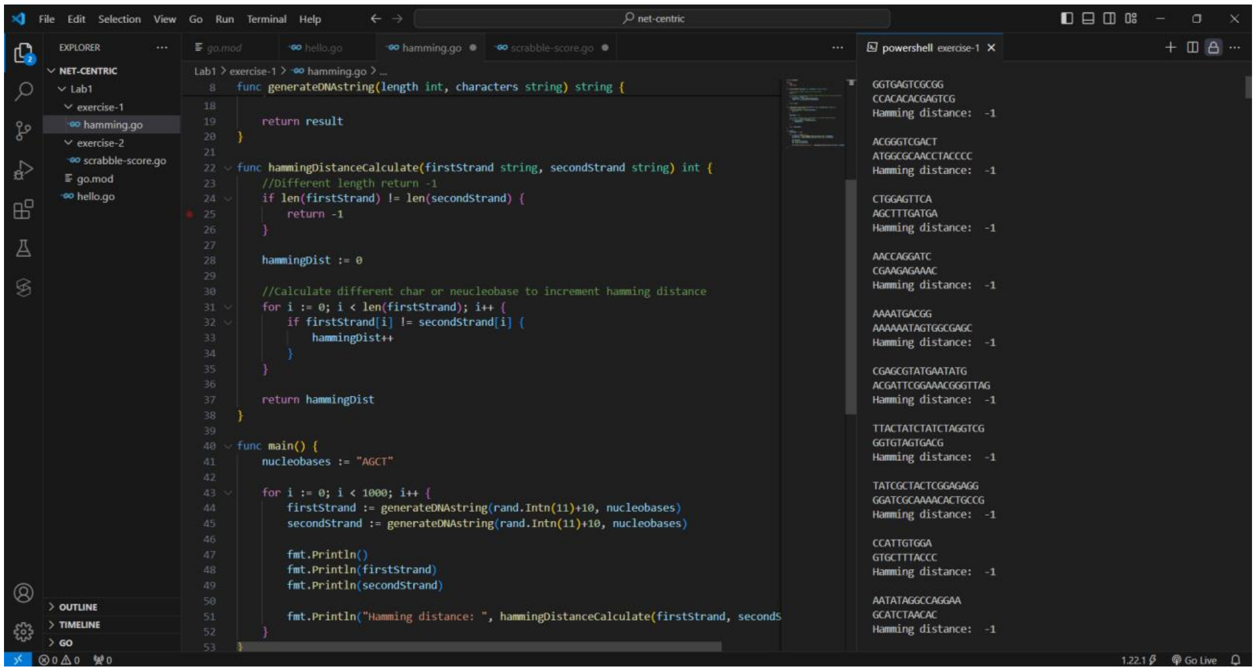
Hồ Hữu Hiệp - ITITIU20202
Net-Centric Programming – Lab 1

