Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №1
по курсу «СиМЗИИС»
на тему: «Генерация паролей»

Выполнил студент группы 121703: Якимович И.В.

Проверил: Захаров В.В.

МИНСК, 2023

Задание

1) Разработать программу, реализующую следующие функции:
— генерация строки с заданной пользователем длиной, состоящей из символов алфавита в соответствии с вариантом задания
– поверка равномерности распределения символов путем визуализации частотного распределения;
– вычисление среднего времени подбора пароля, выбираемого из сгенерированной строки.
2) Построить график зависимости среднего времени подбора пароля от его длины.
3) Дать практические рекомендации по выбору пароля исходя из предположений об алфавите пароля; ценности информации, доступ к которой защищается с помощью этого пароля; производительности вычислительного средства атакующего и времени атаки.
Варианты алфавита для генерации пароля:
4) Буквы русского языка строчные и прописные.

```
const ALPHABET_RUS_UPPER = 'АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'
const ALPHABET_RUS_LOWER = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'

function generatePassword(length) {
  const minElement = 1040
  const maxElement = 1103

  let result = ''
  for (var i = 0; i < length; i++) {
    let randomIndex =
      Math.floor(Math.random() * (maxElement - minElement + 1)) + minElement
    result += String.fromCharCode(randomIndex)
  }
  return result
}

function passwordSelection(correctPassword, alphabetFull, length) {
  const symbolsRegister = new Array(length).fill(0)

  while (true) {
    const currentPassword = formString(symbolsRegister, alphabetFull)

    if (currentPassword === correctPassword) {
      console.log('Пароль подобран: ' + currentPassword)
      break
    }
    increment(symbolsRegister, alphabetFull.length)
  }
}

function increment(symbolsRegister, alphabetLength) {
  for (let i = symbolsRegister.length - 1; i >= 0; i--) {
    symbolsRegister[i]++

    if (symbolsRegister[i] === alphabetLength) {
      symbolsRegister[i] = 0
    } else {
      break
    }
  }
}

function formString(symbolsRegister, alphabetFull) {
  let password = ''
  for (const index of symbolsRegister) {
    password += alphabetFull[index]
  }
  return password
}
```

```
function visualizeFrequencyDistribution(password) {
  const frequencyMap = new Map()

  for (let i = 0; i < password.length; i++) {
    const char = password.charAt(i)
    if (frequencyMap.has(char)) {
      frequencyMap.set(char, frequencyMap.get(char) + 1)
    } else {
      frequencyMap.set(char, 1)
    }
  }
  const sortedMap = new Map(
    [...frequencyMap].sort(([keyA], [keyB]) => {
      const asciiA = keyA.charCodeAt(0)
      const asciiB = keyB.charCodeAt(0)
      return asciiA - asciiB
    })
  )

  console.log('Частотное распределение символов:')
  for (const [char, frequency] of sortedMap) {
    const percentage = (frequency / password.length) * 100
    console.log(` ${char}: ${percentage.toFixed(2)}%`)
  }
}

function main() {
  const passLength = 4
  console.log('Русские строчные и заглавные буквы\n')

  let alphabetFull = ALPHABET_RUS_UPPER + ALPHABET_RUS_LOWER

  const password = generatePassword(passLength)
  console.log(`Пароль длиной ${passLength} сгенерирован: ${password}`)

  console.log('_____Подбор пароля_____')

  const start_time = new Date().getTime()
  passwordSelection(password, alphabetFull, passLength)
  const end_time = new Date().getTime()
  const time = (end_time - start_time) / 1000.0

  console.log('Time: ' + time + ' sec')
  console.log(
    'Количество возможных комбинаций ' +
      Math.pow(alphabetFull.length, passLength)
  )

  console.log('_____Распределение символов_____')
  const secondPass = generatePassword(10000000)
  visualizeFrequencyDistribution(secondPass)
}

main()
```

