Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе №2»**

“Основы теории чисел и их использование в криптографии”

**Выполнил:** студент 3 курса

4 группы специальности ПОИТ

Мозолевский Александр Дмитриевич

**Проверил:** преподаватель

Сазонова Дарья Владимировна

Минск 2023

**На основе соотношений (2.1) и (2.2); k = 7**

Реализация на языке GO:

|  |
| --- |
| package mdr  import "strings"  var alphabet = [...]rune{'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'Ä', 'Ö', 'Ü', 'ß'}  func Encrypt(text string, key int) string {  var result strings.Builder  for \_, char := range text {  index := indexOf(char, alphabet)  if index >= 0 {  encryptedIndex := (index + key) % len(alphabet)  result.WriteRune(alphabet[encryptedIndex])  } else {  result.WriteRune(char)  }  }  return result.String()  }  func Decrypt(text string, key int) string {  var result strings.Builder  for \_, char := range text {  index := indexOf(char, alphabet)  if index >= 0 {  decryptedIndex := (index - key + len(alphabet)) % len(alphabet)  result.WriteRune(alphabet[decryptedIndex])  } else {  result.WriteRune(char)  }  }  return result.String()  }  func indexOf(char rune, arr [30]rune) int {  for i, c := range arr {  if c == char {  return i  }  }  return -1  } |

**Таблица Трисемуса, ключевое слово – enigma**

|  |
| --- |
| package trisemus  import "strings"  var trisemusTable = map[rune]rune{  'A': 'E', 'B': 'N', 'C': 'I', 'D': 'G', 'E': 'M', 'F': 'A', 'G': 'B', 'H': 'C',  'I': 'D', 'J': 'F', 'K': 'H', 'L': 'K', 'M': 'L', 'N': 'O', 'O': 'P', 'P': 'Q',  'Q': 'R', 'R': 'S', 'S': 'T', 'T': 'U', 'U': 'V', 'V': 'W', 'W': 'X', 'X': 'Y',  'Y': 'Z', 'Z': 'F', 'Ä': 'Ä', 'Ö': 'Ö', 'Ü': 'Ü', 'ß': 'ß',  }  func Encrypt(text string) string {  var result strings.Builder  for \_, char := range text {  encrypted, ok := trisemusTable[char]  if ok {  result.WriteRune(encrypted)  } else {  result.WriteRune(char)  }  }  return result.String()  }  func Decrypt(text string) string {  var result strings.Builder  var found bool  for \_, char := range text {  found = false  for key, value := range trisemusTable {  if char == value {  result.WriteRune(key)  found = true  break  }  }  if !found {  result.WriteRune(char)  }  }  return result.String()  } |

Результат выполнения:









