

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._____. 319 а __. __11__ РГР

Виконав студент гр. 319 а _____
(№ групи)

Єфімов Нікіта Максимович _____
(Підпис, дата) (П.І.Б.)

Перевірів к.т.н., доцент _____
(Науковий ступінь, вчене звання)

_____ О. В. Гавриленко
(Підпис, дата) (П.І.Б.)

Перетворення десяткових чисел у двійкову систему числення:

1. Вхідні дані: Десяткове число 166.
2. Покроковий процес:
 - Ділимо 166 на 2.
 - Запам'ятовуємо залишок (0 або 1).
 - Повторюємо кроки, доки частка не стане менше 2.
3. Приклад обчислень:
 - $166 / 2 = 83$ (залишок 0)

- $83 / 2 = 41$ (залишок 1)
 - $41 / 2 = 20$ (залишок 1)
 - $20 / 2 = 10$ (залишок 0)
 - $10 / 2 = 5$ (залишок 0)
 - $5 / 2 = 2$ (залишок 1)
 - $2 / 2 = 1$ (залишок 0)
 - $1 / 2 = 0$ (залишок 1)
4. Зчитуємо знизу вгору: 10100110. Це двійкове представлення числа 166.
5. Перевірка: Зворотне перетворення в десяткову систему:
- $1 * 2^7 + 0 * 2^6 + 1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$
 - $= 128 + 0 + 32 + 0 + 0 + 4 + 2 + 0$
 - $= 166$. Перевірка пройшла успішно.

Перетворення десяткових чисел у шістнадцяткову систему числення:

1. Вхідні дані: Десяткове число 166.
2. Покроковий процес:
 - Ділимо 166 на 16.
 - Запам'ятовуємо залишок (0-9 або A-F).
 - Повторюємо кроки, доки частка не стане менше 16.
3. Приклад обчислень:
 - $166 / 16 = 10$ (залишок A)
 - $10 / 16 = 0$ (залишок A)
4. Зчитуємо знизу вгору: AA. Це шістнадцяткове представлення числа 166.
5. Перевірка: Зворотне перетворення в десяткову систему:
 - $(A * 16^1) + (A * 16^0)$
 - $= (10 * 16^1) + (10 * 16^0)$
 - $= (10 * 16) + (10 * 1)$
 - $= 160 + 10$
 - $= 166$. Перевірка пройшла успішно.

Реалізація алгоритму у вигляді строкової функції на мові C++:

```
#include <iostream>
#include <string>

std::string DecToBinary(int decimal) {
```

```

std::string binary = "";
while (decimal > 0) {
    binary = std::to_string(decimal % 2) + binary;
    decimal /= 2;
}
return binary;
}

std::string DecToHex(int decimal) {
    std::string hex = "";
    while (decimal > 0) {
        int remainder = decimal % 16;
        if (remainder < 10)
            hex = std::to_string(remainder) + hex;
        else
            hex = char('A' + remainder - 10) + hex;
        decimal /= 16;
    }
    return hex;
}

int main() {
    int number = 166;
    std::cout << "Decimal: " << number << std::endl;
    std::cout << "Binary: " << DecToBinary(number) << std::endl;
    std::cout << "Hexadecimal: " << DecToHex(number) << std::endl;
    return 0;
}

```

Перетворення десяткового числа у двійкову систему числення:

1. Вхідні дані: Десяткове число 1826.
2. Покроковий процес:
 - Ділимо 1826 на 2.
 - Запам'ятовуємо залишок (0 або 1).
 - Повторюємо кроки, доки частка не стане менше 2.
3. Приклад обчислень:
 - $1826 / 2 = 913$ (залишок 0)
 - $913 / 2 = 456$ (залишок 1)

- $456 / 2 = 228$ (залишок 0)
 - $228 / 2 = 114$ (залишок 0)
 - $114 / 2 = 57$ (залишок 0)
 - $57 / 2 = 28$ (залишок 1)
 - $28 / 2 = 14$ (залишок 0)
 - $14 / 2 = 7$ (залишок 0)
 - $7 / 2 = 3$ (залишок 1)
 - $3 / 2 = 1$ (залишок 1)
 - $1 / 2 = 0$ (залишок 1)
4. Зчитуємо знизу вгору: 11100101010. Це двійкове представлення числа 1826.
5. Перевірка: Зворотне перетворення в десяткову систему:
- $(1 * 2^{10}) + (1 * 2^9) + (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (1 * 2^5) + (0 * 2^4) + (1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (0 * 2^0)$
 - $= 1024 + 512 + 256 + 0 + 0 + 32 + 0 + 8 + 0 + 2 + 0$
 - $= 1826$. Перевірка пройшла успішно.

