НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

**КУРСОВА РОБОТА**

з дисципліни

Програмування

на тему: «Програма тестування знань з обраної області»

Студента 1 курсу групи КА-08

Спеціальність 122 комп’ютерні науки

Овдія Ростислава Вадимовича

Керівник к.е.н. Т. І. Просянкіна-Жарова

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_

Кількість балів:\_\_\_\_\_ Оцінка: ECTS\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2021

|  |
| --- |
| **ННК „ІПСА” НТУУ „КПІ”** |
| (назва вищого закладу освіти) |

|  |  |
| --- | --- |
| Кафедра | ***математичних методів системного аналізу*** |

|  |  |
| --- | --- |
| Дисципліна | ***Програмування та алгоритмічні мови*** |

|  |  |
| --- | --- |
| Галузь знань | ***12 Інформаційні технології*** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Курс | **перший** | Група | **КА—08** | Семестр | **другий** |

**ЗАВДАННЯ**

**на курсовий проект(роботу) студента**

|  |
| --- |
|  |
| (прізвище, ім’я, по батькові) |

|  |  |
| --- | --- |
| 1. Тема проекту(роботи) |  |
|  | |
|  | |

|  |  |
| --- | --- |
| 2. Строк здачі студентом закінченого проекту(роботи) |  |

|  |  |
| --- | --- |
| 3. Вихідні дані до проекту(роботи) |  |
|  | |
|  | |
|  | |

|  |  |
| --- | --- |
| 4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) |  |
| **1. Постановка задачі.** | |
| **2. Метод розв’язку задачі** | |
| **3. Загальна блок-схема алгоритму та опис алгоритму** | |
| **4. Опис програмного продукту.** | |
| **5. Результати роботи.** | |
| **6. Висновки.** | |
| **7. Список літератури.** | |
| **Додаток А. Текст програми.** | |

|  |  |
| --- | --- |
| 5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень) |  |
| **1. Ілюстрації роботи програми.** | |
|  | |
|  | |

|  |  |
| --- | --- |
| 6. Дата видачі завдання |  |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №/п | Назва етапів курсового проекту (роботи) | Строк виконання  етапів роботи | Примітка |
| 1. | Вибір теми курсової роботи. Опрацювання відповідної літератури.  Оформлення листа Завдання. |  |  |
| 2. | Аналіз постановки задачі. |  |  |
| 3. | Вибір та дослідження методів, вибір відповідних структур даних, розробка алгоритму. Перше узгодження з керівником. |  |  |
| 4. | Проектування інтерфейсу. |  |  |
| 5. | Друге узгодження з керівником. |  |  |
| 6. | Програмна реалізація. |  |  |
| 7. | Демонстрація першого варіанту.  Трете узгодження з керівником. |  |  |
| 8. | Заключне тестування програми. |  |  |
| 9. | Аналіз результатів.  Оформлення звіту. |  |  |
| 10. | Захист та демонстрація курсової роботи. |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Студент |  |  |
|  | (підпис) |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Керівник |  |  |  |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |

|  |  |
| --- | --- |
|  |  |
| (дата) |  |

# ЗМIСТ

[ВСТУП 4](#_Toc10531609)

[РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ 6](#_Toc10531610)

[1.1. Огляд існуючих програм для тестування 6](#_Toc10531611)

[1.2. Обґрунтування вибору засобів розробки 9](#_Toc10531612)

[1.3. Уточнена постановка задачі на розробку програмного забезпечення 10](#_Toc10531613)

[РОЗДІЛ 2. ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗКУ ЗАДАЧІ ТА РОЗРОБКА АЛГОРИТМУ 11](#_Toc10531614)

[2. 1. Математичні моделі алгоритмів 11](#_Toc10531615)

[2. 2. Опис алгоритму 19](#_Toc10531616)

[2. 2. Опис алгоритму пошуку усіх можливих шляхів 21](#_Toc10531617)

[РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО   
ЗАБЕЗПЕЧЕННя 23](#_Toc10531618)

[3. 1. Опис головних структур і змінних програми 23](#_Toc10531619)

[3. 2. Опис головних функцій програми 24](#_Toc10531620)

[3. 3. Інструкція користувача 26](#_Toc10531621)

[ВИСНОВКИ 33](#_Toc10531622)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 35](#_Toc10531623)

[ДОДАТОК А Текст програми 1](#_Toc10531624)

# **ВСТУП**

У час бурхливого розвитку усіх областей науки та технологій, досить зручно вивчати нове за допомогою тестів. Тест — спосіб вивчення глибинних процесів діяльності системи, за допомогою переміщення системи в різні ситуації і відстеження доступних спостереженню змін в ній. Також тест є дуже цікавою грою, яка позитивно впливає на навчання та розвиток дітей.

Метою цієї курсової роботи є поглиблення та закріплення набутих теоретичних знань з дисципліни «Алгоритми та структури даних», а також розробка програмного забезпечення для покращення навчання дітей початкової школи, особливо в карантинних умовах.

Об’єктом дослідження є онлайн ресурси з навчальними тестами.

Завдання курсової роботи – на основі аналізу літературних джерел оцінити методи реалізації даного програмного забезпечення, скласти алгоритм роботи та представити його у вигляді програми для вивчення матеріалу та закріплення його у формі тесту.

Робота складається зі вступу, трьох розділів та додатку.

