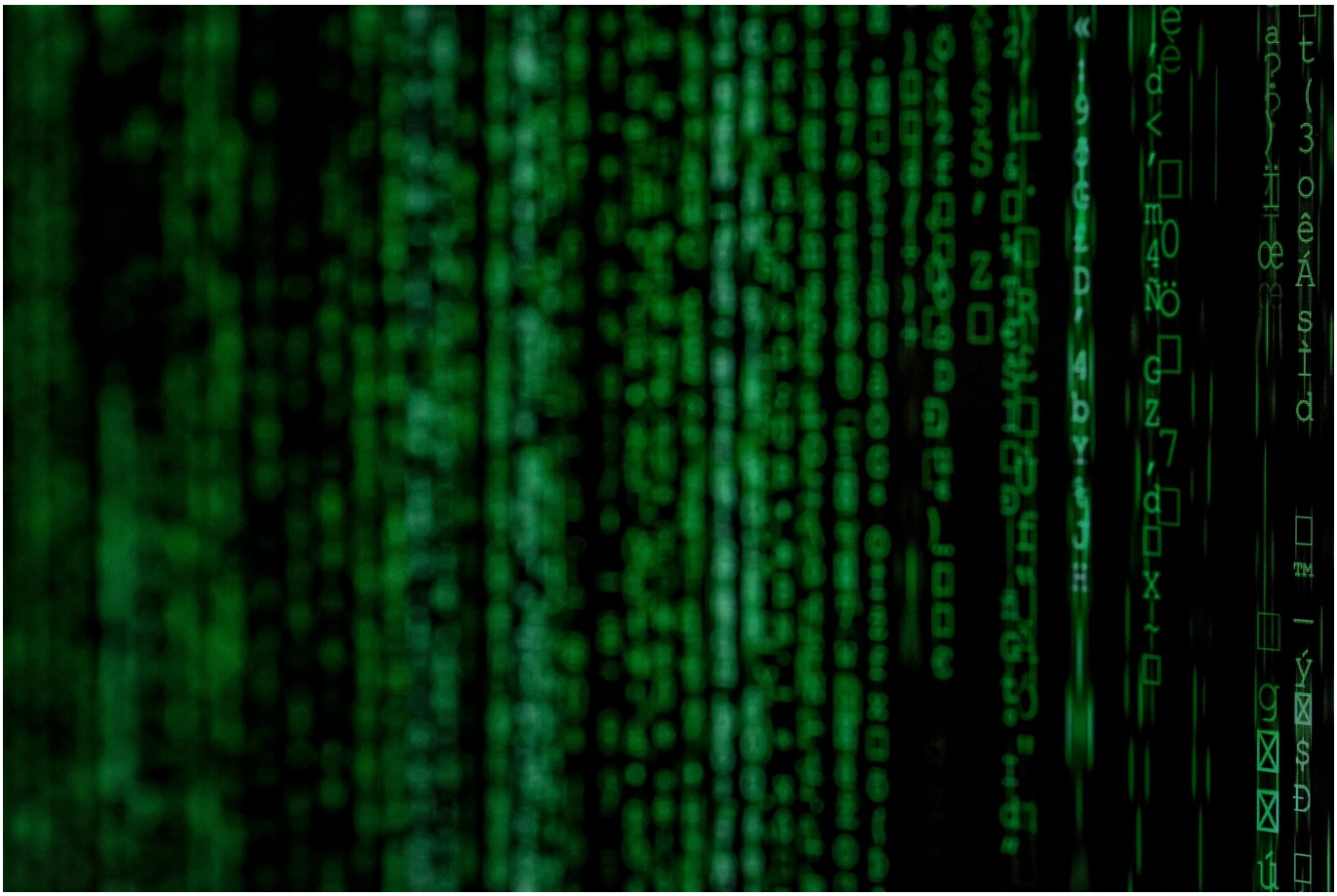
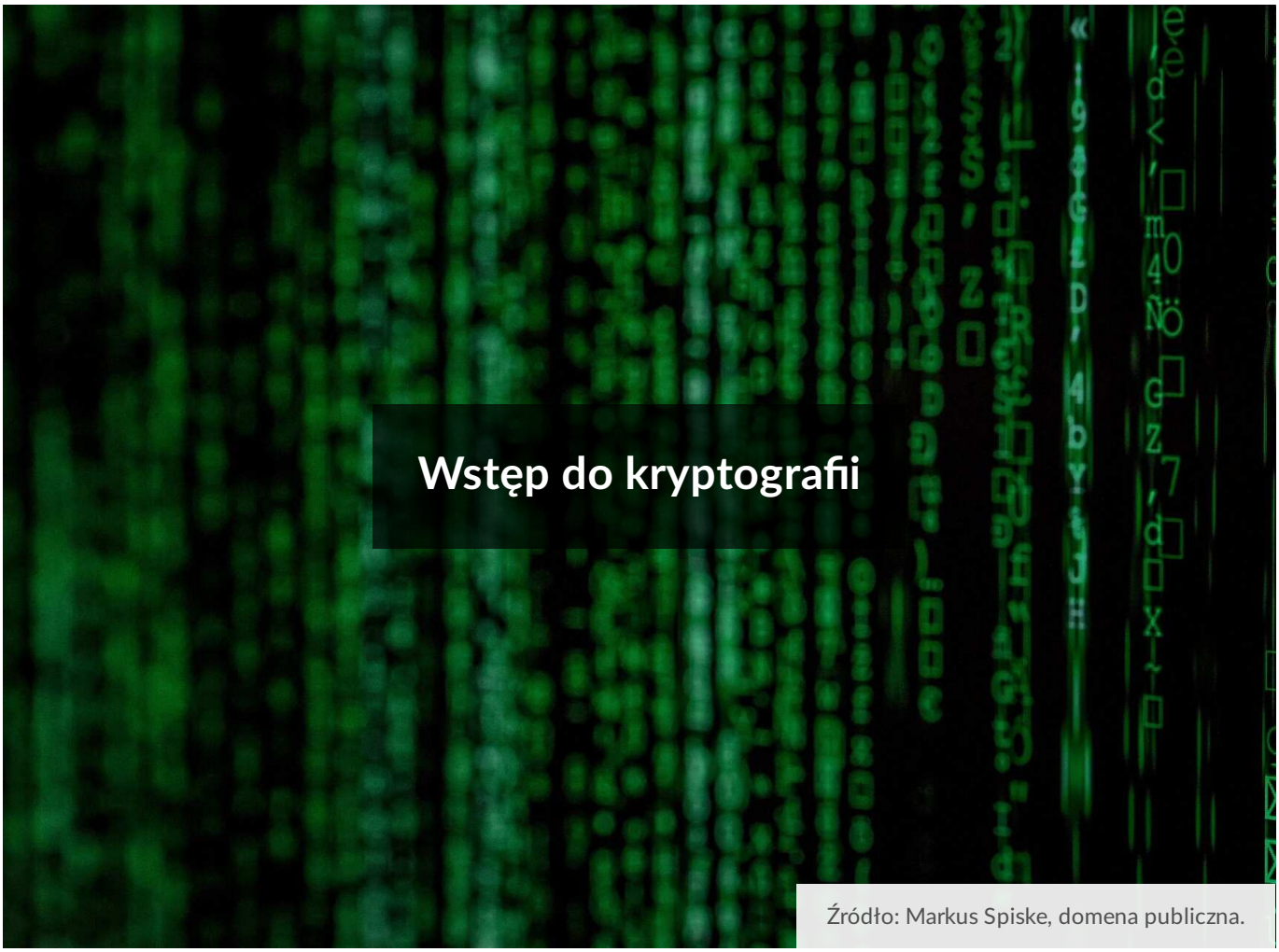