Перетворення десяткових чисел у шістнадцяткову систему числення:

1. Вхідні дані: Десяткове число 1826.
2. Покроковий процес:
 - Ділимо 1826 на 16.
 - Запам'ятовуємо залишок (0-9 або A-F).
 - Повторюємо кроки, доки частка не стане менше 16.
3. Приклад обчислень:
 - $1826 / 16 = 114$ (залишок 2)
 - $114 / 16 = 7$ (залишок 2)
 - $7 / 16 = 0$ (залишок 7)
4. Зчитуємо знизу вгору: 722. Це шістнадцяткове представлення числа 1826.
5. Перевірка: Зворотне перетворення в десяткову систему:
 - $(7 * 16^2) + (2 * 16^1) + (2 * 16^0)$
 - $= (7 * 256) + (2 * 16) + (2 * 1)$
 - $= 1792 + 32 + 2$
 - $= 1826$. Перевірка пройшла успішно.

Діаграма активності алгоритму перетворення числа з десяткової системи числення в ____-річну:

Діаграма активності може бути створена для ілюстрації послідовності дій алгоритму перетворення числа з десяткової системи числення в будь-яку іншу систему числення.

Реалізація алгоритму у вигляді строкової функції на мові C++:

```
#include <iostream>

#include <string>

#include <sstream>

std::string DecToBinary(int decimal) {
    std::string binary = "";
    while (decimal > 0) {
        binary = std::to_string(decimal % 2) + binary;
        decimal /= 2;
    }
    return binary;
}

std::string DecToHex(int decimal) {
    std::stringstream ss;
    ss << std::hex << decimal;
    return ss.str();
}
```

```
}
```

```
int main() {  
  
    int number = 1826;  
  
    std::cout << "Decimal: " << number << std::endl;  
  
    std::cout << "Binary: " << DecToBinary(number) <<  
std::endl;  
  
    std::cout << "Hexadecimal: " << DecToHex(number) <<  
std::endl;  
  
    return 0;  
}
```

Після проведення операцій у двійковій системі числення та перевірки результатів у десятковій системі можна зробити наступні висновки:

Позиційні системи числення: Робота з двійковою системою числення вимагає великої уваги до позиції кожного розряду. Кожен розряд в двійковому числі має значення, яке залежить від його позиції.

Операції: Додавання, віднімання, множення та ділення у двійковій системі виконуються аналогічно операціям у десятковій системі, але з урахуванням особливостей двійкових чисел.

Перевірка результатів: Перевірка правильності результатів зазвичай виконується перетворенням результату у десяткову систему числення та порівнянням з результатом, отриманим із вхідними даними.

Ефективність алгоритмів: Робота з двійковими числами може бути ефективною, оскільки вони широко використовуються в обчислювальній техніці та програмуванні, особливо в сучасних комп'ютерах, де вся інформація зберігається у вигляді двійкових чисел.

Важливість правильного розуміння: Розуміння роботи з двійковими числами є ключовим елементом в обчислювальній техніці та програмуванні, оскільки багато операцій в цих галузях базуються на цій системі числення.

Загалом, робота з двійковими числами виконується за аналогічними принципами, що і з десятковими числами, але вимагає ретельного розуміння їхньої природи та особливостей.

1. Перетворення десяткових чисел в двійкову систему числення:
Нехай ми маємо десяткове число 166. Для перетворення цього числа в двійкову систему використовуємо ділення на 2.

Покроковий процес:

- 166 ділимо на 2: $166 / 2 = 83$, залишок 0
- 83 ділимо на 2: $83 / 2 = 41$, залишок 1
- 41 ділимо на 2: $41 / 2 = 20$, залишок 1
- 20 ділимо на 2: $20 / 2 = 10$, залишок 0
- 10 ділимо на 2: $10 / 2 = 5$, залишок 0
- 5 ділимо на 2: $5 / 2 = 2$, залишок 1
- 2 ділимо на 2: $2 / 2 = 1$, залишок 0
- 1 ділимо на 2: $1 / 2 = 0$, залишок 1

2. Зчитуючи залишки знизу вгору, отримаємо: 10100110.

Перевірка: Перетворимо це двійкове число назад у десяткове.

$$10100110 = 12^7 + 02^6 + 12^5 + 02^4 + 02^3 + 12^2 + 12^1 + 02^0 = 128 + 32 + 4 + 2 = 166. \text{ Результат співпадає з початковим числом.}$$

3. Перетворення десяткових чисел в шістнадцяткову систему числення:

Повторимо аналогічний процес, але тепер будемо ділити на 16.

Покроковий процес:

- 166 ділимо на 16: $166 / 16 = 10$, залишок 6 (6 в шістнадцятковій системі позначається як A)
- 10 ділимо на 16: $10 / 16 = 0$, залишок 10 (A)

4. Отже, 166 в шістнадцятковій системі буде представлено як A6.

Перевірка: Перетворимо це шістнадцяткове число назад у

десяткове. $A6 = 1016^1 + 616^0 = 160 + 6 = 166$. Результат співпадає з початковим числом.

