

Міністерство освіти і науки України
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ім. Богдана Хмельницького

Факультет Обчислювальної техніки, інтелектуальних та управляючих систем
Кафедра Програмного забезпечення автоматизованих систем

ЛАБОРАТОРНА РОБОТА № 2

з дисципліни «Вступ до інженерії програмного забезпечення»

Тема: Переведення дробових чисел з однієї системи числення
в іншу. Перетворення цілих даних з прямого коду в додатковий

Варіант 7

Виконав: студент гр. КС-231
Киба Д.В.

1.1 Постановка завдання :

Ознайомитися зі структурою машинних форматів представлення дробових даних. Розробити та реалізувати алгоритм переведення дробових даних з десяткової системи числення у двійкову. Реалізувати алгоритм переведення цілих даних з прямого коду у додатковий та навпаки.

a) Переведення дробових чисел $10 \rightarrow 2$

b) Перетворення цілих чисел з прямого коду в додатковий при довжині розрядної сітки 8.

1.2 Текст програми для вирішення завдань (лістинг):

```
package main

import (
    "fmt"
    "math"
)

func main() {

    fmt.Println("Introduction to Software Engineering,
Labwork #2\nVariant 7")

    tryAgain:

        fmt.Println("Choose the case:\n1) convert a
decimal fraction to binary;\n2) convert an integer
from a direct code to an additional code with a bit
grid length of 8.")
        var choice int
```

```
fmt.Scan(&choice)

switch choice {
case 1:
    fmt.Printf("Enter the decimal fraction: ")
    var frac float64
    fmt.Scan(&frac)
    fmt.Println(convertDecimalFracToBinary(frac))
    goto tryAgain
case 2:
    fmt.Printf("Enter a direct code integer in
binary with the length of 8. (for example:
1|0100110): ")
    var direct string
    fmt.Scan(&direct)
    fmt.Println(convertDirectIntoAdditional(direct))
    goto tryAgain
case 0:
    break
}

}

func convertDecimalFracToBinary(frac float64) string
```

```

{
    var temp int
    precision := 8
    result := "0."
    for i := 0; i < precision; i++ {
        frac *= 2
        temp = int(math.Floor(float64(frac)))
        if temp > 1 {
            temp--
        }
        if frac ≥ 1 {
            frac--
        }
        result += fmt.Sprint(temp)
    }
    result += " (2)\n"

    return result
}

```

```

func convertDirectIntoAdditional(direct string)
string {
    length := len(direct)
    additional := []byte(direct[2:])

```

```

if direct[0] == '0' {
    return direct
} else if direct[0] == '1' {
    for i := 0; i < length-2; i++ {
        if additional[i] == '0' {
            additional[i] = '1'
        } else if additional[i] == '1' {
            additional[i] = '0'
        }
    }
    additional = addOneToBinary(additional)
}

return "1|" + string(additional)
}

```

```

func addOneToBinary(binary []byte) []byte {
    carry := 1
    for i := len(binary) - 1; i ≥ 0; i-- {
        if carry == 0 {
            break
        }
        sum := int(binary[i] - '0') + carry
    }
}

```

```

        binary[i] = byte(sum%2) + '0'

        carry = sum / 2
    }

    if carry > 0 {
        binary = append([]byte{'1'}, binary...)
    }

    return binary
}

}

```

1.3 Результат виконання програми:

```

o → 2_FracToBinary git:(main) X go run main.go
Introduction to Software Engineering, Labwork #2
Variant 7
Choose the case:
1) convert a decimal fraction to binary;
2) convert an integer from a direct code to an additional code with a bit grid
length of 8.
1
Enter the decimal fraction: 0.31
0.01001111 (2)

```

```

o → 2_FracToBinary git:(main) X go run main.go
Introduction to Software Engineering, Labwork #2
Variant 7
Choose the case:
1) convert a decimal fraction to binary;
2) convert an integer from a direct code to an additional code with a bit grid length of 8.
2
Enter a direct code integer in binary with the length of 8. (for example: 1|01
00110): 1|0100110
1|1011010
Choose the case:
1) convert a decimal fraction to binary;
2) convert an integer from a direct code to an additional code with a bit grid length of 8.
2
Enter a direct code integer in binary with the length of 8. (for example: 1|0100110): 0|0110111
0|0110111

```

1.4 Висновок:

У даній лабораторній роботі були реалізовані два випадки конвертації чисел: перетворення десяткового дробу в двійковий та перетворення цілого числа з прямого коду в додатковий код.

Для перетворення десяткового дробу в двійковий використовується метод множення на 2 та відкидання цілої частини. Реалізована функція `convertDecimalFracToBinary(frac float64)` приймає десятковий дріб, множить його на 2 та відкидає цілу частину, зберігаючи дрібну у двійковому вигляді з заданою точністю.

Для перетворення цілого числа з прямого коду в додатковий код спочатку перевіряється знак числа. Якщо число від'ємне, його код змінюється на додатковий: кожен біт обертається (0 на 1 і навпаки), а потім до результату додається одиниця у двійковому вигляді. Реалізована функція `convertDirectIntoAdditional(direct string)` приймає число у прямому коді та повертає його у додатковому.

Обидва алгоритми реалізовані як функції у мові програмування Go. Користувач може вибрати операцію, яку він хоче виконати, і ввести відповідні дані. Програма повертає результат в стандартний вивід.

Таким чином, реалізація цих алгоритмів дозволяє зрозуміти нам, як відбувається конвертація чисел між різними системами числення.