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»

ХАІ.301.173.310.02 РГР

Виконав студент гр. 310_____
(№ групи)
20.05.2024 Софія Полякова_____
(Підпис, дата) (П.І.Б.)

Перевірив к.т.н., доцент _____
(Науковий ступінь, вчене звання)
_____ О. В. Гавриленко
(Підпис, дата) (П.І.Б.)

2024

ЗАВДАННЯ

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

1) Перетворити десяткові числа ____125_та_2000__ в двійкову систему числення, описати покроково процес перетворень. Виконати перевірку, виконавши зворотне перетворення в десяткову систему.
2) Перетворити десяткові числа ____125_та 2000__ 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 __7__-річну. *Реалізувати алгоритм у вигляді строкової функції DecTo_N_ (D) з вхідним цілочисельним параметром на мові С ++.
4) Для двох чисел ____125_та 2000__ провести операцію ___додавання_ у двійковій системі числення. Виконати перевірку шляхом перетворення результатів в десяткову систему.
5) Зробити висновки.

ЗМІСТ

ВСТУП

Зазвичай ми використовуємо десяткову систему числення, де 10 одиниць становлять десятку, 10 десятків — сотню і так далі. Проте існують й інші системи числення. Система числення — це набір правил для запису чисел за допомогою цифр. Є позиційні та непозиційні системи числення. У непозиційних системах значення знака не залежить від його місця в числі. Наприклад, у римській системі I означає 1, V — 5, X — 10 тощо. В одиничній системі число 7 записується як сім одиниць: $(7)_{10} = (1111111)_1$. Недоліки непозиційних систем — громіздкість і складність обчислень.

Позиційні системи числення зручніші для запису чисел і обчислень. У таких системах значення цифри залежить від її позиції в числі. Кожна позиційна система має обмежену кількість знаків, що визначає її основу. Наприклад, у десятковій системі, яка базується на числі 10, використовуються цифри від 0 до 9. Числа можна представити як степені десятки:

$(237)_{10} = 2 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0$

$(77{,}3)_{10} = 7 \cdot 10^1 + 7 \cdot 10^0 + 3 \cdot 10^{-1}$

Щоб краще освоїти двійкову систему числення, потрібно навчитися виконувати арифметичні операції з двійковими числами. Усі позиційні системи числення подібні: арифметичні операції в них виконуються за одними і тими ж правилами. Зокрема:

1) Діють ті самі закони арифметики:
   - Комутативний
   - Асоціативний
   - Дистрибутивний
2) Використовуються ті самі методи додавання, віднімання, множення та ділення стовпчиком
3) Правила виконання арифметичних операцій базуються на таблицях додавання і множення.

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 125 у двійкове

X	**X/2**	**X%2**
125	*62*	*1*
62	*31*	*0*
31	*15*	*1*
15	*7*	*1*
7	*3*	*1*
3	*1*	*1*
1	*0*	*1*
	Результат:	$125_{10} = 1111101_2$

1.2 Перетворення чотирирозрядного десяткового числа

Покроковий опис перетворення наведено *у табл.1.2.*

Таблиця 1.2 – Перетворення десяткового числа 2000 у двійкове

X	**X/2**	**X%2**
2000	*1000*	*0*
1000	*500*	*0*
500	*250*	*0*
250	*125*	*0*
125	*62*	*1*
62	*31*	*0*
31	*15*	*1*
15	*7*	*1*
7	*3*	*1*
3	*1*	*1*
1	*0*	*1*
	Результат:	$2000_{10} = 11111010000_2$

1.3 Перевірка результатів

Перевірка трирозрядного десяткового числа

Число 1111101:

$1*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 64 + 32 + 16 + 8 + 4 + 0 + 1 = 125$

Перевірка чотирирозрядного десяткового числа

Число 11111010000:

$1*2^{10} + 1*2^9 + 1*2^8 + 1*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 0*2^1 + 0*2^0 =$

$= 1024 + 512 + 256 + 128 + 64 + 0 + 16 + 0 + 0 + 0 + 0 = 2000$

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 125 у шістнадцяткове

X	**X /16**	**X %16**
125	*7*	*13 (D)*
7	*0*	*7*
	Результат:	$125_{10} = 7D_{16}$