Net-centric Programming

Lab1: Introduction to Golang

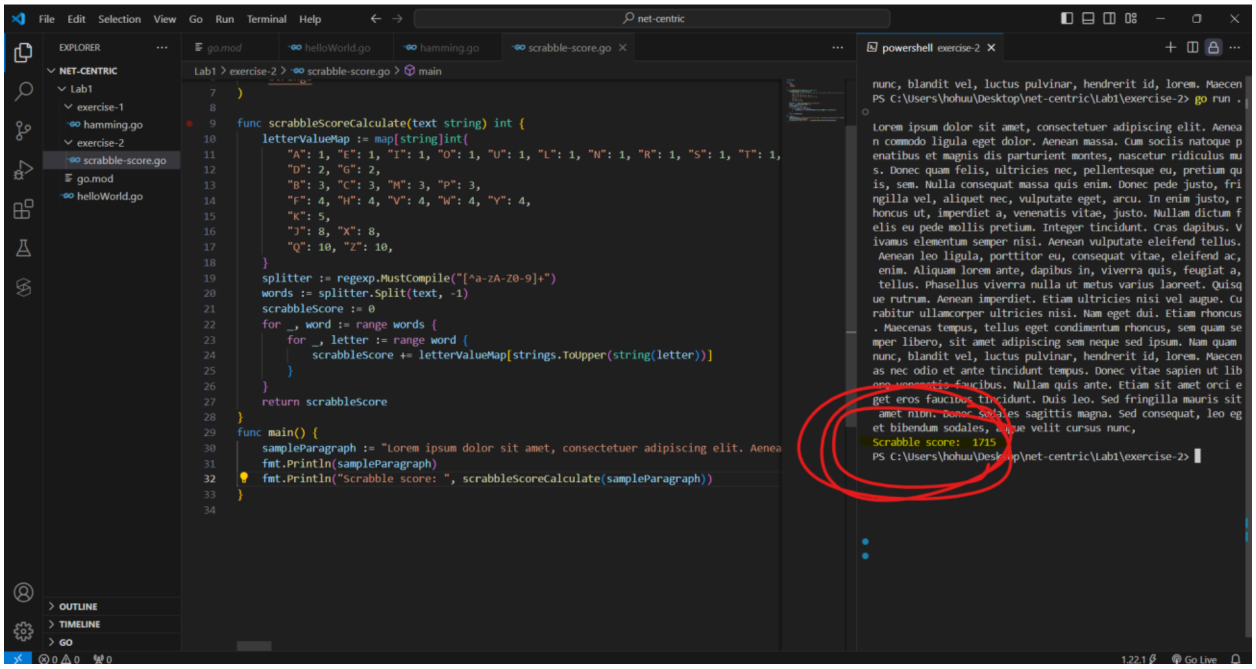
1. Hamming Distance



```go
func generateDNAstring(length int, characters string) string {

    return result
}

func hammingDistanceCalculate(firstStrand string, secondStrand string) int {
    //Different length return -1
    if len(firstStrand) != len(secondStrand) {
        return -1
    }

    hammingDist := 0

    //Calculate different char or neucleobase to increment hamming distance
    for i := 0; i < len(firstStrand); i++ {
        if firstStrand[i] != secondStrand[i] {
            hammingDist++
        }
    }

    return hammingDist
}

func main() {
    nucleobases := "AGCT"

    for i := 0; i < 1000; i++ {
        firstStrand := generateDNAstring(rand.Intn(11)+10, nucleobases)
        secondStrand := generateDNAstring(rand.Intn(11)+10, nucleobases)

        fmt.Println()
        fmt.Println(firstStrand)
        fmt.Println(secondStrand)

        fmt.Println("Hamming distance: ", hammingDistanceCalculate(firstStrand, secondS
    }
}
```

```
GGTGAGTCGCGG
CCACACACGAGTCG
Hamming distance:  -1

ACGGGTCGACT
ATGGCGCAACCTACCCC
Hamming distance:  -1

CTGGAGTTCA
AGCTTTGATGA
Hamming distance:  -1

AACCAGGATC
CGAAGAGAAAC
Hamming distance:  -1

AAAATGACGG
AAAAAATAGTGGCGAGC
Hamming distance:  -1

CGAGCGTATGAATATG
ACGATTCGGAAACGGGTTAG
Hamming distance:  -1

TTACTATCTATCTAGGTCG
GGTGTAGTGACG
Hamming distance:  -1

TATCGCTACTCGGAGAGG
GGATCGCAAAACACTGCCG
Hamming distance:  -1

CCATTGTGGA
GTGCTTTACCC
Hamming distance:  -1

AATATAGGCCAGGAA
GCATCTAACAC
Hamming distance:  -1
```

