Sprawozdanie z zadania:
"Szyfr Cezara"

16 października 2024

Autorzy:

- Maria Małasiewicz (badanie efektywność szyfru oraz język koreański),
- Maciej Pestka (przygotowanie tekstu do szyfrowania oraz szyfr Cezara),
- Zuzanna Strauss (funkcja odkodowująca i testy)

1 Wymagania podstawowe

1. usuń zbędne znaki,
2. zaimplementuj szyfr Cezara (dla n liter, w zależności od wybranego alfabetu),
3. zbadaj, jak różne języki (np. polski, angielski, francuski) wpływają na efektywność szyfru Cezara (ze względu na różne częstotliwości liter).

2 Wymagania dodatkowe

4. zaproponuj, jak można dostosować szyfr Cezara do języków, które nie używają alfabetu łacińskiego.

3 Lista spełnionych wymagań:

1, 2, 3, 4

Repozytorium z kodem

4 Opis realizacji zadania:

1. **Usuń zbędne znaki:** Kod za pomocą funkcji usuwa zbędnie znaki. Użytkownik może zdecydować czy chce usunąć polskie/niemieckie litery także może wybrać opcję, by z tekstu usunąć liczby. Jeśli użytkownik zdecyduje się na usunięcie dodatkowych liter albo liczb, to główna funkcja wywołuje funkcje, która usuwaj dodatkowe litery albo liczby. Użytkownik może także zdecydować czy mają być duże litery, jeśli chce usunąć z tekstu duże litery to zostanie uruchomiona funkcja, która zamienia litery na małe litery. Każda funkcja biegnie od pierwszego znaku aż do ostatniego znaku i szuka znaków, które ma usunąć lub zastąpić (mowa zwłaszcza o polskich/niemieckich literach, które zostaną zamieniony na odpowiednik angielskiej litery). Następnie dzieli tekst na pięć liter i na kolumny, które potem program zaszyfruję.

2. **Zaimplementuj szyfr Cezara:** Funkcja, która służy do zaszyfrowania tekstu, tak samo biegnie jak pozostałe funkcję od pierwszego znaku do ostatniego znaku (już na teksie, który jest przygotowany do szyfrowania). W pętli znajdują się warunki, które mają wykryć czy w teksie jest mała/duża litera, cyfra oraz sylaba koreańska. Jeśli wykryję małą/duża literę to sprawdza funkcja jaki język użytkownik wybrał, by wyłapać dodatkowe znaki do zakodowania. Polskie oraz niemieckie litery nie zostały dodanie do głównego alfabetu danego państwa, tylko jako dodatkowe znaki, które są szyfrowanie względem klucza, są podobnie szyfrowanie jako liczby.

2.5 **Odkodowanie szyfru:** W celu sprawdzenia poprawności działania kodu szyfrującego, został napisany kod który odszyfrowuje tekst znając użyty klucz. Składa się on z funkcji wczytującej plik z zaszyfrowanym tekstem, zapisującej odszyfrowany tekst, usuwającej znaki specjalne, funkcji porównującej zgodność oryginalnego tekstu z odszyfrowanym oraz samej funkcji dekodującej. Funkcja dekodująca zawiera podane alfabety języków, które mogą być zaszyfrowane. Wczytywany jest plik z szyfrem i podajemy klucz. Następnie pętla iteruje po każdej literze w tekście i sprawdza do którego alfabetu należy. Od indeksu litery w alfabecie odejmowany jest klucz, a wynik dzielony modulo przez długość alfabetu. Do odszyfrowanego tekstu dodana jest litera o nowym indeksie. Gdy już cały tekst jest już odkodowany, wywoływana jest funkcja porównująca tekst oryginalny z odkodowanym. Funkcja ta wczytuje tekst oryginalny i w razie potrzeby usuwa z niego pierwszą linie z instrukcją kodowania. Porównywane są ze sobą kolejne litery z obu tekstów i zliczana jest ilość znaków zgodnych. Na końcu pokazywany jest wynik zgodności.

3. **Zbadaj jak różne języki wpływają na efektywność szyfru:** W celu realizacji zadania napisano kod próbujący odnaleźć klucz zaszyfrowanego tekstu. Została napisana funkcja zliczająca częstotliwość występowania poszczególnych liter w zaszyfrowanym tekście. Następnie te dane przyrównywane są z częstotliwością występowania liter dla języka, w którym oryginalny tekst był napisany.

4. **Dostosowanie szyfru Cezara do języków nieużywających alfabetu łacińskiego:** Przeprowadzono research i na podstawie źródeł w różnych językach na temat szyfru Cezara (wikipedia.org) zauważono, że zachowuje się on podobnie dla różnych alfabetów, szczególnie alfabetów niesylabicznych. Podjęto się napisania kodu do szyfrowania tekstów napisane alfabetem koreańskim (hangul). Alfabet składa się z ponad 11000 sylab (zwanych literami) i to one zostają zastępowane o sylabę odsuniętą o dany klucz. Został rozważony warunek, gdy w tekście występują skróty w formie pojedynczych znaków.