В першому розділі було описано існуючі онлайн ресурси з тестами, описано їх переваги та недоліки. Також було обґрунтовано вибір засобів розробки та сформовано уточнену постановку задачі. У другому розділі було наведено теоретичні відомості про моделі тестів, описано та розроблено алгоритм. У третьому розділі було здійснено опис головних структур, змінних та функцій програми, написано інструкцію користувачу та перевірено корекність роботи програмного забезпечення.

Програмне забезпечення було реалізоване засобами об’єктно-орієнтованого програмування мовою С++.

При виконанні роботи були використано середовище Visual Stusio, що працює на платформі Windows. Для пошуку джерел інформації було використано мережу Інтернет.

Інформаційною основою дослідження є чинні міжнародні стандарти та Державні стандарти України в галузі інформаційних технологій та наукових досліджень.

# РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

## 1.1. Огляд існуючих програм для тестування

Тест (проба) — (Від слова англ. Test — «випробування», «перевірка») або випробування — спосіб вивчення глибинних процесів діяльності системи, за допомогою переміщення системи в різні ситуації і відстеження доступних спостереженню змін в ній. Тести зараз є майже невід’ємною частиною навчання. У вигляді тестів можна, наприклад, провести співбесіду, самоперевірку, екзамени. Майже весь екзамен ЗНО – тест.

Термін «Тест» вперше ввів американський психолог Дж. Кеттел у 1890 році.

Загальна схема тестологічного дослідження припускає, крім указаних процедур, можливість охарактеризувати індивіда як сукупність пов'язаних психофізіологічних і особистісних характеристик, а також можливість виділити кожну з них окремо і досліджувати її як особливий фактор, з встановленням кількостей градацій всередині нього. Одна з найбільш розвинених областей застосування тестів пов'язана з визначенням професійної придатності до спеціальностей, для яких брало потрібно швидкість реакції. Масове застосування отримав інший тип тестологічних досліджень - визначення інтелектуального коефіцієнта IQ.

Незважаючи на те, що тести є дуже важливою частиною сучасного світу, існує не досить багато програм\сайтів для комфортної роботи із ними.

Одним з таких онлайн ресурсів є сайт naurok.ua. Naurok.ua – це безкоштовний онлайн-сервіс для проходження тестів з різних тем шкільної програми.

Перевагами Naurok.ua є:

* Простота освоєння – інтуїтивно зрозумілий користувацький інтерфейс дає можливість в один клік розпочати тест;
* Пошук – даний сервіс дає користувачу можливість знайти тест з потрібного уроку, теми, тощо;
* Тести для різного рівня знань.

Ще однією програмою для тестів є zno.osvita.ua . Це також безкоштовний онлайн-сервіс, який може допомогти в підготовці до ЗНО. Функціонал Zno.osvita не такий обширний, як у Naurok.ua оскільки сервіс розрахований на учнів 10-11 класів, які вже пройшли шкільну програму та готуються до ЗНО.

Порівнявши усі характеристики цих двох програм, у якості програми-аналога, а також програми для перевірки коректності роботи розробленого програмного забезпечення було використано онлайн-сервіс Naurok.ua.

## 

## 1.2. Обґрунтування вибору засобів розробки

Для вирішення даної задачі була обрана мова С++, яка є однією з найпоширеніших мов програмування. Вона є прямим нащадком мови С, а також містить вдосконалені властивості попередніх мов програмування. Зараз її лаконічний синтаксис, стиль та основні принципи є стандартом для розробників нових мов програмування. Окрім того, що мова С++ лаконічна, потужна та мобільна, вона підтримує об’єктно-орієнтоване програмування, шаблони, простори імен, нові бібліотеки та типи.

Мова С++ була розроблена в 1979 році Б. Страуструпом[9]. Спочатку мова називалася «С з класами», але в 1983 році її назву було змінено на С++. Ця мова програмування декілька разів була модифікована і кожного разу отримувала нові властивості.

Також саме ця мова була обрана з метою поглиблення знань і навичок в області програмування на С++.

C++ є мовою високого рівня і основою багатьох систем програмування: Borland C++, Visual C++, Borland C++ Builder, Microsoft Visual Studio тощо.

В якості інструменту розробки було використано Visual Studio. Visual Studio – серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. Програма має зручний графічний інтерфейс, який є інтуїтивно зрозумілим для користувача.

Обрані засоби розробки підходять для реалізації усіх задач, що передбачені завданням курсової роботи.

## 1.3. Уточнена постановка задачі на розробку програмного забезпечення

Розроблюване програмне забезпечення передбачає реалізацію тесту. На підставі аналізу інформаційних джерел, поставленої задачі та програм-аналогів, можна сформувати уточнену постановку задачі при розробці програмного забезпечення:

1. Дослідити інформаційні джерела;
2. Дослідити переваги та недоліки існуючих програм-аналогів;
3. Обґрунтувати вибір інструментів розробки;
4. Забезпечити реалізацію усіх функцій програми, в тому числі інтерфейсу користувача;
5. Розробити алгоритм та реалізувати відповідне програмне забезпечення мовою С++;
6. Дослідити якість роботи програми на контрольних прикладах;
7. Створити опис програми;
8. Розробити опис інтерфейсу та інструкцію користувача;
9. Визначити перспективи удосконалення розробки.

# РОЗДІЛ 2. ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗКУ ЗАДАЧІ ТА РОЗРОБКА АЛГОРИТМУ

## 2. 1. Математичні моделі алгоритмів

Для того, щоб розробити тест, потрібно уявити концепцію. Існує один найпростіший спосіб створення тесту за допомогою булевої функції та масиву.

Такий спосіб передбачає призначення кожному варіанту відповіді значення булевої функції, наприклад, 1 чи 0. Потім значення булевої функції записуються в масив і в кінці ми отримуємо результат.

## 2. 2. Опис алгоритму Тесту

Створюємо функцію Test, за допомогою функції switch-case створюємо варіанти відповідей та присвоюєму кожному значення булевої функції, після того, як користувач обере варіант відповіді, відповідне значення булевої функції запишемо в масив і так декілька разів, після останнього обраного варіанту виводимо результат на екран.

# РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 3. 1. Опис головних структур і змінних програми

Програма складається із основної функції Test, декількох допоміжних функцій, а також основної функції main. У функції Test реалізовано безпосередньо сам тест.

Для реалізації програми було створено основну функцію Test, що містить такі змінні:

* Відповідь variant типу int;
* Масив array типу int.
* Функцію flag типу bool;
* Функцію switch-case для вибору варіанту відповіді;
* Змінну-лічильник масиву і типу int;
* Змінну для обрахунку суми sum типу int;

## 3. 2. Опис головних функцій програми

Головні функції відповідають за роботу з меню, виведення на екран навчального матеріалу та сам тест.

Усі основні функції описані у таблиці 1.

Таблиця 1 – Основні функції

|  |  |
| --- | --- |
| Прототип функції | Призначення функції |
| int main() | Меню та виклик інших основних функцій, головна в програмі. |
| int Test() | Показ на екрані тесту з можливістю тестування та виведенням результату. |
| int Open[1,2,3,4]() | Функції, що відповідають за відкриття та копіювання на екран текстових файлів відповідно до обраної теми. |

Для перевірки на можливість вибору варіанту використовується допоміжна функція get\_selection(int count) типу int, її опис наведено в таблиці 2.

Таблиця 2 – Функція get\_selection(int count) для перевірки коректності вводу

|  |  |  |
| --- | --- | --- |
| Прототип функції | Призначення функції | Параметри |
| get\_selection(int count) | Для перевірки коректності вводу | count – максимальний номер вводу (кількість пунктів меню). |

Перевагою розробленого програмного забезпечення є те, що завдяки тому, що код розділений на окремі частини (функції), що послідовно виконують дії та пов’язані між собою обміном даних.

## 3. 3. Інструкція користувача

Щоб почати роботу із програмним забезпеченням, треба скопіювати на комп’ютер файл, який називається Test.exe, а також завантажити на диск D папку з назвою Kursach Files. Програма призначена для роботи на платформі Windows.

Після запуску додатку відкривається консольне вікно програми, де користувачу пропонується вибрати один з пунктів меню:

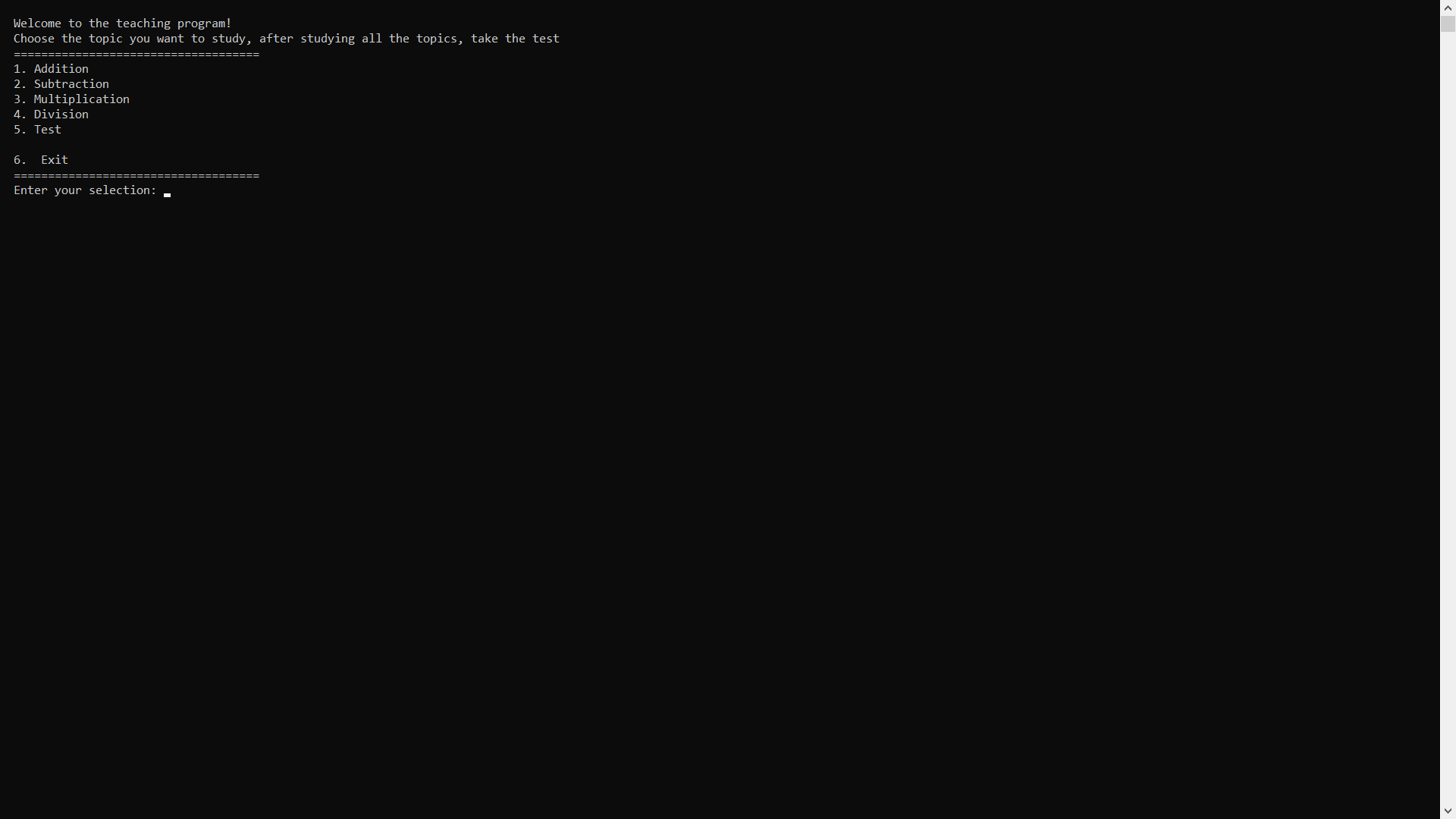


Рис.1. Початкове вікно

Програма була розроблена англійською мовою з метою використання додатку англомовними користувачами.

Користувач може обрати один з 4 навчальних розділів(1,2,3,4), тест(5) або завершити роботу програми(6).