Wstęp do kryptografii

- Wprowadzenie
- Przeczytaj
- Animacja
- Sprawdź się
- Dla nauczyciela



Źródło: Markus Spiske, domena publiczna.

Wiesz już, czym jest szyfrowanie, ale czy znasz konkretne metody szyfrowania?

W tym e-materiale poznasz przykładowy algorytm szyfrowania symetrycznego – szyfrowanie płotkowe.

Implementacje algorytmu w wybranych językach programowania przedstawiamy w e-materiałach:

- Wstęp do kryptografii w języku C++,
- Wstęp do kryptografii w języku Java,
- Wstęp do kryptografii w języku Python.

Więcej zadań? Sięgnij do: Wstęp do kryptografii – zadania maturalne.

Twoje cele

- Wymienisz różne rodzaje szyfrów.
- Przeanalizujesz terminologię stosowaną w kryptografii.
- Zapiszesz algorytm szyfrujący podaną wiadomość za pomocą szyfru płotkowego.

Przeczytaj

Kryptografia

Ludzie od wieków stosowali rozmaite metody, by chronić ważne, poufne informacje. Spartanom przypisuje się autorstwo pierwszego narzędzia szyfrującego o nazwie **skytale** – był to skórzany lub pergaminowy pasek, na którym znajdowały się litery, odczytanie wiadomości było możliwe po nawinięciu go na laskę takiej samej grubości, jaką miał autor wiadomości. W trakcie II wojny światowej Niemcy wykorzystywali urządzenie szyfrujące **Enigma**, a Polacy wsławili się dzięki rozszyfrowaniu wiadomości wojskowych, które ta maszyna szyfrowała. Do szyfrowania nie zawsze konieczne są specjalne urządzenia, czasami wystarczy kartka i ołówek.

