

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

ПОЗИЦІЙНІ СИСТЕМИ ЧИСЛЕННЯ

Пояснювальна записка до розрахунково-графічної роботи

з дисципліни «Алгоритмізація і програмування»

XAI.301.312.2 РГР

Виконав студент гр. 312

(№ групи)

Ксенія ВЕЛІКОДАНОВА

(Підпис, дата)

(П.І.Б.)

Перевірив к.т.н., доцент

(Науковий ступінь, вчене звання)

О. В. Гавриленко

(Підпис, дата)

(П.І.Б.)

ЗАВДАННЯ

Дослідити шляхом власних обчислень, розробити і реалізувати алгоритми роботи з числами в різних позиційних системах числення:

- 1) Перетворити десяткові числа 133 та 2128 в двійкову систему числення, описати покроково процес перетворень. Виконати перевірку, виконавши зворотне перетворення в десяткову систему.
- 2) Перетворити десяткові числа 133 та 2128 в шістнадцяткову систему числення, описати покроково процес перетворень. Виконати перевірку шляхом зворотного перетворення в десяткову і двійкову систему.
- 3) Розробити діаграму активності алгоритму перетворення числа з десяткової системи числення в 17-річну. Реалізувати алгоритм у вигляді строкової функції `DecTo_17_(D)` з вхідним цілочисельним параметром на мові C++.
- 4) Для двох чисел 133 та 2128 провести операцію додавання у двійковій системі числення. Виконати перевірку шляхом перетворення результатів в десяткову систему.
- 5) Зробити висновки.

ЗМІСТ

Завдання	2
Зміст	3
Вступ	4
1 Перетворення чисел в двійкову систему числення	4
1.1 Перетворення трирозрядного десяткового числа	5
1.2 Перетворення чотирирозрядного десяткового числа	5
1.3 Перевірка результатів	6
2 Перетворення чисел в шістнадцяткову систему числення	7
2.1 Перетворення трирозрядного десяткового числа	7
2.2 Перетворення чотирирозрядного десяткового числа	7
2.3 Перевірка результатів	7
3 Перетворення чисел в 17-річну систему числення	8
4 Двійкова арифметика	9
Висновки	10
Додаток А	11
Додаток Б	12

ВСТУП

Системи числення — це спосіб подання чисел за допомогою певного набору символів і правил. У повсякденному житті ми звикли до десяткової системи числення, в якій використовуються цифри від 0 до 9. Людині вона інтуїтивно зрозуміла: ми легко виконуємо арифметичні дії, читаємо та записуємо числа, швидко оцінюємо їх розмір. Проте в комп'ютерній техніці головну роль відіграє двійкова система, де для запису чисел застосовуються лише дві цифри — 0 і 1.

Двійкова система має велике значення для електроніки, оскільки її можна реалізувати фізично — через пристрої, що мають лише два стани (наприклад, увімкнено/вимкнено). В основі роботи всіх цифрових пристроїв лежить саме ця система, оскільки вона дозволяє зручно кодувати та обробляти інформацію за допомогою логічних елементів. Попри це, людині важко сприймати довгі послідовності нулів і одиниць, тому для зручності часто використовуються й інші системи — шістнадцяткова, восьмерична, а також нестандартні, наприклад, сімнадцяткова.

Перетворення чисел між різними системами є важливим практичним навиком для програмістів і технічних спеціалістів. У цій роботі було розглянуто особливості позиційних систем числення, виконано переведення чисел з десяткової у двійкову, шістнадцяткову та 17-кову системи, а також реалізовано алгоритм у вигляді програмної функції. Додатково було закріплено навички виконання арифметичних дій у двійковій формі, що є основою для розуміння роботи комп'ютерної логіки.

1 ПЕРЕТВОРЕННЯ ЧИСЕЛ В ДВІЙКОВУ СИСТЕМУ ЧИСЛЕННЯ

1.1 Перетворення трирозрядного десяткового числа

Покроковий опис перетворення наведено у табл.1.1.

Таблиця 1.1 – Перетворення десяткового числа у двійкове

X	X/2	X%2
133	66	1
66	33	0
33	16	1
16	8	0
8	4	0
4	2	0
2	1	0
1	0	1
Результат		$133_{10} = 10000101_2$

1.2 Перетворення чотирирозрядного десяткового числа

Покроковий опис перетворення наведено у табл.1.2.

Таблиця 1.2 – Перетворення десяткового числа у двійкове

X	X/2	X%2
2128	1064	0
1064	532	0
532	266	0
266	133	0
133	66	1
66	33	0
33	16	1
16	8	0
8	4	0
4	2	0
2	1	0

<i>1</i>	<i>0</i>	<i>1</i>
Результат		$2128_{10} = 100001010000_2$

1.3 Перевірка результатів

Перетворення в десяткову 2х чисел: $10000101_2 = (1 \times 128) + (0 \times 64) + (0 \times 32) + (0 \times 16) + (0 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$

$$= 128 + 0 + 0 + 0 + 0 + 4 + 0 + 1$$

$$= 133_{10}$$

$$100001010000_2 = (1 \times 2048) + (1 \times 64) + (1 \times 16)$$

$$= 2048 + 64 + 16$$

$$= 2128$$

2 ПЕРЕТВОРЕННЯ ЧИСЕЛ В ШІСТНАДЦЯТКОВУ СИСТЕМУ ЧИСЛЕННЯ

2.1 Перетворення трирозрядного десяткового числа

Покроковий опис перетворення наведено у табл.2.1.

Таблиця 2.1 – Перетворення десяткового числа у шістнадцяткове

X	X /16	X %16
133	8	5
8	0	8
Результат		$133_{10} = 85_{16}$

2.2 Перетворення чотирирозрядного десяткового числа

X	X /16	X %16
2128	133	0
133	8	5
8	0	8
Результат		$2128_{10} = 850_{16}$

2.3 Перевірка результатів

Перетворення в десяткову 2х чисел: $85_{16} = 8 \cdot 16^1 + 5 \cdot 16^0 = 128 + 5 = 133_{10}$,
 $8 \cdot 16^2 + 5 \cdot 16^1 + 0 \cdot 16^0 = 2048 + 80 + 0 = 2128$

в двійкову 2х чисел:

8 – 1000 , 5 – 0101 $\rightarrow 85_{16} = 10000101_2$

8 – 1000 , 5 – 0101 , 0 – 0000 $\rightarrow 850_{16} = 100001010000_2$

(співпадає з пп.2.1,2.2)

3 ПЕРЕТВОРЕННЯ ЧИСЕЛ В 17-РІЧНУ СИСТЕМУ ЧИСЛЕННЯ

Діаграму активності представлено на рис. 1 в дод.А
код на C++ в дод.Б

ВИСНОВКИ

Було вивчено способи переведення чисел із десяткової системи у позиційні системи з різними основами, зокрема з основою 17. Закріплено на практиці побудову алгоритму та реалізацію функції перетворення у систему числення з нестандартним алфавітом. Відпрацьовано в коді програми обробку залишків та формування рядкового результату. Отримано навички бінарного додавання та перевірки коректності результатів.