2.2 Перетворення чотирирозрядного десяткового числа

Покроковий опис перетворення наведено *у табл.2.2.*

Таблиця 2.1 – Перетворення десяткового числа 2000 у шістнадцяткове

X	**X /16**	**X %16**
2000	*125*	*0*
125	*7*	*13 (D)*
7	0	7
	Результат:	$2000_{10} = 7D0_{16}$

2.3 Перевірка результатів

Перевірка трирозрядного десяткового числа

Число 7D в десяткову:

7 * 16^1 + D * 16^0 = 7 * 16 + 13 = 125.

Число 7D в двійкову:

7 — 0111; D — 1101; $7D_{16} = 01111101_2$

Компактна форма: $7D_{16} = 1111101_2$

Перевірка чотирирозрядного десяткового числа

Число 7D0 в десяткову:

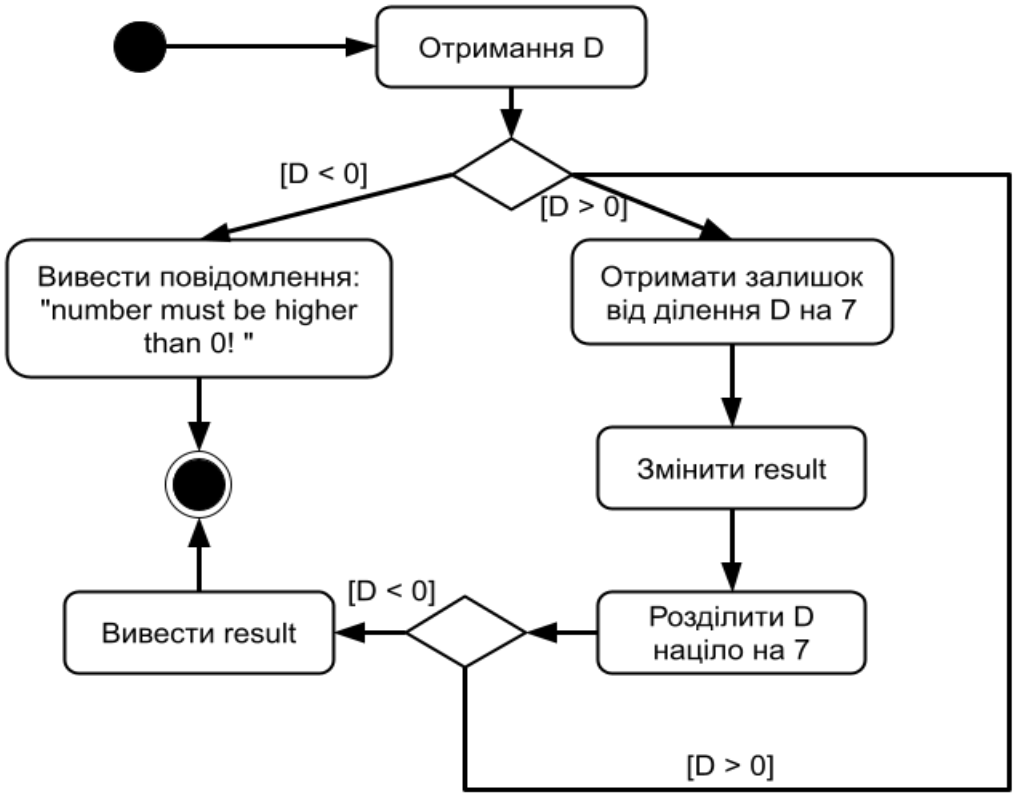
7 * 16^2 + D * 16^1 + 0 * 16^0 = 7 * 256 + 13 * 16 + 0 = 2000.

Число 7D0 в двійкову:
7 — 0111; D — 1101; 0 — 0000 $7D0_{16} = 011111010000_2$
Компактна форма: $7D_{16} = 11111010000_2$

3 ПЕРЕТВОРЕННЯ ЧИСЕЛ В _7_-РІЧНУ СИСТЕМУ ЧИСЛЕННЯ

Діаграму активності представлено на рис. 1

Рисунок 1 — Діаграма активності



Код програми наданий у Додатку А.