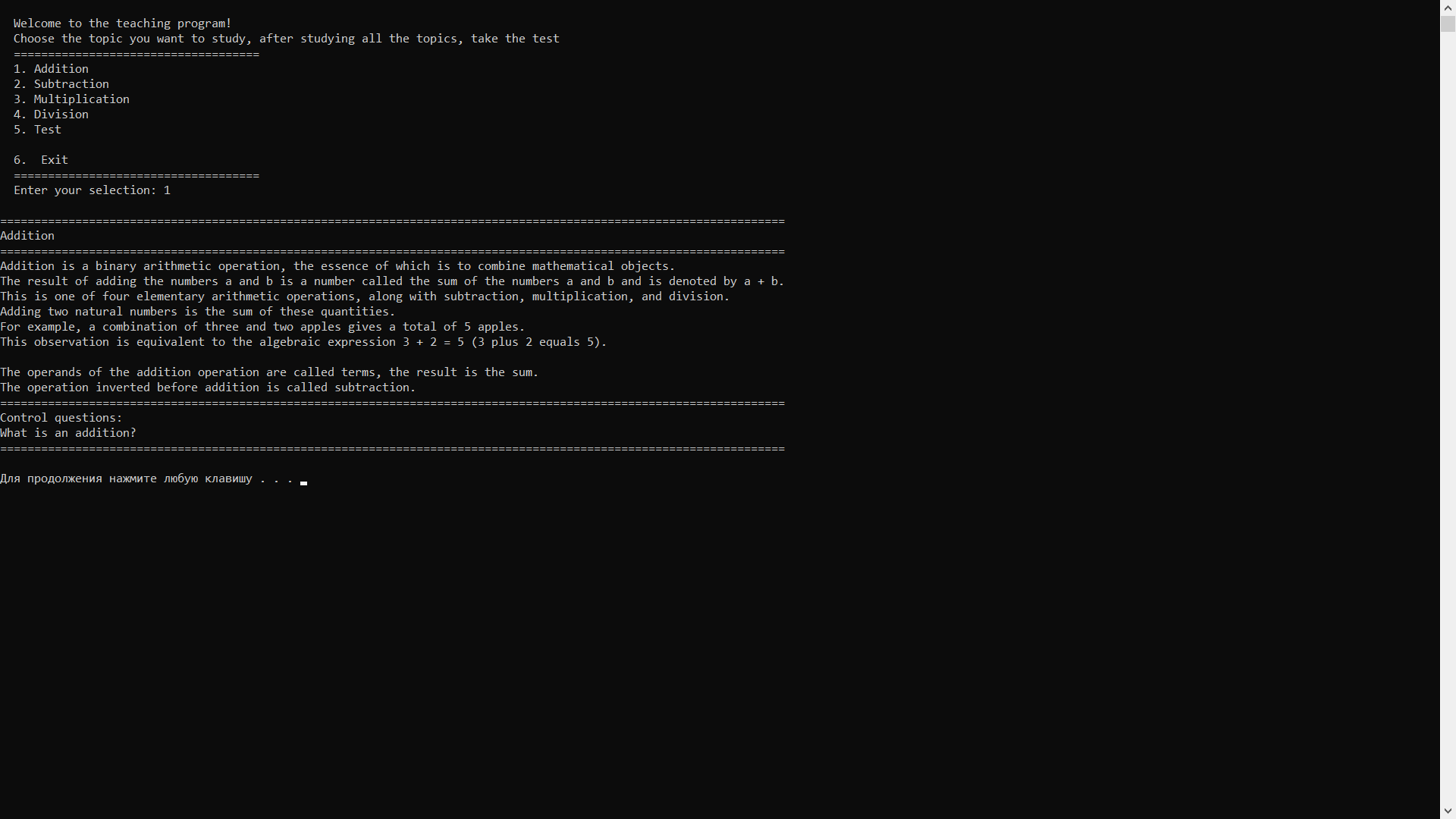


Рис.2. Обрано один з навчальних пунктів

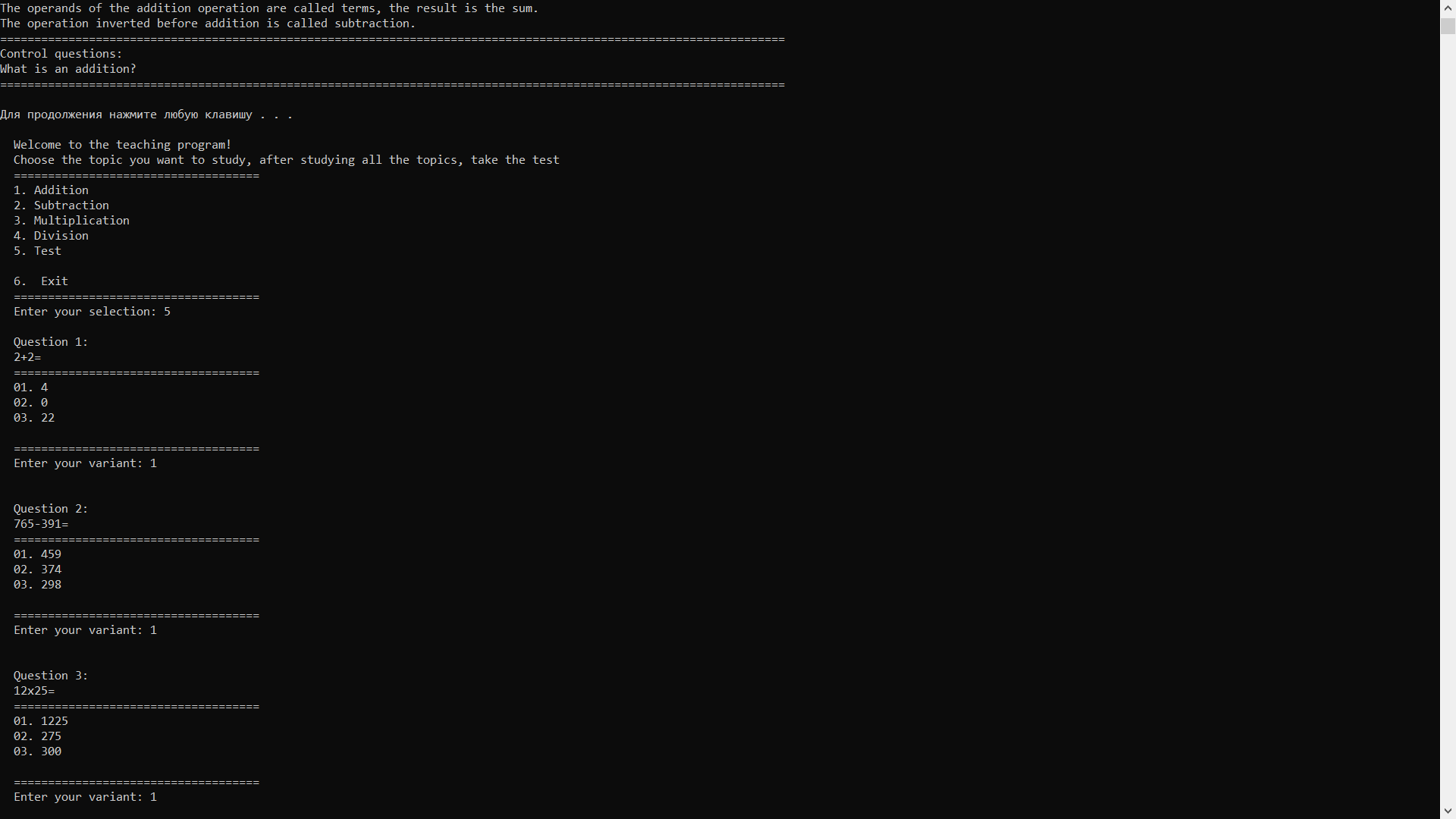


Рис.3. Обрано тест

Після вибору тесту, перед користувачем з’являються нове питання після відповіді на попереднє, усього питань 10. Після відповіді на всі, користувач отримує свій результат:

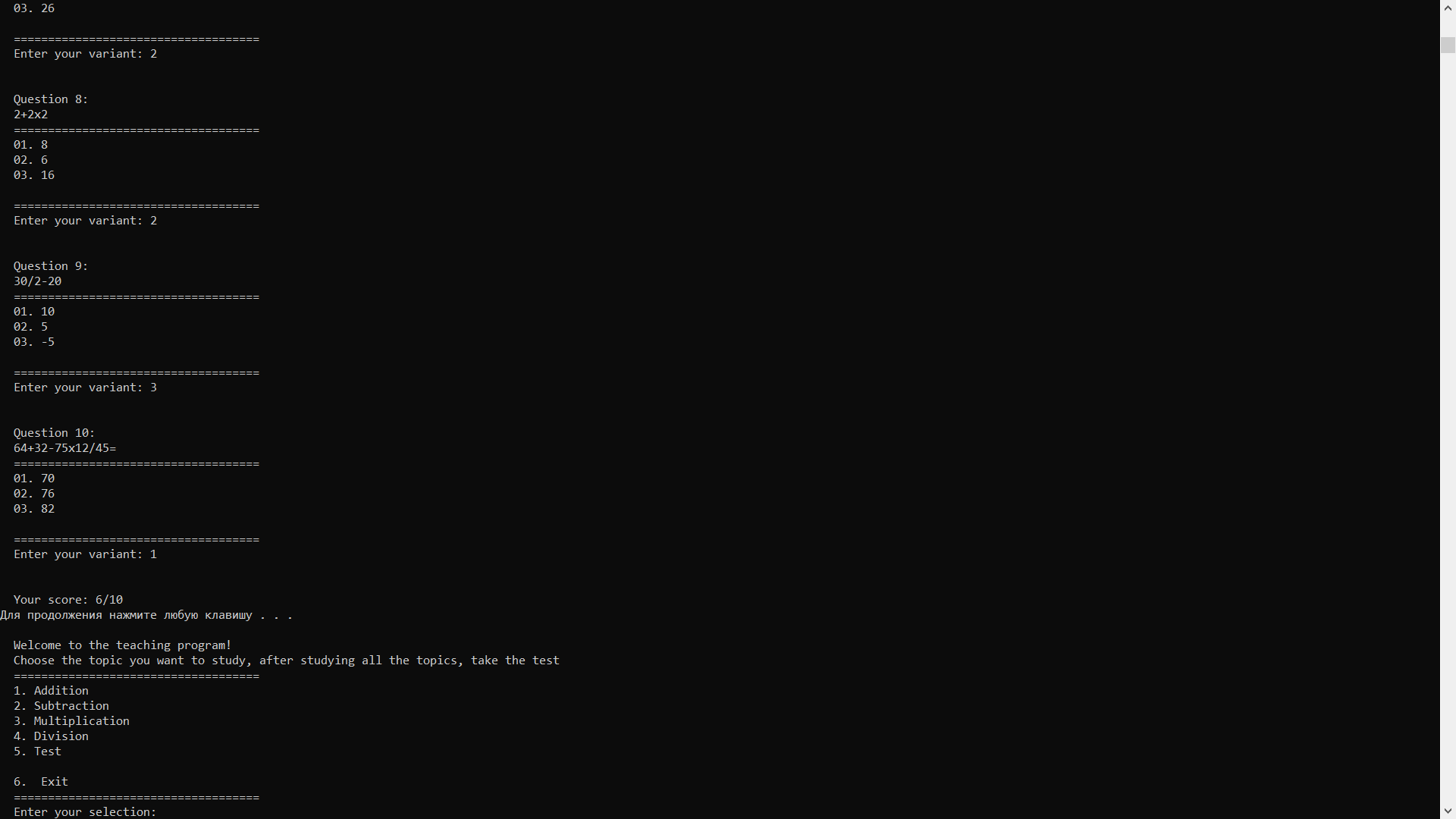


Рис.4. Кінець тесту

Після проходження тесту та отримання результату програма не завершує роботу аби користувач мав змогу повторити матеріал та пройти тест ще раз, якщо користувач вдоволений результатом він може вимкнути програму обравши варіант 6.

Розроблене програмне забезпечення має зручне меню та приємний і інтуїтивно зрозумілий користувацький інтерфейс. Отже, дана програма задовольняє усім технічним вимогам поставленого завдання.

# 3. 4. Перевірка правильності роботи програми

Перевірити програму на працездатність дуже просто, оскільки при виборі пунків з навчанням ми отримуємо заголовок:

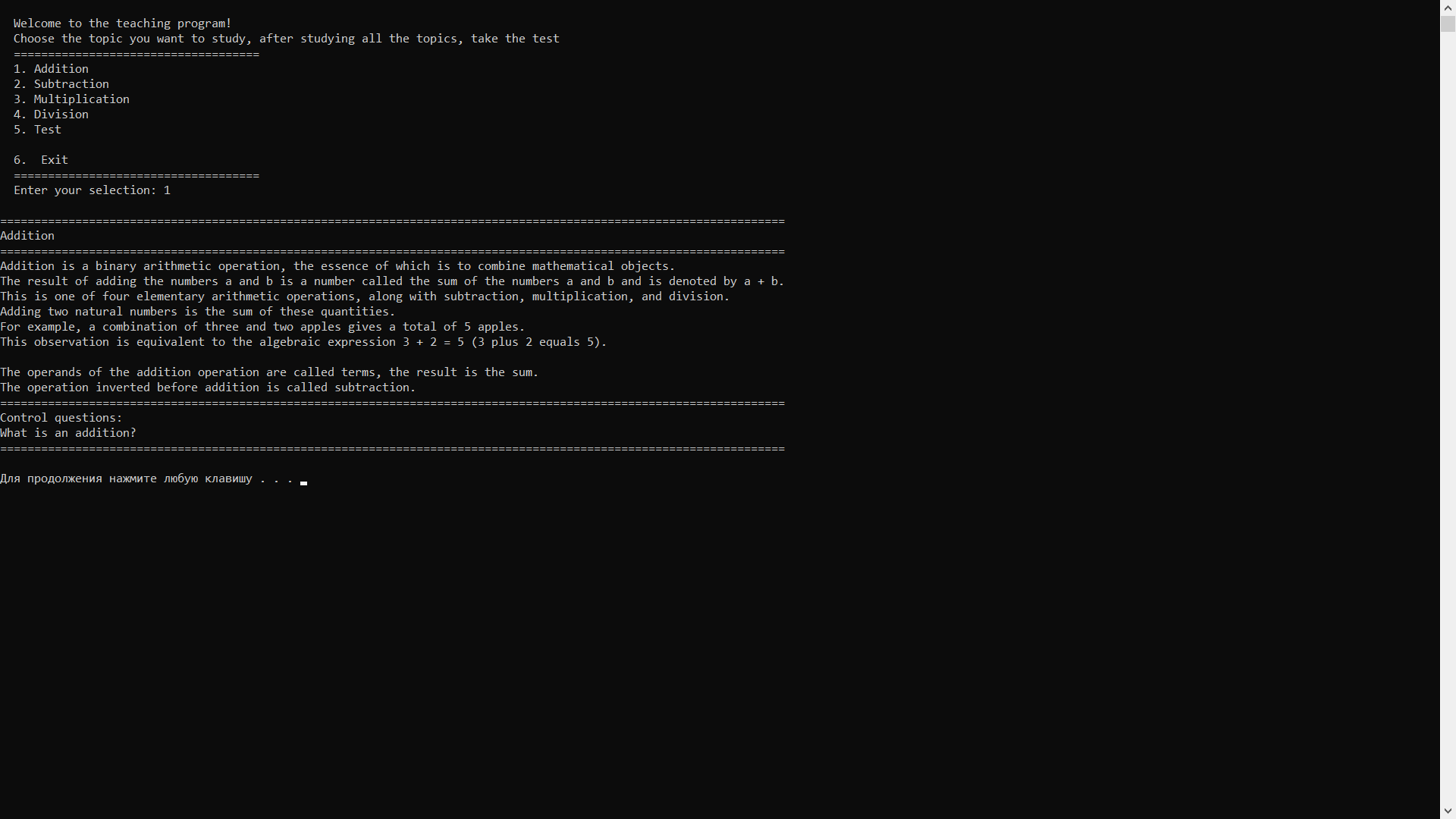


Рис.5. Обраний варіант Addition програма вивела інформацію про Addition

Тест перевірено за допомогою звичайного калькулятора.

# ВИСНОВКИ

В результаті виконання курсової роботи було розв’язано актуальну проблему – написано консольний додаток, який дозволяє спростити та зробити цікавим навчання дітей в карантинних умовах.

У першому розділі було розглянуто інформаційні джерела та наведені основні відомості по даній тематиці. Також було розглянуто алгоритми-аналоги, проаналізовано їх мінуси та плюси, обрано найоптимальніші шляхи розв’язку окремих частин задачі та обґрунтовано вибір інструментів розробки. Також було сформульовано уточнену постановку даної задачі.

У другому розділі було досліджено та проаналізовано існуючі методи виконання задач. Також було розроблено алгоритм.

У третьому розділі було виконано опис розробленої програми, в тому числі основних змінних, що використовуються, структур та функцій. Також було перевірено коректність роботи даної програми, створено інструкцію користувача.

Програма коректно реалізує усі функції, що прописані у завданні. Правильність роботи програми була перевірена. В якості програми-аналогу було використано онлайн-сервіс Naurok.ua. Програма має зручний та зрозумілий інтерфейс.

Отже, програма задовольняє вимогам поставленого завдання. В подальшому вона може бути вдосконалена додаванням нової інформації або ж тестів, вдосконаленням інтерфейсу, наведення короткого текстового опису використаних алгоритмів тощо.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Оператори switch-case - https://ravesli.com/urok-65-operator-switch/
2. Текстові файли в с++. http://cppstudio.com/post/446/
3. Посібник з лабораторними роботами з дисципліни алгоритмізація та програмування для першого курсу, другий семестр.
4. https://uk.wikipedia.org/wiki/Тест
5. https://uk.wikipedia.org/wiki/Microsoft\_Visual\_Studio
6. https://dic.academic.ru/dic.nsf/enc\_philosophy/3468/ТЕСТ#:~:text=Термин%20«Т.»%20впервые%20ввёл,Кеттел%20(1890).
7. Інтернет-ресурс Naurok.ua. https://naurok.ua
8. Офіційний сайт Zno.osvita.ua. https://zno.osvita.ua
9. 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. 32с.

# 

**ДОДАТКИ**

# ДОДАТОК А

#include <iostream>

#include <fstream>

#include <string>

#include <windows.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

using namespace std;

int get\_selection(int count) {

int selection;

string s; // строка для считывания введённых данных

getline(cin, s); // считываем строку

// пока ввод некорректен, сообщаем об этом и просим повторить его

while (sscanf\_s(s.c\_str(), "%d", &selection) != 1 || selection < 1 || selection > count) {

cout << "Incorrect input. Try again: "; // выводим сообщение об ошибке

getline(cin, s); // считываем строку повторно

}

return selection;

}

int Test()

{

int array[10];

bool flag;

int variant;

//первый вопрос

cout << " Question 1:\n";;

cout << " 2+2=\n";

cout << " ====================================\n";

cout << " 01. 4\n";

cout << " 02. 0\n";

cout << " 03. 22\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 1;

cout << "\n";

break;

case 2:

flag = 0;

cout << "\n";

break;

case 3:

flag = 0;

cout << "\n";

break;

}

if (flag)

array[0] = 1;

else

array[0] = 0;

//второй вопрос

cout << " Question 2:\n";

cout << " 765-391=\n";

cout << " ====================================\n";

cout << " 01. 459\n";

cout << " 02. 374\n";

cout << " 03. 298\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 0;

cout << "\n";

break;

case 2:

flag = 1;

cout << "\n";

break;

case 3:

flag = 0;

cout << "\n";

break;

}

if (flag)

array[1] = 1;

else

array[1] = 0;

//третий вопрос

cout << " Question 3:\n";

cout << " 12x25=\n";

cout << " ====================================\n";

cout << " 01. 1225\n";

cout << " 02. 275\n";

cout << " 03. 300\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 0;

cout << "\n";

break;

case 2:

flag = 0;

cout << "\n";

break;

case 3:

flag = 1;

cout << "\n";

break;