Rozmawiając ze znajomymi, korzystasz z różnych komunikatorów. Czasami zdarza się, że chcesz przekazać jakąś poufną informację. Zależy ci, by nikt poza odbiorcą jej nie przeczytał. Z pomocą może przyjść kryptografia.

Istnieje wiele algorytmów szyfrujących, które można wykorzystać. Najczęściej stosowane algorytmy polegają na specyficznym przekształceniu wiadomości za pomocą klucza.

Szyfry możemy podzielić na kilka rodzajów:

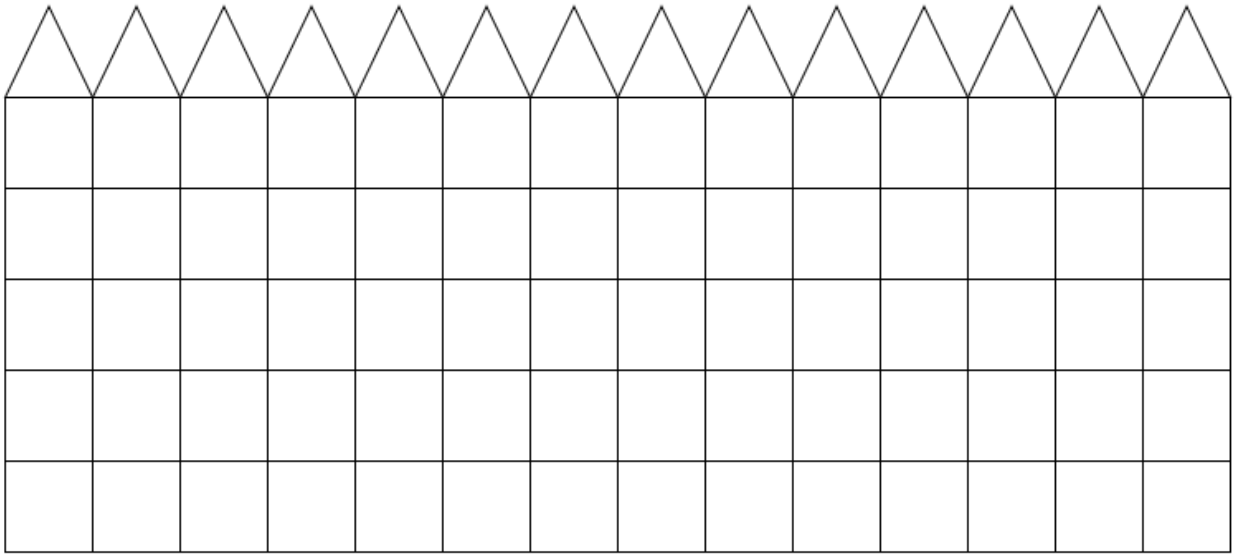
- **szyfr podstawieniowy** – każdy znak szyfrowanej wiadomości zostaje zastąpiony innym; deszyfrowanie polega na odwróceniu podstawienia; przykładem takiego szyfru jest szyfr Cezara;
- **szyfr przestawieniowy** – w zaszyfrowanej wiadomości obecne są wszystkie znaki użyte w tekście jawnym, zmieniona zostaje jedynie ich pozycja w szyfrogramie; przykładem takiego szyfru jest szyfr płotkowy;
- **szyfr symetryczny** – wiadomość jest szyfrowana za pomocą tajnego klucza, który następnie służy do odczytania wiadomości;
- **szyfr asymetryczny** – wiadomość zostaje zaszyfrowana oraz deszyfrowana różnymi kluczami; klucz używany w procesie szyfrowania jest publiczny, natomiast drugi klucz, wykorzystywany podczas deszyfrowania, jest prywatny; przykładem takiego szyfru jest szyfr RSA.

W dalszej części materiału poznasz przykład szyfru przedstawieniowego – szyfr płotkowy.

Szyfr płotkowy – opis algorytmu szyfrowania

Chcesz przekazać przyjacielowi tajne miejsce waszego spotkania – punktem zbiórki jest fontanna w parku.

Aby zaszyfrować wiadomość za pomocą szyfru płotkowego, na początku należy wybrać klucz symetryczny, za pomocą którego zaszyfrujemy wiadomość. W przypadku tego algorytmu kluczem będzie wysokość płotku, czyli w podanym przykładzie: pięć kwadratów.

