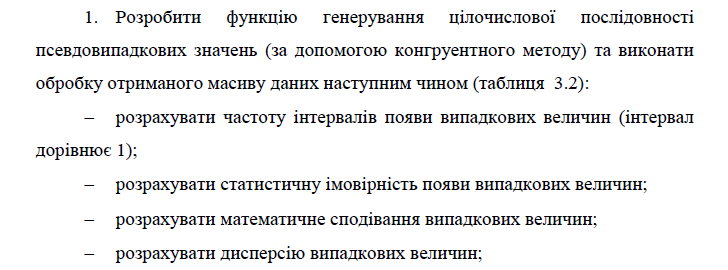
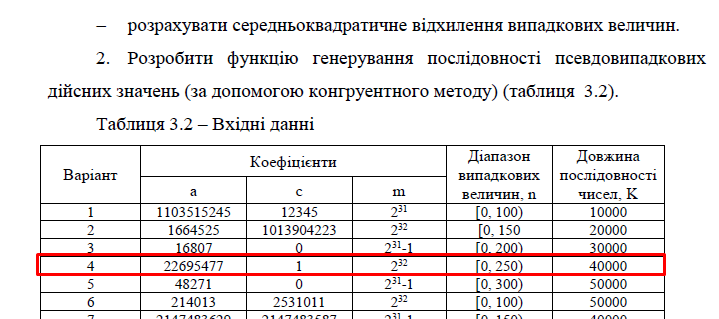
**ЛАБОРАТОРНА РОБОТА № 3**

**Управляючі конструкції в мові Go. Функції. Введення та виведення інформації у консоль**

***Мета:*** ознайомитися з управляючими конструкціями мови програмування Go, навчитися користуватися функціями вводу/виводу та створювати користувацькі функції. Ознайомитись з методами генерування випадкових чисел.

**Хід роботи**





Функція генерування чисел

func lcm(x0, a, c, m, k int) []int {

    results := []int{}

*for* i := 1; i <= k; i++ {

        x0 = (a\*x0 + c) % m

        results = append(results, x0%250)

    }

*return* results

}

func main() {

    x0 := 1

    var a, c, m, k int

    a = 22695477

    c = 1

    m = 4294967296 *// 2^32*

    k = 400

    randNums := lcm(x0, a, c, m, k)

*for* i, value := *range* randNums {

        fmt.Println("Value", i+1, "->", value)

    }

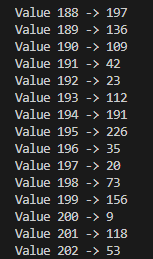
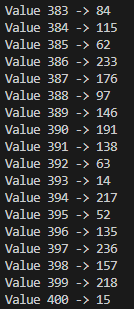
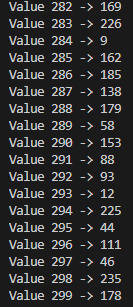
 

Рис.1. Результат

Розрахунок частоти інтервалів та статистичної імовірності

func checked(arr []int, k, num int) float64 {

    var freq float64 = 0

*for* i := 0; i < k; i++ {

*if* arr[i] == num {

            freq++

        }

    }

*return* freq

}

*// імовірність появи*

    frequency := [400]float64{}

    fmt.Println("\nЧастота інтервалів появи випадкових чисел")

*for* i := 0; i < k; i++ {

        frequency[i] = checked(randNums, k, randNums[i])

        fmt.Println("Value №", i+1, "=", frequency[i])

    }

*// Статистична імовірність*

    fmt.Println("\nСтатистична імовірність")

    stat := [400]float64{}

*for* i := 0; i < k; i++ {

        stat[i] = (frequency[i] / 400)

    }

*for* i, value := *range* stat {

        fmt.Println("Value №", i+1, "->", value)

    }

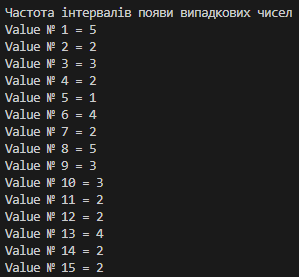
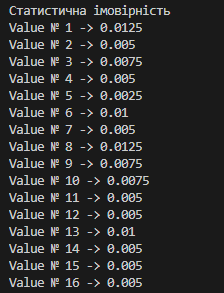
 

Рис.2. Результат

Математичне сподівання, дисперсія та середньоквадратичне відхилення

*// математичне сподівання*

    var mathExp float64 = 0

*for* i := 0; i < k; i++ {

        mathExp += float64(i) \* stat[i]

    }

    fmt.Println("Математичне сподівання ->", mathExp)

*// дисперсія*

    var despersion float64 = 0

*for* i := 0; i < k; i++ {

        despersion += math.Pow((float64(i)-mathExp), 2) \* stat[i]

    }

    fmt.Println("Дисперсія ->", despersion)

*// середньоквадратичне відхилення*

    fmt.Println("Середньоквадратичне відхилення ->", math.Sqrt(despersion))



Рис.3. Результат

Функція генерування послідовності псевдовипадкових дійсних чисел:

package main

import "fmt"

func lcm(x0, a, c, m, k int) []int {

    results := []int{}

*for* i := 1; i <= k; i++ {

        x0 = (a\*x0 + c) % m

        results = append(results, x0%25000)

    }

*return* results

}

func main() {

    x0 := 1

    var a, c, m, k int

    a = 22695477

    c = 1

    m = 4294967296 *// 2^32*

    k = 400

    randNums := lcm(x0, a, c, m, k)

    randNumsFloat := [400]float64{}

*for* i := 0; i < k; i++ {

        randNumsFloat[i] = float64(randNums[i]) / 100

    }

*for* i, value := *range* randNumsFloat {

        fmt.Println("Value", i+1, "->", value)

    }

}

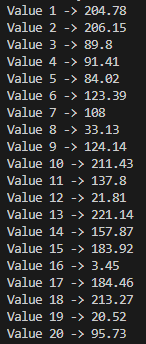
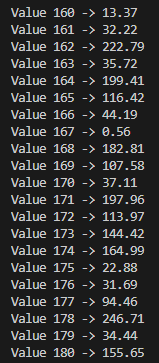
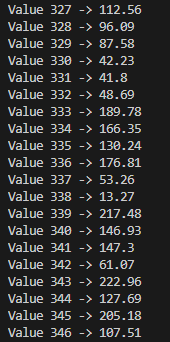
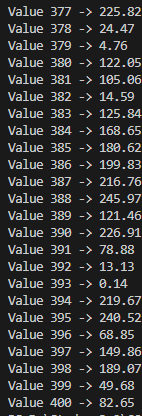
   

Рис.4. Результат

***Висновки:*** у ході виконання лабораторної роботи було ознайомлено з управляючими конструкціями мови програмування Go, навчено, як користуватися функціями вводу/виводу та створювати користувацькі функції, ознайомлено з методами генерування випадкових чисел.