5 Prawa autorskie do kodu:

Prowadzący udostępnił fragmenty kodu związanego z przygotowaniem tekstem do kodowania.

6 Wnioski:

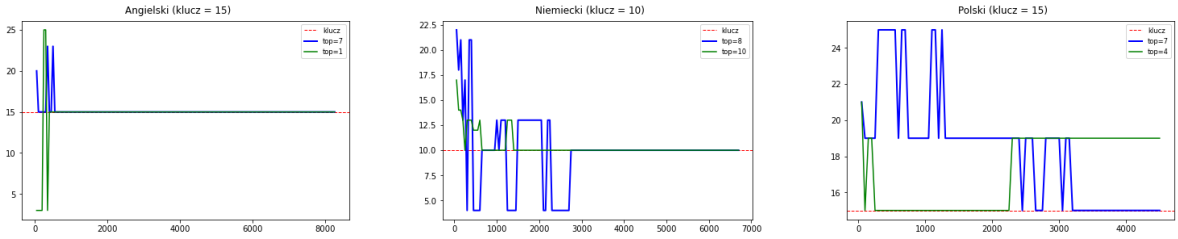
- Dodanie polskich lub niemieckich liter do alfabetu (nie jako dodatkowe znaki), mogłoby spowodować, że kod programu byłby bardziej nie czytelny oraz osoba, która by próbująca odszyfrować ten kod musiałaby więcej czasu to poświecić. Dodanie polskich/niemieckich znaków skróciło zapis kodu. Oraz osoba pisząca program musiałaby więcej czasu poświecić by to dobrze zakodować.

- W szyfrowaniu nie jest uwzględniony czy użytkownik chce uwzględniać polskie/niemiecki litery, ponieważ do warunków musiałbym dodać kolejne warunki. W warunki jest tylko język

uwzględniony, jak funkcję usuwa polskie/niemieckie litery to automatyczne tamten warunek i tak się nie wykona.

- Podmianka scharfes S ("ß") na zwykła mała literę b, powoduję że szyfr jest mniejsza szansa na odszyfrowanie tekstu. Zazwyczaj 'ß' zamieniane jest na "ss", jednak zmienia to częstotliwość występowania liter, a zwłaszcza par liter ułatwiając łamanie szyfru .

- Złamanie szyfru Cezara jest stosunkowo łatwe, szczególnie porównując najczęściej występującą literę w zaszyfrowanym tekście z najczęściej występującą literą w danym języku.
- Trudność pojawia się w napisaniu kodu łamiącego szyfr, i wyłącznie przez komputer bez wykorzystania dużych bibliotek zwierających słowniki i wyszukujących istniejących słów. Pojawia się pytanie jak sprawdzić czy wyliczony klucz jest poprawny bez udziału człowieka?
  - Spróbowano wybierać $n < 26$ najczęściej występujących liter i sprawdzać czy zgadza się dla nich wyliczony klucz. Jednak różnice w kolejności liter mogą różnic się pojedynczymi wystąpieniami i powodować niezgodności.
  - Problem ten spróbowano naprawić posortowaniem list (o długości n), a następnie szukając odpowiedniego przesunięcia. Do analizy skuteczności przygotowano wykresy, które pokazują jak długość tekstu (oraz wartość n) wpływa na wyliczany klucz.



- Ciekawym problemem było kodowanie języka koreańskiego. Hangul składa się z 51 liter ('jamo'), z czego 19 to spółgłoski a 32 to samogłoski. Z nich budowane są sylaby, które są zapisywane w blokach od dwóch do czterech liter. String zawierający jeden blok ma długość 1. Zatem pojawia się pytanie jak szyfrować tekst napisany hangul?
  - Zaproponowaliśmy zastępowanie całych sylab, jednak jako że sylaba składa się z wielu liter, więc aby zmienić pierwszą literę sylaby jest wymagany klucz o dużej wartości. Jednak zastępowanie całych sylab nie zaszyfruje nam skrótów z pojedynczych liter.
  - Rozkładanie sylab na pojedyncze litery? Mogłoby się to wydawać rozwiązaniem, jednak wymagałoby to użycia dodatkowej biblioteki do rozkładania sylab oraz pojawiałby się problem ze strukturą tekstu zaszyfrowanego (nie wszystkie podmienione litery z sylaby mogą zbudować nową sylabę). W rezultacie mogłoby to utrudniać czytelność odszyfrowanego tekstu przez jego odbiorcę.

7 Źródła

- wikipedia.org:
  - szyfr Cezara (źródła także w innych językach)
  - Hangul
- Frekwencja liter w polskich tekstach; PWN
- The frequency of the letters of the alphabet in English
- Letter Frequency by Language