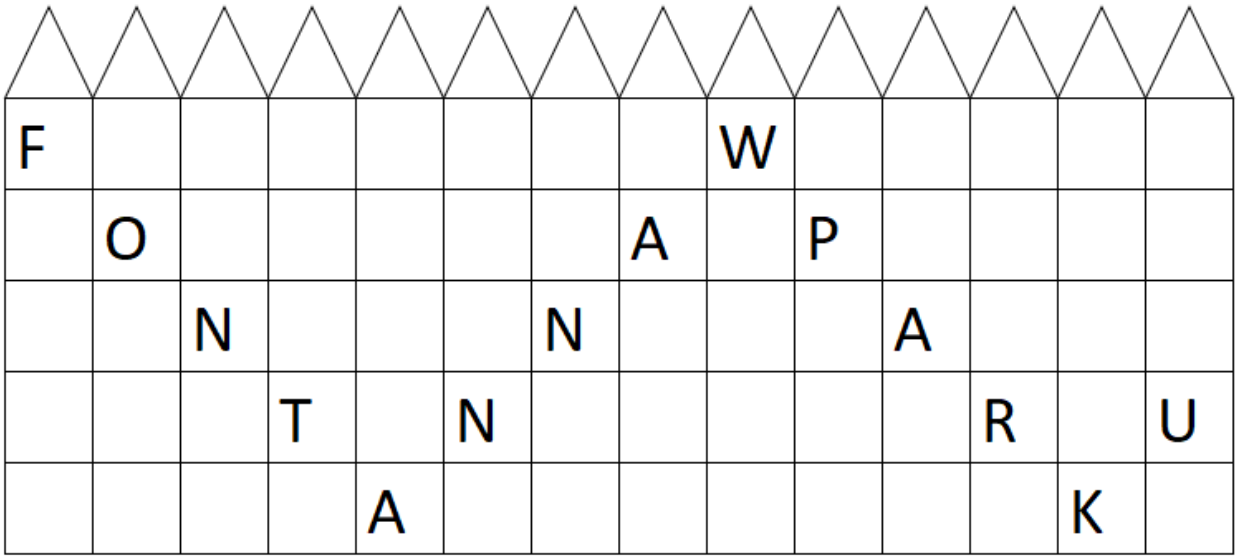


Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Pierwszym krokiem będzie właściwe przygotowanie wiadomości. Należy zmienić wielkość wszystkich liter na duże (np. a na A) oraz usunąć spacje. Zmiana wielkości liter utrudni odczytanie szyfrogramu przez niepowołaną osobę. W efekcie otrzymamy następujący tekst:

FONTANNAWPARKU

Następnie należy wypełnić pola w płotku literami tekstu jawnego, począwszy od lewej górnej kratki schematu. W każdym kolejnym kroku następny znak szyfrowanego tekstu wpisujemy do pola diagramu ulokowanego o jedną pozycję na prawo oraz w dół w stosunku do poprzedniej kratki. W momencie, w którym osiągniemy pole znajdujące się w najniższym rzędzie płotku, zmieniamy kierunek wyboru kolejnych kratek – każde następne pole będzie zlokalizowane o jedną pozycję na prawo oraz do góry w stosunku do poprzedniej kratki. Gdy dotrzemy do pozycji w rzędzie najwyższym, ponownie zmieniamy kierunek. Proces ten powtarzamy, aż do wpisania w pola płotku wszystkich liter tekstu jawnego. Utworzony przez znaki kształt powinien przypominać zygzak.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Kolejne litery odczytujemy w wierszach, zaczynając zawsze od lewej strony. W ten sposób otrzymamy następujący zaszyfrowany tekst:

FWOAPNNATNRUAK

Aby twój przyjaciel mógł odczytać informację, należy przekazać mu zaszyfrowaną wiadomość wraz z kluczem symetrycznym.

Pseudokod algorytmu szyfrowania płotkowego

Specyfikacja problemu:

Dane:

- `n` – liczba naturalna; liczba znaków w tekście jawnym
- `tekst[0..n - 1]` – przekazany do zaszyfrowania szyfrem płotkowym tekst jawny; tablica znaków zawierająca wielkie i małe litery alfabetu angielskiego oraz spacje
- `klucz` – liczba rzędów, z których składa się wykorzystywany w szyfrowaniu schemat płotku; liczba naturalna dodatnia.

Wynik:

- `szyfrogram[0..m - 1]` – zaszyfrowany szyfrem płotkowym tekst jawny; tablica znaków zawierająca wielkie litery alfabetu angielskiego; gdzie `m` jest równe `n` minus liczba spacji w tekście jawnym

> **Ważne!**

> W wykonaniu opisanych w algorytmie operacji przydatne będzie operowanie na kodach ASCII, pozwalających reprezentować wybrane znaki, np. litery, cyfry czy znaki interpunkcyjne w postaci liczb całkowitych z zakresu od 0 do 127.
>
> W pseudokodzie wykorzystamy trzy dodatkowe funkcje:
>
> - `znakDoKodu(znak)` – funkcja, która jako parametr przyjmuje pojedynczy znak, a następnie konwertuje go na odpowiadający mu kod ASCII i zwraca otrzymaną wartość liczbową.
> - `kodDoZnaku(kod)` – funkcja, która jako parametr przyjmuje kod ASCII w postaci liczby całkowitej z zakresu od 0 do 127, a następnie konwertuje ją na odpowiadający jej znak oraz go zwraca.
> - `rozmiar(tablica)` – funkcja, która jako parametr przyjmuje tablicę, a następnie podaje liczbę jej elementów.