2. Scrabble Score

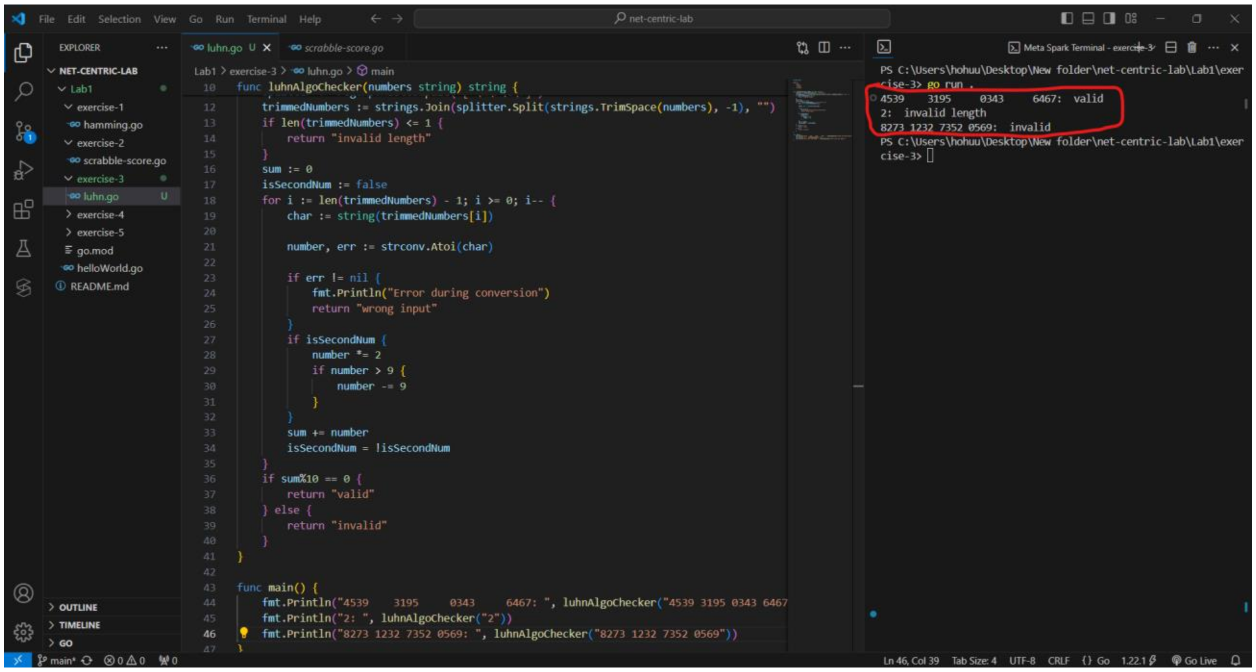


```go
)

func scrabbleScoreCalculate(text string) int {
    letterValueMap := map[string]int{
        "A": 1, "E": 1, "I": 1, "O": 1, "U": 1, "L": 1, "N": 1, "R": 1, "S": 1, "T": 1,
        "D": 2, "G": 2,
        "B": 3, "C": 3, "M": 3, "P": 3,
        "F": 4, "H": 4, "V": 4, "W": 4, "Y": 4,
        "K": 5,
        "J": 8, "X": 8,
        "Q": 10, "Z": 10,
    }
    splitter := regexp.MustCompile("[^a-zA-Z0-9]+")
    words := splitter.Split(text, -1)
    scrabbleScore := 0
    for _, word := range words {
        for _, letter := range word {
            scrabbleScore += letterValueMap[strings.ToUpper(string(letter))]
        }
    }
    return scrabbleScore
}
func main() {
    sampleParagraph := "Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenea
    fmt.Println(sampleParagraph)
    fmt.Println("Scrabble score: ", scrabbleScoreCalculate(sampleParagraph))
}
```

```
nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecen
PS C:\Users\hohuu\Desktop\net-centric\Lab1\exercise-2> go run .

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenea
n commodo ligula eget dolor. Aenean massa. Cum sociis natoque p
enatibus et magnis dis parturient montes, nascetur ridiculus mu
s. Donec quam felis, ultricies nec, pellentesque eu, pretium qu
is, sem. Nulla consequat massa quis enim. Donec pede justo, fri
ngilla vel, aliquet nec, vulputate eget, arcu. In enim justo, r
honcus ut, imperdiet a, venenatis vitae, justo. Nullam dictum f
elis eu pede mollis pretium. Integer tincidunt. Cras dapibus. V
ivamus elementum semper nisi. Aenean vulputate eleifend tellus.
 Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac,
 enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a,
 tellus. Phasellus viverra nulla ut metus varius laoreet. Quisq
ue rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Cu
rabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus
. Maecenas tempus, tellus eget condimentum rhoncus, sem quam se
mper libero, sit amet adipiscing sem neque sed ipsum. Nam quam
nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecen
as nec odio et ante tincidunt tempus. Donec vitae sapien ut lib
ero venenatis faucibus. Nullam quis ante. Etiam sit amet orci e
get eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit
 amet nibh. Donec sodales sagittis magna. Sed consequat, leo eg
et bibendum sodales, augue velit cursus nunc,
Scrabble score:  1715
PS C:\Users\hohuu\Desktop\net-centric\Lab1\exercise-2>
```

