**Лабораторна робота №1**

**Обчислення машинного епсілону**

**Мета роботи**: набуття теоретичних знань та практичних навичок аналізу розрахунків на ЕОМ.

Завдання

Виконати обчислення машинного епсілону для типу даних double, float. Вказати час витрачений на розрахунок та кількість ітерацій.

Короткі теоретичні відомості

Машинний нуль – числове значення з таким негативним порядком, яке сприймається машиною як нуль.

Машинний епсилон (англ. Machine epsilon) – числове значення, менше якого неможливо ставити відносну точність для будь-якого алгоритму, що повертає речові числа. Абсолютне значення «машинного епсилон» залежить від розрядності сітки застосовується ЕОМ, типу (розрядності) використовуваних при розрахунках чисел, і від прийнятої в конкретному трансляторі структури подання дійсних чисел (кількості біт, що відводяться на мантиссу і на порядок). Формально машинний епсилон зазвичай визначають як мінімальне з чисел ε, для якого 1 + ε> 1 при машинних розрахунках з числами даного типу. Альтернативне визначення - максимальне ε, для якого справедливо рівність 1 + ε = 1.

Практична важливість машинного епсилон пов'язана з тим, що два (відмінних від нуля) числа є однаковими з точки зору машинної арифметики, якщо їх відносна різниця по модулю менше (при визначенні першого типу) або не перевищує (при визначенні другого типу) машинного епсилон.

Хід роботи

Алгоритм знаходження машинного епсілону можна представити за допомогою наступної блок-схеми:



Використовуючи мову програмування GoLang напише програму, що реалізує даний алгоритм:

package main

import "fmt"

func main() {

fmt.Println("Calculating Float Epsilon:")

fEps := calcFloatEps()

fmt.Println("\nCalculating Double Epsilon:")

dEps := calcDoubleEps()

fmt.Println("\nResults:")

fmt.Printf("Float Epsilon: %v\n", fEps)

fmt.Printf("Double Epsilon: %v\n", dEps)

}

func calcFloatEps() float32 {

eps := float32(1)

for ; eps+1 > 1; eps /= 2 {

fmt.Printf("1 + %v = %v\n", eps, eps+1)

}

fmt.Printf("1 + %v = %v\n", eps, eps+1)

return eps \* 2

}

func calcDoubleEps() float64 {

eps := float64(1)

for ; eps+1 > 1; eps /= 2 {

fmt.Printf("1 + %v = %v\n", eps, eps+1)

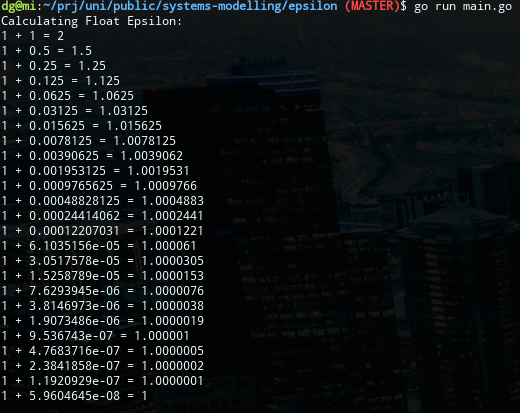
}

fmt.Printf("1 + %v = %v\n", eps, eps+1)

return eps \* 2

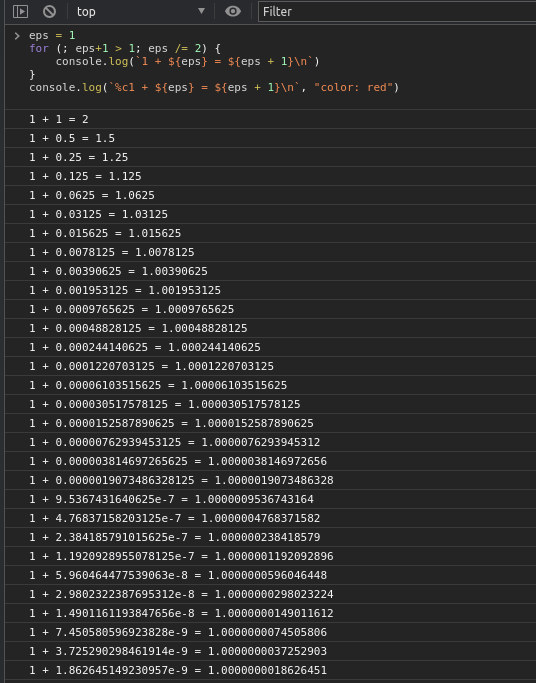
}

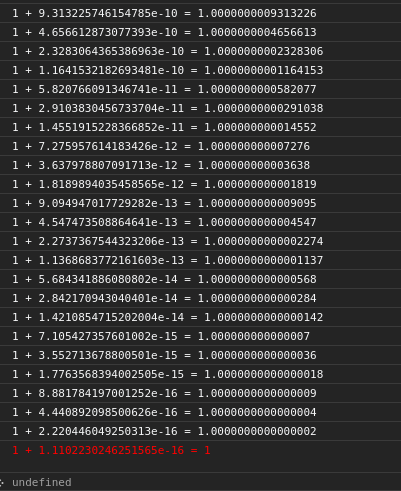
Результат виконання програми:





Також, спробуємо знайти епсілон у мові JavaScript, яка має лише один числовий тип даних – number (за винятком BigInt та інших спеціалізованих можливостей мови):





Можна бачити, що тип number у мові JavaScript має таку ж точність як float64 у мові GoLang (double).

**Висновок:** виконавши лабораторну роботу, я набув теоретичних знань та практичних навичок аналізу розрахунків на ЕОМ. За результатами обчислень можна бачити, що тип даних double (float64 у мові GoLang) має більшу точність обчислень, ніж float, що зрозуміло, адже для чисел типу double у оперативній пам’яті виділяється у два рази більше байтів – 4 для float, та 8 для double.