Do funkcji szyfrującej przekazywany jest podany przez użytkownika tekst jawny, liczba znaków w nim zawarta oraz klucz. Postępujemy według przedstawionego schematu.

Krok 1. Usuń spacje i zmień małe litery na wielkie, jeżeli jest to konieczne.

W funkcji początkowo definiujemy tablicę `jawny[]` o długości `n`, która docelowo przechowywać będzie tekst jawny składający się jedynie z wielkich liter alfabetu angielskiego (po usunięciu spacji oraz zmianie małych liter na wielkie). Dodatkowo przydatne będzie utworzenie zmiennej `a` wskazującej na indeks elementu tablicy `jawny[]`, do którego zapisywać będziemy znak tekstu jawnego. Inicjalizujemy ją wartością 1.

```
funkcja szyfrPlotkowy(tekst[0..n - 1], klucz)
    jawny[0..n - 1] ← wprowadzamy wiadomość do zaszyfrowania
    a ← 1
```

Każdy element tablicy `tekst[]` porównujemy ze znakiem „ „ (spacja). Jeżeli dany element nie jest spacją, to weryfikujemy, czy stanowi małą literę alfabetu poprzez sprawdzenie, czy jego kod ASCII jest większy lub równy kodowi litery „a”. Zasadność takiego porównania wynika z faktu, że małe litery w kodzie ASCII mają wartości od 97 do 122, a wielkie od 65 do 90. Jeżeli sprawdzany znak okaże się małą literą, pomniejszamy jego kod ASCII o 32 i zapisujemy do tablicy `tekst[]`. Jest to stała różnica pomiędzy kodami tej samej litery w wariancie wielkim i małym, np. „a” (97) oraz „A” (65), czyli 97 - 65 = 32. W ten sposób małe litery zostają zamienione na wielkie. **Niezależnie**, czy przeprowadzona została operacja zmiany wielkości litery, sprawdzany znak (o ile nie jest spacją) zapisujemy do elementu tablicy `jawny[]` o indeksie `a`. Na koniec inkrementujemy wartość zmiennej `a`.

```
funkcja szyfrPlotkowy(tekst[0..n - 1], klucz)
```

```
    jawny[0..n - 1] ← wprowadzamy wiadomość do zaszyfrowania
    a ← 1

    dla i = 0, 1, ..., n - 1 wykonuj:
        jeżeli tekst[i] ≠ ' ':
            jeżeli znakDoKodu(tekst[i]) >= znakDoKodu('a') wykona
                tekst[i] ← kodDoZnaku(znakDoKodu(tekst[i]) - 32)
            jawny[a] ← tekst[i]
            a ← a + 1
```

Krok 2. Określ liczbę znaków w tekście jawnym po przeprowadzonych operacjach usuwania spacji oraz zmiany wielkości liter.

Długość tablicy `jawny[]` początkowo określiliśmy jako `n`, ponieważ jest to maksymalna liczba znaków, które może zawierać tekst jawny po usunięciu spacji (w przypadku gdy pierwotnie nie znajdowały się w nim żadne spacje). Może jednak dojść do sytuacji, że długość tekstu – po wykonaniu opisanych operacji – ulegnie zmianie. Aby określić liczbę liter w tablicy `jawny[]`, sprawdzimy, ile z jej elementów posiada kod ASCII pomiędzy 65 (kod litery „A") a 90 (kod litery „Z"). Poszukiwaną długość tekstu jawnego, po modyfikacjach, przechowywać będziemy w zmiennej `m`.

```
m ← 0
dla i = 0, 1, ..., n - 1 wykonuj:
    jeżeli znakDoKodu(jawny[i]) >= znakDoKodu('A') i znakDoKodu(j
        m ← m + 1
```

Krok 3. Umieść tekst jawny w tablicy dwuwymiarowej.

Tworzymy tablicę `plotek[][]` zawierającą liczbę wierszy równą kluczowi oraz składającą się z liczby kolumn równej wartości zmiennej `m`. Wypełniamy ją znakami spacji. Następnie uzupełniamy ją kolejnymi znakami tekstu jawnego w formie opisanego w szyfrze płotkowym zygzaka. Pomoże nam w tym flaga `dol`, która będzie zmieniać swoją wartość na 1 (gdy tekst będzie iść w dół) lub 0 (gdy tekst będzie iść do góry) – w zależności od tego, czy osiągnięta zostanie minimalna lub maksymalna wysokość tablicy `plotek[][]`.

```
plotek[0..klucz - 1][0..m - 1]
wiersz ← 1
dol ← 1

dla i = 0, 1, ..., klucz - 1 wykonuj:
```

```
    dla j = 0, 1, ..., m - 1 wykonuj:
        plotek[i][j] ← ' '
i ← 1

dopóki i <= m wykonuj:
    plotek[wiersz][i] ← jawny[i]
    i ← i + 1

    jeżeli wiersz = klucz:
        dol ← 0
    jeżeli wiersz = 1:
        dol ← 1
    jeżeli dol = 1:
        wiersz ← wiersz + 1
    w przeciwnym razie:
        wiersz ← wiersz - 1
```

Krok 4. Odczytaj zaszyfrowaną wiadomość.