Hồ Hữu Hiệp - ITITIU20202
Net-Centric Programming – Lab 1

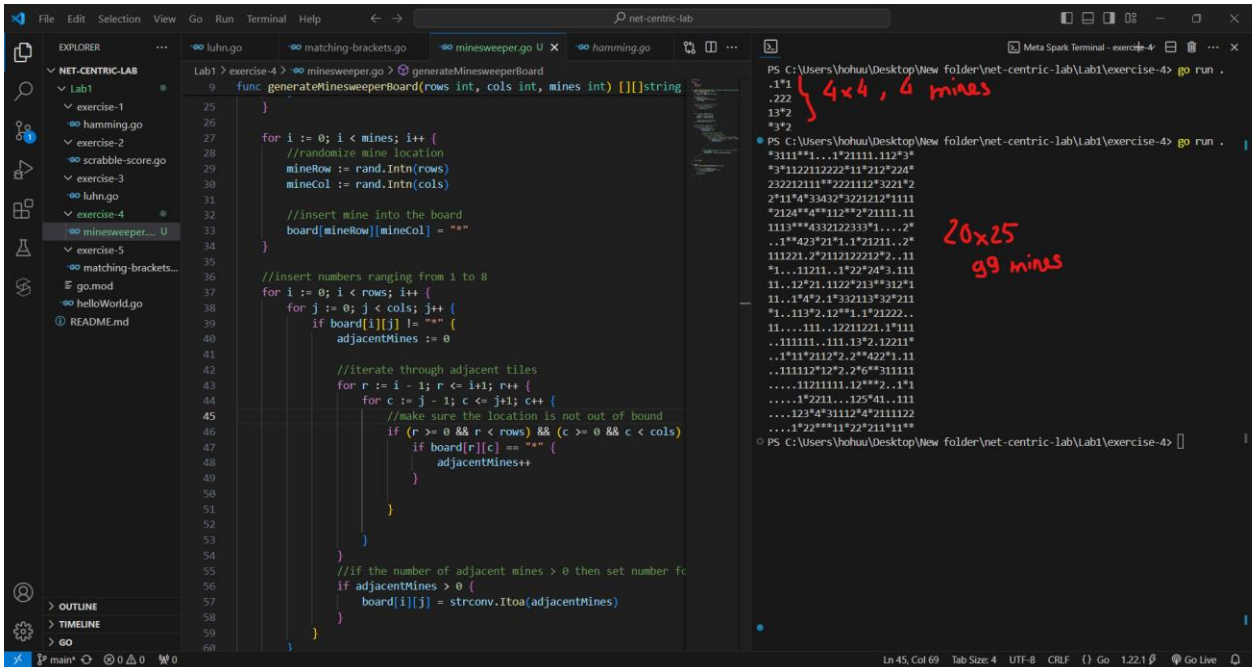
3. Luhn



```go
func luhnAlgoChecker(numbers string) string {
	trimmedNumbers := strings.Join(splitter.Split(strings.TrimSpace(numbers), -1), "")
	if len(trimmedNumbers) <= 1 {
		return "invalid length"
	}
	sum := 0
	isSecondNum := false
	for i := len(trimmedNumbers) - 1; i >= 0; i-- {
		char := string(trimmedNumbers[i])

		number, err := strconv.Atoi(char)

		if err != nil {
			fmt.Println("Error during conversion")
			return "wrong input"
		}
		if isSecondNum {
			number *= 2
			if number > 9 {
				number -= 9
			}
		}
		sum += number
		isSecondNum = !isSecondNum
	}
	if sum%10 == 0 {
		return "valid"
	} else {
		return "invalid"
	}
}

func main() {
	fmt.Println("4539    3195     0343     6467: ", luhnAlgoChecker("4539 3195 0343 6467
	fmt.Println("2: ", luhnAlgoChecker("2"))
	fmt.Println("8273 1232 7352 0569: ", luhnAlgoChecker("8273 1232 7352 0569"))
}
```

```
PS C:\Users\hohuu\Desktop\New folder\net-centric-lab\Lab1\exer
cise-3> go run .
4539    3195     0343     6467:  valid
2:  invalid length
8273 1232 7352 0569:  invalid
PS C:\Users\hohuu\Desktop\New folder\net-centric-lab\Lab1\exer
cise-3> 
```

