindow.Right - 16)/2; k++) cout

<< " ";

        cout <<"Have a nice day!"<<endl;
        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";

    }
}
system("pause");
break;
}

case 3:{

    return;

    break;
}

}
break;
}
case 27:
{
    system("cls");
    for (int k = 0; k < (csbuf.srWindow.Right)/2; k++) cout << " ";
    cout << "Now I am upset(\n";
    system("pause");
    break;
}
}
}
}

void game(int n){

    int variant = 0;
    int q = 0;
    int a, b, c, d;

```

```

char* var[6] = {"Your chyslo", "Computer's chyslo", "Rating", "Rules for game", "Owner", "Exit (Do not try Esc)"};
GetConsoleScreenBufferInfo(handle, &csbuf);

while (1)
{
    cout << endl << endl;

    system("cls");
    draw_second_menu(var, 6, q);

    variant = getch();

    switch (variant)
    {
        case 80:
        {
            q++;
            if (q > 5)q = 0;
            break;
        }

        case 72:
        {
            q--;
            if (q < 0)q = 5;
            break;
        }

        case 13:
        {
            switch (q)
            {
                case 0:{
                    system("cls");
                    bulls_and_cows();
                    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
                    cout << "Enter your chyslo please (4 digits):" << endl;

```



```

cout << "Please do not try to enter non-integer values "<<endl;
cin >> a;
    while (1 > a ^ a > 9){
        clear_input();
        cin >> a;
    }

    cin >> b;
    while (0 > b ^ b > 9 ^ b==a){
// clear_input();
        cin >> b;
    }

    cin >> c;
    while (0 > c ^ c > 9 ^ c==b ^ c==a){
// clear_input();
        cin >> c;
    }

    cin >> d;
    while (0 > d ^ d > 9 ^ d==a ^ d==b ^ d==c){
// clear_input();

        cin >> d;
    }
    int A[4] = {a, b, c, d};
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    for (int i = 0; i < 4; i++){

        cout << A[i];
        for (int j = 0; j < (csbuf.srWindow.Right)/4; j++) cout << "

";

    }

    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
        for (int k = 0; k < (csbuf.srWindow.Right - 23)/2; k++) cout << " ";
        cout << "Press any key to play :)" <<endl;
        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";

    system("pause");
    int e, f, g, h;
    int bo = 0;

```

```

for (int k = 0; k < 200; k++){

    e = 1 + rand()%9;

    do
    f = rand()%10;
    while (f == e);

    do
    g = rand()%10;
    while (g == e ^ g == f);

    do
    h = rand()%10;
    while (h == e ^ h == f ^ h == g);

    int B[4] = {e, f, g, h};

    cout << "|" << B[0] << "|";
    for (int j = 0; j < (csbuf.srWindow.Right)/4; j++) cout << " ";

    cout << "|" << B[1] << "|";
    for (int j = 0; j < (csbuf.srWindow.Right)/4; j++) cout << " ";

    cout << "|" << B[2] << "|";
    for (int j = 0; j < (csbuf.srWindow.Right)/4; j++) cout << " ";

    cout << "|" << B[3] << "|";
    for (int j = 0; j < (csbuf.srWindow.Right)/4; j++) cout << " ";

    if (a==B[0] & b==B[1] & c==B[2]){
        bo++;
    }
}

```

```

        if (bo == 0){

            cout <<endl<<endl;
            Victory();
            result[n]++;
            bo = 0;
            system("pause");
        }

        else{

            cout <<endl;
            Defeat();
            result[n]--;
            bo = 0;
            if (result[n] < 0) result[n] = 0;
            system("pause");
        }
        break;
    }

case 1:{

    system ("cls");
    bulls_and_cows();
    a = 1 + rand()%9;

    do
    b = rand()%10;
    while (b == a);

    do
    c = rand()%10;
    while (c == a ^ c == b);

    do
    d = rand()%10;
    while (d == a ^ d == b ^ d == c);
    // cout<<a<<b<<c<<d;
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    //
    for (int k = 0; k < (csbuf.srWindow.Right)/8; k++) cout << " ";

```

```

for (int j = 0; j < 4; j++){

    cout << "*";
    for (int i = 0; i < (csbuf.srWindow.Right)/4; i++) cout << " ";
}
for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
// for (int k = 0; k < (csbuf.srWindow.Right)/2.5; k++) cout << " ";