Na koniec definiujemy tablicę `szyfrogram` o długości `m`, a następnie zapisujemy do niej litery, którymi wypełniliśmy pola utworzonego płotku (według zasad szyfru płotkowego). Rezultat przedstawionych operacji zwracamy.

```
szyfrogram[0..m - 1]
b ← 1

dla i = 0, 1, ..., klucz - 1 wykonuj:
    dla j = 0, 1, ..., m - 1 wykonuj:
        jeżeli plotek[i][j] ≠ ' ':
            szyfrogram[b] ← plotek[i][j]
            b ← b + 1

zwróć szyfrogram[]
```

Krok 5. Gotowa funkcja zapisana w pseudokodzie.

```
funkcja szyfrPlotkowy(tekst[0..n - 1], klucz)
    jawny[0..n - 1] ← wprowadzamy wiadomość do zaszyfrowania
    a ← 1

```

```
    dla i = 0, 1, ..., n - 1 wykonuj:
        jeżeli tekst[i] ≠ ' ':
            jeżeli znakDoKodu(tekst[i]) >= znakDoKodu('a') wykona
                tekst[i] ← kodDoZnaku(znakDoKodu(tekst[i]) - 32)
            jawny[a] ← tekst[i]
            a ← a + 1

m ← 0
dla i = 0, 1, ..., n - 1 wykonuj:
    jeżeli znakDoKodu(jawny[i]) >= znakDoKodu('A') i znakDoKodu(j
        m ← m + 1

plotek[0..klucz - 1][0..m - 1]
wiersz ← 1
dol ← 1

dla i = 0, 1, ..., klucz - 1 wykonuj:
    dla j = 0, 1, ..., m - 1 wykonuj:
        plotek[i][j] ← ' '
i ← 1

dopóki i <= m wykonuj:
    plotek[wiersz][i] ← jawny[i]
    i ← i + 1

    jeżeli wiersz = klucz:
        dol ← 0
    jeżeli wiersz = 1:
        dol ← 1
    jeżeli dol = 1:
        wiersz ← wiersz + 1
    w przeciwnym razie:
        wiersz ← wiersz - 1

szyfrogram[0..m - 1]
b ← 1

dla i = 0, 1, ..., klucz - 1 wykonuj:
    dla j = 0, 1, ..., m - 1 wykonuj:
        jeżeli plotek[i][j] ≠ ' ':
            szyfrogram[b] ← plotek[i][j]
            b ← b + 1
```

```
47
48 zwróć szyfrogram[]
```

Słownik

iteracja

technika programowania, która polega na powtarzaniu tej samej operacji określoną liczbę razy lub do momentu, w którym zadany warunek zostanie spełniony

klucz

informacja, która jest wykorzystywana do szyfrowania i/lub deszyfrowania wiadomości

klucz prywatny

tajny klucz wykorzystywany w procesie deszyfrowania w szyfrach asymetrycznych; powinien być znany jedynie adresatowi zaszyfrowanej wiadomości

klucz publiczny

udostępniony publicznie klucz wykorzystywany w procesie szyfrowania w szyfrach asymetrycznych

kod ASCII

7-bitowy system kodowania znaków, w którym każdy z obsługiwanych symboli jest reprezentowany przez liczbę; 7 bitów umożliwia przechowanie informacji o znakach o kodach z zakresu 0-127. Używany m.in. we współczesnych komputerach oraz sieciach komputerowych

kryptografia

gałąź wiedzy o zapisywaniu informacji w sposób utrudniający, bądź całkowicie uniemożliwiający jej odczytanie

tablica ASCII

spis kodów znaków wykorzystywany w komputerach

szyfrogram

zaszyfrowana wiadomość

szyfrowanie

przekształcanie tekstu jawnego w szyfrogram

Animacja

Polecenie 1

Dzień po przesłaniu do przyjaciela zaszyfrowanej wiadomości, z opisanym miejscem spotkania, otrzymujesz odpowiedź – również zaszyfrowaną. Zapoznaj się z prezentacją, z której dowiesz się, jak odszyfrować otrzymaną wiadomość.

1

Potrafisz już szyfrować tekst jawny za pomocą szyfru płotkowego. Teraz nauczysz się, jak – znając klucz – deszyfrować wiadomość.