4. Minesweeper

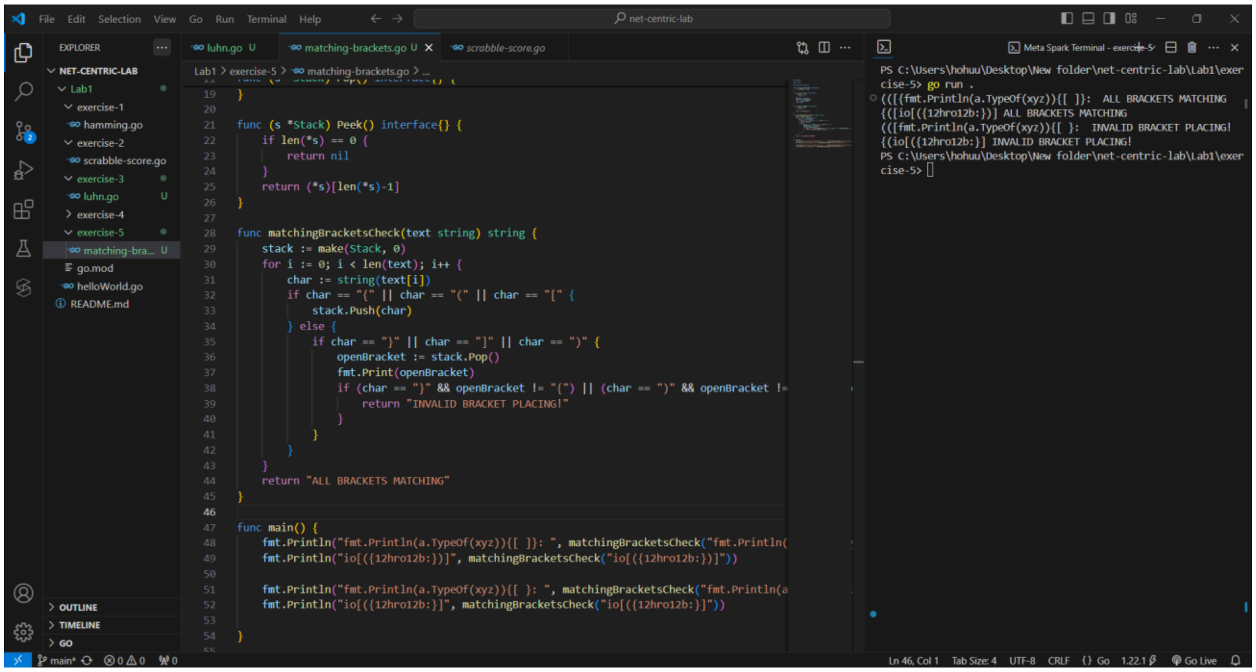


```go
func generateMinesweeperBoard(rows int, cols int, mines int) [][]string
	}

	for i := 0; i < mines; i++ {
		//randomize mine location
		mineRow := rand.Intn(rows)
		mineCol := rand.Intn(cols)

		//insert mine into the board
		board[mineRow][mineCol] = "*"
	}

	//insert numbers ranging from 1 to 8
	for i := 0; i < rows; i++ {
		for j := 0; j < cols; j++ {
			if board[i][j] != "*" {
				adjacentMines := 0

				//iterate through adjacent tiles
				for r := i - 1; r <= i+1; r++ {
					for c := j - 1; c <= j+1; c++ {
						//make sure the location is not out of bound
						if (r >= 0 && r < rows) && (c >= 0 && c < cols)
							if board[r][c] == "*" {
								adjacentMines++
							}

						}

					}
				}
				//if the number of adjacent mines > 0 then set number fc
				if adjacentMines > 0 {
					board[i][j] = strconv.Itoa(adjacentMines)
				}
			}
		}
```

```
PS C:\Users\hohuu\Desktop\New folder\net-centric-lab\Lab1\exercise-4> go run .
.1*1
.222
13*2
*3*2
PS C:\Users\hohuu\Desktop\New folder\net-centric-lab\Lab1\exercise-4> go run .
*3111**1...1*21111.112*3*
*3*1122112222*11*212*224*
232212111**2221112*3221*2
2*11*4*33432*3221212*1111
*2124**4**112**2*21111.11
1113***4332122333*1....2*
..1**423*21*1.1*21211..2*
111221.2*2112122212*2..11
*1...11211..1*22*24*3.111
11..12*21.1122*213**312*1
11..1*4*2.1*332113*32*211
*1..113*2.12**1.1*21222..
11....111..12211221.1*111
..111111..111.13*2.12211*
..1*11*2112*2.2**422*1.11
..111112*12*2.2*6**311111
.....11211111.12***2..1*1
.....1*2211...125*41..111
....123*4*31112*4*2111122
....1*22***11*22*211*11**
PS C:\Users\hohuu\Desktop\New folder\net-centric-lab\Lab1\exercise-4> 
```