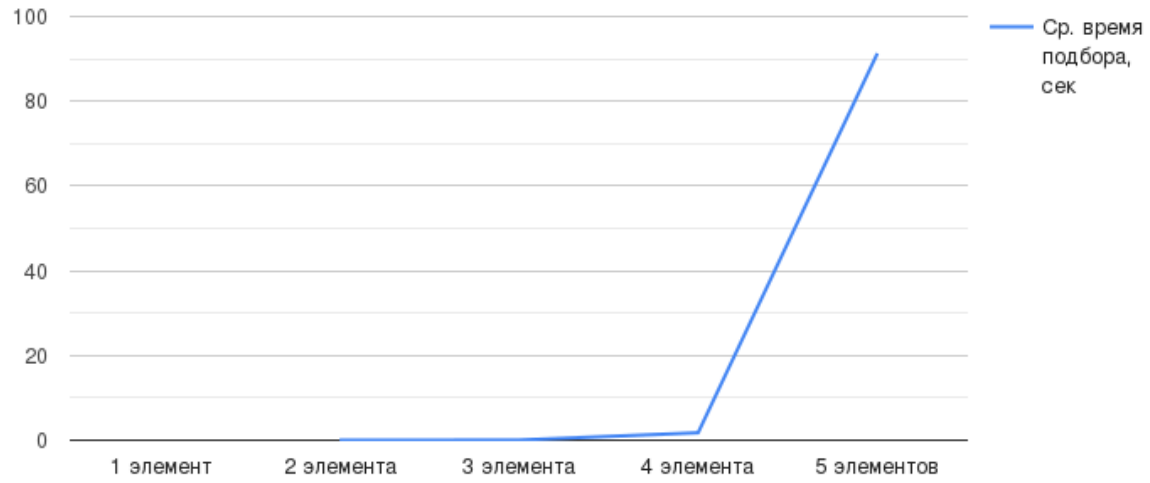
Подбор пароля(пример)

```
Пароль длиной 4 сгенерирован: рюэЬ
_____Подбор пароля_____
Пароль подобран: рюэЬ
Time: 2.735 sec
Количество возможных комбинаций 16777216
```

	1 попытка	2 попытка	3 попытка	4 попытка	5 попытка	Среднее время	Кол. Комбинаций
1 элемент	с <0.001с	О <0.001с	Э <0.001с	й <0.001с	ф <0.001с	<0.001с	64
2 элемента	ьЗ 0.006с	гч 0.003с	иГ 0.002с	чТ 0.013с	ХЙ 0.003с	0.005с	4096
3 элемента	ЦьЩ 0.044с	кгД 0.036с	Гыр 0.017с	оЛУ 0.046с	оаХ 0.056с	0.04с	262144
4 элемента	эЩек 3.893с	бЗЕЮ 1.962с	ЖжПб 0.226с	пейЪ 2.036с	ИышЧ 0.688с	1.761с	16777216
5 элементов	гЫьПн 89.171с	юьтвХ 137.293с	бБЭьй 79.253с	ЕРфУП 17.396с	шУЫбя 133.394с	91.301с	1073741824

Дальше рассчитаем вручную

6 элементов	1.62 ч	68719476736
7 элементов	4,32 дня	4398046511104
8 элементов	276 дней	281474976710656
9 элементов	48 лет	18014398509481984
10 элементов	3072 года	1152921504606847000



Вывод: Время, необходимое для подбора пароля, экспоненциально зависит от размера алфавита и длины пароля. Пароль, состоящий только из последней буквы алфавита, будет подбираться дольше всего, в то время как пароль, состоящий из первой буквы, будет подобран быстрее.

Если представить, что взламывать будут на очень мощном компьютере, способном проверять 1 миллиард (10^9) комбинаций в секунду, то времени на взлом пароля из 8 символов понадобится 2 часа, поэтому стоит создавать пароль длиннее 12 символов, ведь даже на очень мощном компьютере он будет взламываться больше 2000 лет.

Вот несколько рекомендаций для создания надежных паролей:

1. Нужно использовать пароли, состоящие из минимум 12 символов.
2. Использовать разные типы символов, такие как прописные и заглавные буквы, цифры и специальные символы (например, !, @, #, $).
3. Избегать использования персональной информации, такой как имена, даты рождения или номера телефонов.
4. Использовать уникальные пароли для каждой учетной записи.