2

Następnego dnia twój kolega – w odpowiedzi – wysłał ci następującą wiadomość "ZIANAPMLMOE". Wiesz, że zaszyfrował ją za pomocą klucza symetrycznego o wartości cztery.

3

Specyfikacja:

Dane:

- `szyfrogram[0..m - 1]` – tekst zaszyfrowanej wiadomości; tablica znaków zawierająca wielkie litery alfabetu angielskiego.
- `klucz` – liczba rzędów, z których składa się wykorzystywany w odszyfrowywaniu schemat płotku; liczba naturalna (nie uwzględniając 0).

- `m` – zmienna przechowująca liczbę znaków w szyfrogramie; liczba naturalna

Wynik:

- `jawny[0..m - 1]` – odszyfrowana wiadomość; tablica znaków zawierająca wielkie litery alfabetu angielskiego

4

Na początku, podobnie jak w przypadku algorytmu szyfrującego, należy utworzyć pustą tablicę dwuwymiarową `plotek[][]`. Będzie ona miała liczbę wierszy równą wartości klucza oraz liczbę kolumn równą długości szyfrogramu.

5

Wypełniamy tablicę znakami spacji.

```
1 funkcja
  deszyfrowanie(szyfrogram,
  klucz, m)
2     plotek[0..klucz - 1]
  [0..m - 1]
3
4     dla i = 0, 1, ...,
  klucz - 1 wykonuj:
5         dla j = 0, 1, ...,
  m - 1 wykonuj:
6             plotek[i][j] ←
  ' '
7
```

6

Uzupełnij tablicę `plotek[][]` o znaki `'*'`, które będą oznaczały miejsce wpisania kolejnego znaku szyfrogramu. Zrób to w taki sam sposób jak podczas szyfrowania, tak aby otrzymany wzór przypominał zygzak.

7

Zmienne `wiersz` i `kolumna` wskazują miejsce, gdzie należy umieścić kolejną literę szyfrogramu. Zmienna `dol` określa kierunek wpisywania znaków. Jeżeli jej wartość wynosi jeden, numer wiersza zwiększa się o jeden. Jeżeli jej wartość wynosi zero, numer wiersza zmniejsza się o jeden.

8

```
wiersz ← 1
dol ← 1
dla kolumna = 0, 1, ..., m - 1 wykonuj:
    plotek[wiersz][kolumna] ← '*'
    jeżeli wiersz = klucz:
        dol ← 0
    jeżeli wiersz = 1:
        dol ← 1

    jeżeli dol = 1:
        wiersz ← wiersz + 1
    w przeciwnym razie:
        wiersz ← wiersz - 1

```

9

Aby w miejsce znaku `'*'` wpisać kolejną literę szyfrogramu, należy utworzyć zmienną `znak`, która będzie przechowywała indeksy kolejnych znaków zawartych w zaszyfrowanej wiadomości. Początkowo będzie ona zawierała wartość 1, gdyż tablicę indeksujemy od 1.

Następnie, za pomocą zagnieżdżonych pętli `dla`, będziemy – znak po znaku – zapisywać kolejne litery zaszyfrowanej wiadomości w tablicy `plotek[][]`, w miejscach, w których wpisany został znak `'*'`

```
1 znak ← 1
2 dla i = 0, 1, ..., klucz -
  1 wykonuj:
3     dla j = 0, 1, ..., m -
  1 wykonuj:
4         jeżeli plotek[i]
  [j] = '*':
5             plotek[i][j] ←
  szyfrogram[znak]
6             znak ← znak +
  1
7
```

10

Podobnie jak w poprzednim fragmencie kodu, zgodnie ze wskazaniami zmiennych `wiersz` i `kolumna`, kolejne znaki tablicy `plotek[][]` zapisywane są do tablicy `jawny[]`.

```
1 jawny[0..m - 1]
2 wiersz ← 1
3 dol ← 1
4 dla kolumna = 0, 1, ..., m
  - 1 wykonuj:
5     jawny[kolumna] ←
  plotek[wiersz][kolumna]
6
7     jeżeli wiersz = klucz:
```

```
        dol ← 0
    jeżeli wiersz = 1:
        dol ← 1

    jeżeli dol = 1:
        wiersz ← wiersz +
1
    w przeciwnym razie:
        wiersz ← wiersz -
1

```

11

Zawartość tablicy `jawny[ ]` jest następująca – "ZAPOMNIALEM". Dzięki temu wiesz, że dzisiejsze spotkanie przy fontannie nie odbędzie się.