Hồ Hữu Hiệp - ITITIU20202
Net-Centric Programming – Lab 1

5. Matching Brackets



```go
}

func (s *Stack) Peek() interface{} {
    if len(*s) == 0 {
        return nil
    }
    return (*s)[len(*s)-1]
}

func matchingBracketsCheck(text string) string {
    stack := make(Stack, 0)
    for i := 0; i < len(text); i++ {
        char := string(text[i])
        if char == "{" || char == "(" || char == "[" {
            stack.Push(char)
        } else {
            if char == "}" || char == "]" || char == ")" {
                openBracket := stack.Pop()
                fmt.Print(openBracket)
                if (char == "}" && openBracket != "{") || (char == ")" && openBracket !=
                    return "INVALID BRACKET PLACING!"
                }
            }
        }
    }
    return "ALL BRACKETS MATCHING"
}

func main() {
    fmt.Println("fmt.Println(a.TypeOf(xyz)){[ ]}: ", matchingBracketsCheck("fmt.Println(
    fmt.Println("io[({12hro12b:})]", matchingBracketsCheck("io[({12hro12b:})]"))

    fmt.Println("fmt.Println(a.TypeOf(xyz)){[ }: ", matchingBracketsCheck("fmt.Println(a
    fmt.Println("io[({12hro12b:}]", matchingBracketsCheck("io[({12hro12b:}]"))

}
```

```
PS C:\Users\hohuu\Desktop\New folder\net-centric-lab\Lab1\exer
cise-5> go run .
(([{fmt.Println(a.TypeOf(xyz)){[ ]}:  ALL BRACKETS MATCHING
{([io[({12hro12b:})] ALL BRACKETS MATCHING
(([fmt.Println(a.TypeOf(xyz)){[ }:  INVALID BRACKET PLACING!
{(io[({12hro12b:}] INVALID BRACKET PLACING!
PS C:\Users\hohuu\Desktop\New folder\net-centric-lab\Lab1\exer
cise-5>
```