4 ДВІЙКОВА АРИФМЕТИКА

Покроковий опис_додавання__ чисел_125 та 2000__представлено *в табл.4.1*.

Таблиця 4.1 – Додавання двійкових чисел

перенесення	0	*1*	*1*	*1*	*1*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	**Перевірка**
4розр.		*1*	*1*	*1*	*1*	*1*	*0*	*1*	*0*	*0*	*0*	*0*	*2000 +*
3розр.						*1*	*1*	*1*	*1*	*1*	*0*	*1*	*125*
результат	1	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*1*	*1*	*0*	*1*	*2125*
перевірка	$1*2^{11}+0*2^{10}+0*2^{9}+0*2^{8}+0*2^{7}+1*2^{6}+0*2^{5}+$ $0*2^{4}+1*2^{3}+1*2^{2}+0*2^{1}+1*2^{0}=$ $=2048+0+0+0+0+64+0+0+8+4+0+1=2125$												

ВИСНОВКИ

Було вивчено різні види систем числення: десяткову, бінарну, шістнадцятиричну. Закріплено на практиці арифметичні дії з двійковою системою числення: додавання, віднімання, множення, ділення. Відпрацьовано в коді програми багаторазовий вибір завдань, поділ коду на функції, а також введення і виведення даних у консолі. Набуто навички переведення з десяткової системи обчислення в семеричну, яка є нетиповою в математиці та програмуванні. Виникли труднощі з додаванням двох бінарних чисел та оформленням пояснювальної записки, зокрема заповненням таблиць та діаграм для третього завдання.