12

Gotowy pseudokod:

```
funkcja
deszyfrowanie(szyfrogram,
klucz, m)
    plotek[0..klucz - 1]
[0..m - 1]

    dla i = 0, 1, ...,
klucz - 1 wykonuj:
        dla j = 0, 1, ...,
m - 1 wykonuj:
            plotek[i][j] ←
' '

    wiersz ← 1
    dol ← 1
    dla kolumna = 0, 1,
..., m - 1 wykonuj:
```

```
11          plotek[wiersz]
[kolumna] ← '*'
12
13          jeżeli wiersz =
klucz:
14              dol ← 0
15          jeżeli wiersz = 1:
16              dol ← 1
17
18          jeżeli dol = 1:
19              wiersz ←
wiersz + 1
20          w przeciwnym
razie:
21              wiersz ←
wiersz - 1
22
23      znak ← 1
24      dla i = 0, 1, ...,
klucz - 1 wykonuj:
25          dla j = 0, 1, ...,
m - 1 wykonuj:
26              jeżeli
plotek[i][j] = '*':
27                  plotek[i]
[j] ← szyfrogram[znak]
28                  znak ←
znak + 1
29
30      jawny[0..m - 1]
31      wiersz ← 1
32      dol ← 1
33      dla kolumna = 0, 1,
..., m - 1 wykonuj:
34          jawny[kolumna] ←
plotek[wiersz][kolumna]
35
36          jeżeli wiersz =
klucz:
37              dol ← 0
38          jeżeli wiersz = 1:
39              dol ← 1
40
41          jeżeli dol = 1:
```

```
42              wiersz ←
wiersz + 1
43          w przeciwnym
razie:
44              wiersz ←
wiersz - 1
45
46      zwróć jawny[]
47
```

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

> **Polecenie 2**
>
> Zapoznaj się z animacją. Wymień, jakie inne szyfry znasz. Poszukaj informacji na temat tego, czy są jeszcze stosowane.

Film dostępny pod adresem /preview/resource/R1Se0df0YRBmT

Film nawiązujący do treści materiału: Kryptografia w starożytności.

Ćwiczenie 1

Poszukaj informacji na temat innych szyfrów, które odegrały ważną rolę w historii. Opisz jeden z nich.

Sprawdź się

Pokaż ćwiczenia:

Ćwiczenie 1

Ćwiczenie 2

Ćwiczenie 3

Ćwiczenie 4

Ćwiczenie 5

Ćwiczenie 6

Ćwiczenie 7

Ćwiczenie 8

Dla nauczyciela

Autor: Maurycy Gast
Przedmiot: Informatyka

Temat: Wstęp do kryptografii

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres podstawowy

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

V. Przestrzeganie prawa i zasad bezpieczeństwa. Respektowanie prywatności informacji i ochrony danych, praw własności intelektualnej, etykiety w komunikacji i norm współżycia społecznego, ocena zagrożeń związanych z technologią i ich uwzględnienie dla bezpieczeństwa swojego i innych.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

b) na tekstach: porównywania tekstów, wyszukiwania wzorca w tekście metodą naiwną, szyfrowania tekstu metodą Cezara i przestawieniową,

4) porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji;

5) sprawdza poprawność działania algorytmów dla przykładowych danych.

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Wymienisz różne rodzaje szyfrów.
- Przeanalizujesz terminologię stosowaną w kryptografii.
- Zapiszesz algorytm szyfrujący podaną wiadomość za pomocą szyfru płotkowego.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium i ćwiczeń interaktywnych;
- metody aktywizujące;
- burza mózgów;
- mapa myśli.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda.

Przebieg lekcji

Przed lekcją:

1. Uczniowie przygotowują krótkie prezentacje dotyczące wskazanych przez nauczyciela szyfrów.

2. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Wstęp do kryptografii". Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj".

Faza wstępna:

1. Uczniowie prezentują przygotowane zagadnienia.
2. Nauczyciel wyświetla temat i cele zajęć. Prosi uczniów, by na podstawie wiadomości zdobytych przed lekcją zaproponowali kryteria sukcesu.
3. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj". Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Animacja", wybrany uczeń czyta treść polecenia nr 1: „Dzień po przesłaniu do przyjaciela zaszyfrowanej wiadomości, z opisanym miejscem spotkania, otrzymujesz odpowiedź – również zaszyfrowaną. Aby dokonać jej deszyfracji, przeanalizuj prezentację." i omawia przykładowe rozwiązanie postawionego problemu. Następnie nauczyciel prosi uczniów o zapoznanie się z animacją i wykonanie ćwiczenia nr 1.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1-8 z sekcji „Sprawdź się". Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.
4. Metodą burzy mózgów uczniowie przygotowują mapę myśli będącą podsumowaniem wiadomości na temat przedstawionych na lekcji szyfrów.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie" i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).
2. Nauczyciel prosi uczniów o podsumowanie zgromadzonej wiedzy.

Praca domowa:

1. Zaszyfruj tekst „INFORMATYKA" szyfrem płotkowym o kluczu 3 i 4. Porównaj uzyskane szyfrogramy.

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj", „Animacja", „Sprawdź się" jako materiał do lekcji powtórkowej.