ДОДАТОК А

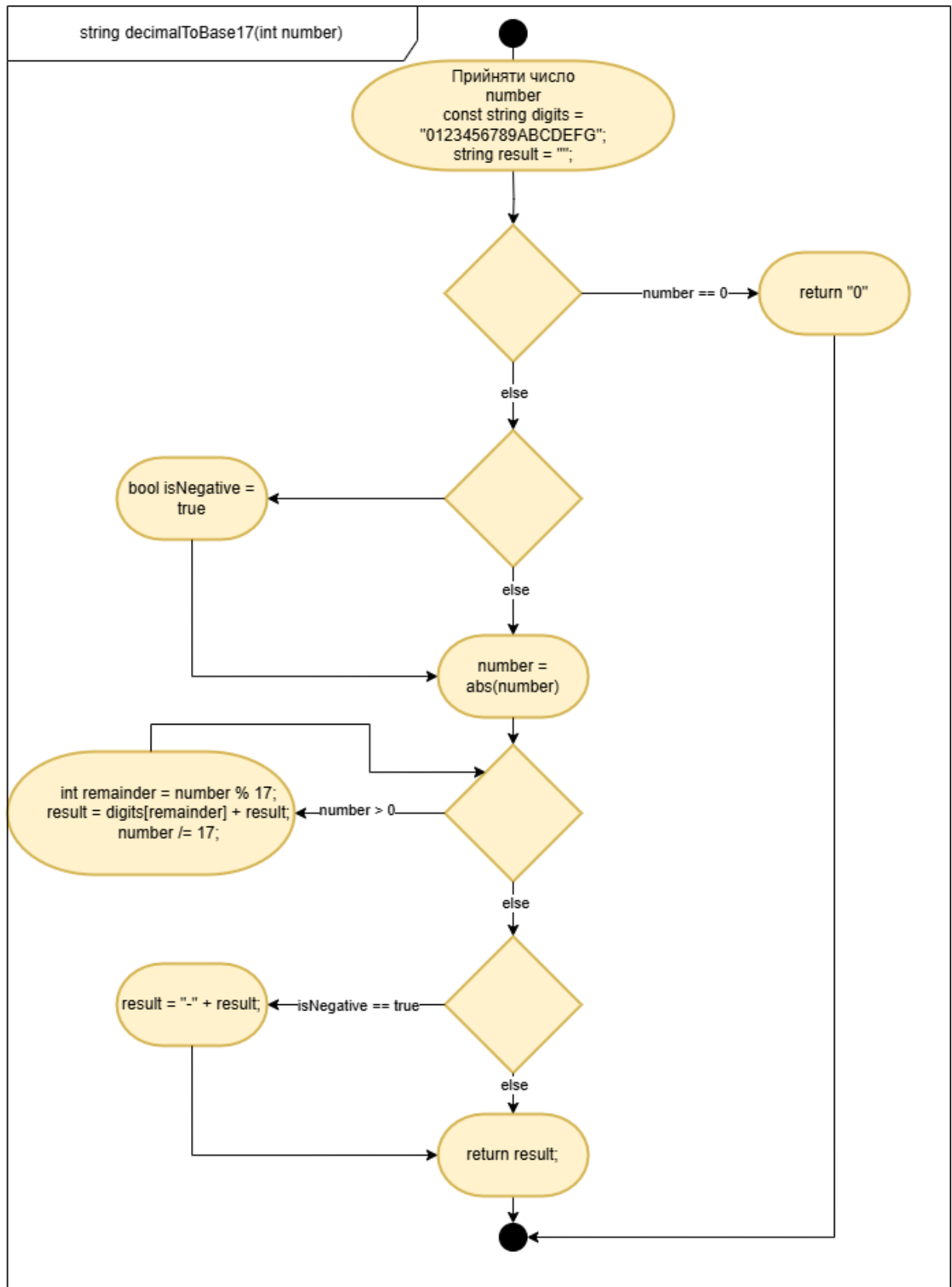


Рисунок 1 Алгоритм переводу числа з десятикової системи в 17-річну систему

ДОДАТОК Б

```
#include <iostream>
#include <string>
using namespace std;

string decimalToBase17(int number) {
    if (number == 0) return "0";

    const string digits = "0123456789ABCDEFGH";
    string result = "";

    bool isNegative = number < 0;
    number = abs(number);

    while (number > 0) {
        int remainder = number % 17;
        result = digits[remainder] + result;
        number /= 17;
    }

    if (isNegative) {
        result = "-" + result;
    }

    return result;
}

// Example usage
int main() {
    int num;
    cout << "Enter a decimal number: ";
    cin >> num;
    cout << "Base-17: " << decimalToBase17(num) << endl;
    return 0;
}
```