// cout << "please enter 4 digits:"<<endl<<endl;
int flg = 10;
int j, k, l, m;
int truemesto[4];
int sovpadenie = 0;

while (flg > 0){

    sovpadenie = 0;
    for (int k = 0; k < (csbuf.srWindow.Right - 26)/2; k++) cout << " ";
    cout << "The number of attempts -> " << flg<<endl;

    cout <<endl<< "Please, enter the first digit:"<<endl;
    cin >> j;
    while (1 > j ^ j > 9){

        cout << "Please, enter norm digit ->"<<endl;
        cin >> j;
    }
    if(j==a ^ j==b ^ j==c ^ j==d) sovpadenie++;
    if(j==a) truemesto[0] = j;

    for (int k = 0; k < (csbuf.srWindow.Right)/8; k++) cout << " ";
    printf("\nPlease, enter the second digit:\n");
    cin >> k;
    while (0 > k ^ k > 9 ^ k==j){
        printf("Please, enter norm digit -> ");
        cin >> k;
    }

    if(k==a ^ k==b ^ k==c ^ k==d) sovpadenie++;

```

```

        if(k==b) truemesto[1] = k;

for (int k = 0; k < (csbuf.srWindow.Right)/8; k++) cout << " ";
printf("\nPlease, enter the third digit:\n");
cin >> l;
while (0 > l ^ l > 9 ^ l==j ^ l==k){
    printf("Please, enter norm digit -> ");
    cin >> l;
}

if(l==a ^ l==b ^ l==c ^ l==d) sovpadenie++;
if(l==c) truemesto[2] = l;

for (int k = 0; k < (csbuf.srWindow.Right)/8; k++) cout << " ";
printf("\nPlease, enter the fourth digit:\n");
cin >> m;
while (0 > m ^ m > 9 ^ m==j ^ m==k ^ m==l){
    printf("Please, enter norm digit -> ");
    cin >> m;
}

if(m==a ^ m==b ^ m==c ^ m==d) sovpadenie++;
if(m==d) truemesto[3] = m; cout<<endl<<endl;

        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
//      for (int k = 0; k < (csbuf.srWindow.Right)/8; k++) cout <<

" ";

0; i < (csbuf.srWindow.Right)/4; i++) cout << " ";
= 0; i < (csbuf.srWindow.Right)/4; i++) cout << " ";
0; i < (csbuf.srWindow.Right)/4; i++) cout << " ";
= 0; i < (csbuf.srWindow.Right)/4; i++) cout << " ";

<< " ";

//      cout << truemesto[j];
if (a==truemesto[0]) cout << a; else cout<< "*"; for (int i =
if (b==truemesto[1]) cout << b; else cout<< "*"; for (int i
if (c==truemesto[2]) cout << c; else cout<< "*"; for (int i =
if (d==truemesto[3]) cout << d; else cout<< "*"; for (int i

for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
for (int k = 0; k < (csbuf.srWindow.Right - 9)/2; k++) cout

```

```

        cout << endl << "Cows -> " << sovpadenie << endl << endl;

        if (a==truemesto[0] & b==truemesto[1] & c==truemesto[2] &
d==truemesto[3]){

            Victory();
            result[n]++;
            break;
        }else flg--;
    }

    if (flg == 0){

        Defeat();
        result[n]--;
        if (result[n] < 0) result[n] = 0;
    }

    system("pause");
    break;
}

case 2:{
    system("cls");
    for (int k = 0; k < (csbuf.srWindow.Top)/20; k++) cout << " ";
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout << "-";
    for (int k = 0; k < (csbuf.srWindow.Right - 11)/2; k++) cout << " ";
    cout << "Your rating: ";
    cout << result[n] << endl;
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout << "-";
    system("pause");

    break;
}

case 3:
{
    system("cls");
    rules();
    system("pause");
    system("cls");
    break;
}

case 4:

```

```

{
    system("cls");
    for (int k = 0; k < (csbuf.srWindow.Top)/20; k++) cout << " ";
        for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
            for (int k = 0; k < (csbuf.srWindow.Right - 25)/2; k++) cout << " ";
    cout << "Created by: Artem Titikov"<<endl;
    for (int k = 0; k < (csbuf.srWindow.Right - 21)/2; k++) cout << " ";
    cout << "Telegram: @temyartemy"<<endl;
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    system("pause");
    system("cls");
    break;
}
case 5:
{
    return;
    break;
}
}
break;
}
case 27:
{
    system("cls");
    cout << "Now I am upset(\n";
    system("pause");
    break;
}
}
}
}
}

```