5. Діаграма активності алгоритму перетворення числа з десятичної системи числення в 17-річну:

6. Реалізація алгоритму у вигляді строкової функції DecTo_N_ (D) на мові C++:

```
#include <iostream>

#include <string>

std::string DecToBaseN(int decimal, int base) {

    std::string result = "";

    while (decimal > 0) {

        int remainder = decimal % base;

        char digit;

        if (remainder < 10) {

            digit = remainder + '0';

        } else {

            digit = remainder - 10 + 'A';

        }

        result = digit + result;

        decimal /= base;

    }

    return result;

}
```



```
}
```

```
int main() {  
  
    int decimalNumber = 166;  
  
    int base = 17;  
  
    std::string baseNNumber = DecToBaseN(decimalNumber,  
base);  
  
    std::cout << "Decimal " << decimalNumber << " in base "  
<< base << " is: " << baseNNumber << std::endl;  
  
    return 0;  
}
```

1. Операція у двійковій системі числення:

Для двійкових чисел 10100110 та 11001001, спробуємо виконати операцію, наприклад, додавання.

Покроковий процес:

10100110 (166 в десятковій) +

- 11001001 (201 в десятковій)
- Результат: 101101111 (367 в десятковій)

2. Перевірка: Перетворимо це двійкове число назад у десяткове.

$101101111 = 12^8 + 12^7 + 12^6 + 12^4 + 12^3 + 12^1 + 1 \cdot 2^0 = 256 + 128 + 64 + 32 + 8 + 2 + 1 = 367$. Результат співпадає з очікуванням.

3.

4. Перетворення десяткових чисел в двійкову систему числення:

Нехай ми маємо десяткове число 1826. Для перетворення цього числа в двійкову систему використовуємо ділення на 2.

Покроковий процес:

- 1826 ділимо на 2: $1826 / 2 = 913$, залишок 0
 - 913 ділимо на 2: $913 / 2 = 456$, залишок 1
 - 456 ділимо на 2: $456 / 2 = 228$, залишок 0
 - 228 ділимо на 2: $228 / 2 = 114$, залишок 0
 - 114 ділимо на 2: $114 / 2 = 57$, залишок 0
 - 57 ділимо на 2: $57 / 2 = 28$, залишок 1
 - 28 ділимо на 2: $28 / 2 = 14$, залишок 0
 - 14 ділимо на 2: $14 / 2 = 7$, залишок 0
 - 7 ділимо на 2: $7 / 2 = 3$, залишок 1
 - 3 ділимо на 2: $3 / 2 = 1$, залишок 1
 - 1 ділимо на 2: $1 / 2 = 0$, залишок 1
5. Зчитуючи залишки знизу вгору, отримаємо: 11100111010.
 Перевірка: Перетворимо це двійкове число назад у десяткове.
 $11100111010 = 12^{10} + 12^9 + 12^8 + 12^5 + 12^4 + 12^3 + 1 \cdot 2^1 = 1024 + 512 + 256 + 32 + 16 + 8 + 2 = 1826$. Результат співпадає з початковим числом.
6. Перетворення десяткових чисел в шістнадцяткову систему числення:
 Повторимо аналогічний процес, але тепер будемо ділити на 16.
 Покроковий процес:
- 1826 ділимо на 16: $1826 / 16 = 114$, залишок 2
 - 114 ділимо на 16: $114 / 16 = 7$, залишок 2
 - 7 ділимо на 16: $7 / 16 = 0$, залишок 7
7. Отже, 1826 в шістнадцятковій системі буде представлено як 722.
 Перевірка: Перетворимо це шістнадцяткове число назад у десяткове.
 $722 = 7 \cdot 16^2 + 2 \cdot 16^1 + 2 \cdot 16^0 = 1792 + 32 + 2 = 1826$.
 Результат співпадає з початковим числом.
8. Діаграма активності алгоритму перетворення числа з десяткової системи числення в 17-річну:
9. Реалізація алгоритму у вигляді строкової функції DecTo_N_ (D) на мові C++:

```

10. #include <iostream>
11. #include <string>
12.
13. std::string DecToBaseN(int decimal, int base) {
14.     std::string result = "";

```

```

15. while (decimal > 0) {
16.     int remainder = decimal % base;
17.     char digit;
18.     if (remainder < 10) {
19.         digit = remainder + '0';
20.     } else {
21.         digit = remainder - 10 + 'A';
22.     }
23.     result = digit + result;
24.     decimal /= base;
25. }
26. return result;
27. }
28.
29. int main() {
30.     int decimalNumber = 1826;
31.     int base = 17;
32.     std::string baseNNumber =
        DecToBaseN(decimalNumber, base);
33.     std::cout << "Decimal " << decimalNumber << " in
        base " << base << " is: " << baseNNumber <<
        std::endl;
34.     return 0;
35. }

```

1. Операція у двійковій системі числення:

Для двійкових чисел проведемо операцію додавання, наприклад.

Покроковий процес:

36. 11100111010 (1826 в десятковій) +
 - 10101111111 (1495 в десятковій)
 - Результат: 110101110101 (3321 в десятковій)

2. Перевірка: Перетворимо це двійкове число назад у десяткове.

$110101110101 = 12^{11} + 12^{10} + 12^9 + 12^7 + 12^6 + 12^4 + 12^2 + 12^0 = 2048 + 1024 + 512 + 128 + 64 + 16 + 4 + 1 = 3321$.
 Результат співпадає з очікуванням.

37.