

## 1. Closure-based Calculator

Write a Groovy program that defines a method `calculate(a, b, operation)` that takes two numbers and a closure. Use this to:

- **Add two numbers**

```
def calculate(a, b, operation) {  
    return operation(a, b)}  
  
def add = { x, y -> x + y }  
  
def a1 = 10  
  
def b1 = 5  
  
println "=== Addition ==="  
  
println "$a1 + $b1 = " + calculate(a1, b1, add)
```

- **Subtract two numbers**

```
def calculate(a, b, operation) {  
    return operation(a, b)}  
  
def subtract = { x, y -> x - y }  
  
def a2 = 10  
  
def b2 = 5  
  
println "\n=== Subtraction ==="  
  
println "$a2 - $b2 = " + calculate(a2, b2, subtract)
```

- **Multiply two numbers**

```
def calculate(a, b, operation) {  
    return operation(a, b)}
```

```

def multiply = { x, y -> x * y }

def a3 = 10

def b3 = 5

println "\n=== Multiplication ==="

println "$a3 * $b3 = " + calculate(a3, b3, multiply)

```

## 2. Word Frequency Counter

Ask the user for a sentence and count how many times each word appears using a map.

**Input:** "hello world hello"

**Output:**

```

hello - 2
world - 1

```

### PROGRAMS:

```

def sentence="hello world hello"
def words=sentence.split(" ")
def map=[:]
words.each { word->map[word]=(map[word]?:0)+1
}
map.each { word,count->println"$word → $count"
}

```

## 3. Group Strings by Length

Given a list of words, group them into a map where the key is the word length and the value is a list of words with that length.

**Input:** ["hi", "hello", "bye", "good", "sun"]

**Output:**

```
2 - ["hi"]
3 - ["bye", "sun"]
4 - ["good"]
5 - ["hello"]
```

### **PROGRAMS:**

```
def words=["hi","hello","bye","good","sun"]
def map=[:]
words.each{ word ->
  def len=word.length()
  map[len]=(map[len]?:[])+word
}
map.each {k,v ->
  println "$k → $v"
}
```