```

void rules(){

```

```

    for (int k = 0; k < (csbuf.srWindow.Top)/20; k++) cout << " ";
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";

```

```

FILE* fp;

```

```

if ((fp = fopen("rules.txt", "r")) == NULL){

```

```

        cout <<"Please install my game correctly and try again!";
        return;
    }
    int no_problem_to_find_the_file;

    while ((no_problem_to_find_the_file = fgetc(fp)) != EOF){

        printf("%c", no_problem_to_find_the_file);
    }
    cout <<endl<<endl;

    fclose(fp);
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    return;
}

void bull(){

    FILE* fp;
    if ((fp = fopen("bull.txt", "r")) == NULL){

        cout <<"Please install my game correctly and try again!";
        return;
    }
    int no_problem_to_find_the_file;

    while ((no_problem_to_find_the_file = fgetc(fp)) != EOF){

        printf("%c", no_problem_to_find_the_file);
    }
    cout <<endl<<endl;

    fclose(fp);

    return;
}

void bulls_and_cows(){

```



```

FILE* fp;
if ((fp = fopen("bulls_and_cows.txt", "r")) == NULL){

    cout <<"Please install my game correctly and try again!";
    return;
}
int no_problem_to_find_the_file;

while ((no_problem_to_find_the_file = fgetc(fp)) != EOF){

    printf("%c", no_problem_to_find_the_file);
}
cout <<endl<<endl;

fclose(fp);

return;
}

void Victory(){

FILE* fp;
if ((fp = fopen("Victory.txt", "r")) == NULL){

    cout <<"Please install my game correctly and try again!";
    return;
}
int no_problem_to_find_the_file;

while ((no_problem_to_find_the_file = fgetc(fp)) != EOF){

    printf("%c", no_problem_to_find_the_file);
}
cout <<endl<<endl;

fclose(fp);

return;
}

```

```

void Defeat(){

    FILE* fp;
    if ((fp = fopen("Defeat.txt", "r")) == NULL){

        cout <<"Please install my game correctly and try again!";
        return;
    }
    int no_problem_to_find_the_file;

    while ((no_problem_to_find_the_file = fgetc(fp)) != EOF){

        printf("%c", no_problem_to_find_the_file);
    }
    cout <<endl<<endl;

    fclose(fp);

    return;
}

void draw_first_menu(char* points[], int size, int vudileno){

    bull();
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    int a[size];

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

        a[i] = (csbuf.srWindow.Right - strlen(points[i]))/2;
    }
    for (int i = 0; i < size; i++){

        for (int j = 0; j < a[i]; j++) cout << " ";

        if (i == vudileno){

            SetConsoleTextAttribute(handle, BACKGROUND_RED | BACKGROUND_BLUE | FOREGROUND_BLUE | FOREGROUND_RED
| FOREGROUND_GREEN);

```

```

        cout << points[i] << endl << endl;

        SetConsoleTextAttribute(handle, BACKGROUND_BLUE | BACKGROUND_RED | BACKGROUND_GREEN |
FOREGROUND_BLUE);
    }
    else
        cout << points[i] << endl << endl;
}
for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
}

void draw_second_menu(char* points[], int size, int vudileno){

    bulls_and_cows();
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
    int a[size];

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

        a[i] = (csbuf.srWindow.Right) / 8;
    }
    for (int i = 0; i < size; i++){

        for (int j = 0; j < a[i]; j++) cout << " ";

        if (i == vudileno){

            SetConsoleTextAttribute(handle, BACKGROUND_RED | BACKGROUND_GREEN | BACKGROUND_BLUE |
FOREGROUND_RED | FOREGROUND_GREEN);
            cout << points[i] << endl << endl;

            SetConsoleTextAttribute(handle, BACKGROUND_BLUE | BACKGROUND_RED | BACKGROUND_GREEN |
FOREGROUND_BLUE);
        }
        else
            cout << points[i] << endl << endl;
    }
    for (int i = 0; i <= csbuf.srWindow.Right; i++) cout<<"-";
}

```