}

if (flag)

array[2] = 1;

else

array[2] = 0;

//четвертый вопрос

cout << " Question 4:\n";

cout << " 846/18=\n";

cout << " ====================================\n";

cout << " 01. 46\n";

cout << " 02. 47\n";

cout << " 03. 48\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 0;

cout << "\n";

break;

case 2:

flag = 1;

cout << "\n";

break;

case 3:

flag = 0;

cout << "\n";

break;

}

if (flag)

array[3] = 1;

else

array[3] = 0;

//пятый вопрос

cout << " Question 5:\n";

cout << " 6x6/9=\n";

cout << " ====================================\n";

cout << " 01. 6\n";

cout << " 02. 4\n";

cout << " 03. 3\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 0;

cout << "\n";

break;

case 2:

flag = 1;

cout << "\n";

break;

case 3:

flag = 0;

cout << "\n";

break;

}

if (flag)

array[4] = 1;

else

array[4] = 0;

//шестой вопрос

cout << " Question 6:\n";

cout << " 45+45/15=\n";

cout << " ====================================\n";

cout << " 01. 4\n";

cout << " 02. 15\n";

cout << " 03. 48\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 0;

cout << "\n";

break;

case 2:

flag = 0;

cout << "\n";

break;

case 3:

flag = 1;

cout << "\n";

break;

}

if (flag)

array[5] = 1;

else

array[5] = 0;

//седьмой вопрос

cout << " Question 7:\n";

cout << " (13+13)x0\n";

cout << " ====================================\n";

cout << " 01. 0\n";

cout << " 02. 13\n";

cout << " 03. 26\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 1;

cout << "\n";

break;

case 2:

flag = 0;

cout << "\n";

break;

case 3:

flag = 0;

cout << "\n";

break;

}

if (flag)

array[6] = 1;

else

array[6] = 0;

//восьмой вопрос

cout << " Question 8:\n";

cout << " 2+2x2\n";

cout << " ====================================\n";

cout << " 01. 8\n";

cout << " 02. 6\n";

cout << " 03. 16\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 0;

cout << "\n";

break;

case 2:

flag = 1;

cout << "\n";

break;

case 3:

flag = 0;

cout << "\n";

break;

}

if (flag)

array[7] = 1;

else

array[7] = 0;

//девятый вопрос

cout << " Question 9:\n";

cout << " 30/2-20\n";

cout << " ====================================\n";

cout << " 01. 10\n";

cout << " 02. 5\n";

cout << " 03. -5\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 0;

cout << "\n";

break;

case 2:

flag = 0;

cout << "\n";

break;

case 3:

flag = 1;

cout << "\n";

break;

}

if (flag)

array[8] = 1;

else

array[8] = 0;

//десятый вопрос

cout << " Question 10:\n";

cout << " 64+32-75x12/45=\n";

cout << " ====================================\n";

cout << " 01. 70\n";

cout << " 02. 76\n";

cout << " 03. 82\n";

cout << "\n";

cout << " ====================================\n";

cout << " Enter your variant: ";

variant = get\_selection(3);

cout << endl;

switch (variant)

{

case 1:

flag = 0;

cout << "\n";

break;

case 2:

flag = 1;

cout << "\n";

break;

case 3:

flag = 0;

cout << "\n";

break;

}

if (flag)

array[9] = 1;

else

array[9] = 0;

int sum = 0;

for (int i = 0; i < 10; i++)

{

sum = sum + array[i];

}

cout << " Your score: ";

cout << sum << "/10";

return 0;

}

int Open1()

{

ifstream file;

file.open("D:\\Kursach Files\\File1.txt");

char ch;

cout << "\n";

while (file.get(ch))

{

cout << ch;

}

cout << "\n";

return 0;

}

int Open2()

{

ifstream file;

file.open("D:\\Kursach Files\\File2.txt");

cout << "\n";

char ch;

while (file.get(ch))

{

cout << ch;

}

cout << "\n";

return 0;

}

int Open3()

{

ifstream file;

file.open("D:\\Kursach Files\\File3.txt");

cout << "\n";

char ch;

while (file.get(ch))

{

cout << ch;

}

cout << "\n";

return 0;

}

int Open4()

{

ifstream file;

file.open("D:\\Kursach Files\\File4.txt");

cout << "\n";

char ch;

while (file.get(ch))

{

cout << ch;

}

cout << "\n";

return 0;

}

int main()

{

int selection;

do

{

cout << "\n";

cout << " Welcome to the teaching program!\n";

cout << " Choose the topic you want to study, after studying all the topics, take the test\n";

cout << " ====================================\n";

cout << " 1. Addition\n";

cout << " 2. Subtraction\n";

cout << " 3. Multiplication\n";

cout << " 4. Division\n";

cout << " 5. Test\n";

cout << "\n";

cout << " 6. Exit\n";

cout << " ====================================\n";

cout << " Enter your selection: ";

selection = get\_selection(6);

switch (selection)

{

case 1:

Open1();

cout << "\n";

break;

case 2:

Open2();

cout << "\n";

break;

case 3:

Open3();

cout << "\n";

break;

case 4:

Open4();

cout << "\n";

break;

case 5:

cout << "\n";

Test();

cout << "\n";

break;

}

if (selection != 6)

system("pause");

} while (selection != 6);

return 0;

}

Посилання на пакет інструментів: https://github.com/Bereex/Kursova-Ovdiy