ДОДАТОК А

```
#include <iostream>
#include <string>
#include <sstream>
#include <cmath>
using namespace std;
// task 1
void task1();
string decimalToBinary(int decimal);
int binaryToDecimal(const std::string &binary);
// task 2
void task2();
string decimalToHex(int decimal);
int hexToDecimal(const std::string& hexStr);
string hexToBinary(const string& hexStr);
// task 3
void task3();
string DecTo7(int D);
// task 4
void task4();
string binaryAddition(const string &a, const string &b);
// menu
int main () {
    cout << " " << endl;
    int task_num;  // diclaration
    do {
        cout << " Task number (0 - exit):  ";
        cin >> task_num;
        if (!cin) {
            cout << " Ups!" << endl; continue;
            // error notification
        }
        switch (task_num) {
            case 1 : task1(); break; // task 1
            case 2 : task2(); break; // task 2
            case 3 : task3(); break; // task 3
            case 4 : task4(); break; // task 4
            case 0 : cout << " The program is over!" << endl; break;
            default : cout << " Wrong task number!" << endl;
            // output for incorrect numbers
        }
    } while (task_num != 0); // end the program
    return 0;
}
// task 1
void task1(){
    cout << " ### Task 1 ### " << endl;
```

```
    int num1 = 125;
    int num2 = 2000;

    // converting decimal numbers to binary
    string binary1 = decimalToBinary(num1);
    string binary2 = decimalToBinary(num2);

    // output of conversion results
    cout << " Decimal number: " << num1 << " --> Binary number: " << binary1 <<
endl;
    cout << " Decimal number: " << num2 << " --> Binary number: " << binary2 <<
endl;

    // converting binary numbers back to decimal
    int reversedDecimal1 = binaryToDecimal(binary1);
    int reversedDecimal2 = binaryToDecimal(binary2);

    // outputting the results of the inverse conversion
    cout << " Binary number: " << binary1 << " --> Decimal number: " <<
reversedDecimal1 << endl;
    cout << " Binary number: " << binary2 << " --> Decimal number: " <<
reversedDecimal2 << endl;

    return;
}
// converting decimal numbers to binary
string decimalToBinary(int decimal) {
    string binary = "";
    while (decimal > 0) {
        binary = to_string(decimal % 2) + binary;
        decimal /= 2;
    }
    return binary;
}
// converting binary numbers to decimal
int binaryToDecimal(const string &binary) {
    int num = 0;
    int length = binary.length();
    for (int i = 0; i < length; ++i) {
        if (binary[length - i - 1] == '1') {
            num += pow(2, i);
        }
    }
    return num;
}
// task 2
void task2(){
    cout << " ### Task 2 ### " << endl;
    int num1 = 125;
    int num2 = 2000;
```

```
    // converting decimal numbers to hexadecimal
    string hex1 = decimalToHex(num1);
    string hex2 = decimalToHex(num2);

    // output of results
    cout << " Decimal number: " << num1 << " --> Hexadecimal number: " << hex1
<< endl;
    cout << " Decimal number: " << num2 << " --> Hexadecimal number: " << hex2
<< endl;

    // converting hexadecimal numbers back to decimal
    int dec1 = hexToDecimal(hex1);
    int dec2 = hexToDecimal(hex2);

    // outputting the results of the inverse conversion
    cout << " Hexadecimal number: " << hex1 << " --> Decimal number: " << dec1
<< endl;
    cout << " Hexadecimal number: " << hex2 << " --> Decimal number: " << dec2
<< endl;

    // converting hexadecimal numbers to binary
    string hexBinary1 = hexToBinary(hex1);
    string hexBinary2 = hexToBinary(hex2);

    // outputting binary conversions
    cout << " Hexadecimal number: " << hex1 << " --> Binary number: " <<
hexBinary1 << endl;
    cout << " Hexadecimal number: " << hex2 << " --> Binary number: " <<
hexBinary2 << endl;

    return;
}

// converting decimal numbers to hexadecimal
string decimalToHex(int decimal) {
    stringstream ss;
    ss << hex << decimal;
    string hexStr = ss.str();
    for (char &c : hexStr) {
        if (c >= 'a' && c <= 'f') {
            c = toupper(c);
        }
    }
    return hexStr;
}
// converting hexadecimal numbers to decimal
int hexToDecimal(const std::string& hexStr) {
    int decimal;
    stringstream ss;
```

```
    ss << hex << hexStr;
    ss >> decimal;
    return decimal;
}

// converting hexadecimal numbers to binary
string hexToBinary(const string& hexStr) {
    int decimal = hexToDecimal(hexStr);
    return decimalToBinary(decimal);
}
// task 3
void task3(){
    cout << " Task 3 " << endl;
    int number;
    cout << " Decimal number: ";
    cin >> number;
    // converting decimal numbers to septal
    cout << " Septal number: " << DecTo7(number) << endl;

    return;
}
// converting decimal numbers to septal
string DecTo7(int D) {
    string result = "";
    if (D > 0) {
        int remainder = D % 7;
        result = to_string(remainder) + result;
        D /= 7;
    }
    else {
        cout << "number must be higher than 0! " << endl;
    }
    return result;
}
// task 4
void task4(){
    cout << " Task 4 " << endl;
    int num1 = 125;
    int num2 = 2000;
    // converting decimal numbers to binary
    string binaryNum1 = decimalToBinary(num1);
    string binaryNum2 = decimalToBinary(num2);
    // adding two double numbers
    string binarySum = binaryAddition(binaryNum1, binaryNum2);
    // translation of the result into a dozen system for verification
    int decimalSum = binaryToDecimal(binarySum);
    // output of results
    cout << "Перше число (у двійковій системі): " << binaryNum1 << endl;
    cout << "Друге число (у двійковій системі): " << binaryNum2 << endl;
    cout << "Результат додавання (у двійковій системі): " << binarySum << endl;
```

```
    cout << "Результат додавання (у десятковій системі): " << decimalSum <<
endl;

    return;
}
// sum calculation
string binaryAddition(const string &a, const string &b) {
    string result = "";
    int carry = 0;
    // equalize the length of the lines by adding leading zeros to the shorter
line
    string bin1 = a.size() > b.size() ? a : string(b.size() - a.size(), '0') +
a;
    string bin2 = b.size() > a.size() ? b : string(a.size() - b.size(), '0') +
b;
    // add bit by bit from right to left
    for (int i = bin1.size() - 1; i >= 0; --i) {
        int sum = (bin1[i] - '0') + (bin2[i] - '0') + carry;
        result = to_string(sum % 2) + result;
        carry = sum / 2;
    }
    // if there is a carryover left, we add it
    if (carry) {
        result = "1" + result;
    }
    return result;
}
```