Szyfrowanie i deszyfrowanie – zadania maturalne

Bibliografia:

- Źródło: oprac. własne.

Szyfrowanie i deszyfrowanie – zadania maturalne

Źródło: Tomasz_Mikolajczyk, dostępny w internecie: pixabay.com, domena publiczna.

Poznaliśmy już kilka algorytmów szyfrujących. Rozumiemy, jak działają, i potrafimy je zaimplementować w różnych językach programowania. Z e-materiału Szyfrowanie i deszyfrowanie – projekt dowiedzieliśmy się, jak zaplanować pracę nad projektem programu szyfrującego. Z jego implementacjami w poszczególnych językach programowania zapoznaliśmy się w e-materiałach:

- Szyfrowanie i deszyfrowanie w języku C++,
- Szyfrowanie i deszyfrowanie w języku Java,
- Szyfrowanie i deszyfrowanie w języku Python.

W tym e-materiale przećwiczymy zdobytą wiedzę, rozwiązując zadania maturalne.

> **Twoje cele**
>
> - Sprawdzisz w praktyce wiedzę dotyczącą szyfrowania i deszyfrowania informacji.
> - Rozwiążesz przykładowe zadania typu maturalnego z zakresu szyfrowania i deszyfrowania informacji.
> - Przeanalizujesz schemat oceniania rozwiązywanych zadań.

Przeczytaj

Przypomnienie najważniejszych informacji

Szyfrowanie to proces mający na celu zamianę jawnej wiadomości w szyfrogram – przekształconą wiadomość, której odczytanie przez osobę postronną nie sprawi, że pozna ona oryginalną informację. Do szyfrowania wykorzystujemy zwykle pewien klucz, którego forma i wartość dostosowana jest do wybranego przez nas algorytmu kryptograficznego. W zależności od typu szyfrowania przechwycenie klucza przez osobę nieupoważnioną może, ale nie musi pozwolić jej na odszyfrowanie wiadomości. Można spotkać różne podziały szyfrów, m.in.:

- szyfry symetryczne – do procesu szyfrowania i deszyfrowania wykorzystujemy ten sam klucz,
- szyfry asymetryczne – stosuje się osobne klucze do szyfrowania i deszyfrowania wiadomości,
- szyfry podstawieniowe – podczas procesu szyfrowania każdy znak jawnej wiadomości zamieniany jest na inny znak lub ciąg znaków,
- szyfry przestawieniowe – w trakcie szyfrowania zamieniana jest kolejność znaków w wiadomości.

Obecnie stosowane szyfry to skomplikowane algorytmy, często uniemożliwiające szybkie ich złamanie. W zadaniach maturalnych skupimy się jednak na o wiele prostszych, historycznych przykładach metod szyfrowania.

Zadanie 1 – Szyfr kolumnowy

Zadanie pochodzi z pierwszej części egzaminu maturalnego z informatyki na poziomie rozszerzonym, który odbył się w czerwcu 2019 r.

Treść zadania

Szyfrowanie kolumnowe jest jedną z metod szyfrowania przestawieniowego, polegającego na zmianie kolejności znaków w szyfrowanym tekście. W tej metodzie jest wykorzystywana tabela o dodatniej liczbie wierszy równej k. Liczba k jest nazywana kluczem. Wiersze i kolumny tabeli są numerowane liczbami naturalnymi, począwszy od 1. Znaki tekstu, który ma być zaszyfrowany, wpisujemy do kolejnych kolumn tabeli, zaczynając od jej lewego górnego rogu. W kolumnach nieparzystych znaki wpisujemy od góry do dołu, a w parzystych od dołu do góry. Puste miejsca w ostatniej rozpoczętej kolumnie wypełniamy znakiem „_” oznaczającym spację. Następnie odczytujemy kolejne wiersze od góry do dołu (każdy z nich od lewej do prawej), w wyniku czego uzyskujemy szyfrogram.

Przykład: dla klucza k=3 i tekstu MATURA_Z_INFORMATYKI budujemy tabelę:

M	A	_	F	O	Y	K
A	R	Z	N	R	T	I
T	U	_	I	M	A	_

i otrzymujemy szyfrogram MA_FOYKARZNRTITU_IMA_.

Podpunkt 1

Do zaszyfrowania pewnego 40-znakowego cytatu z wypowiedzi Juliusza Cezara użyto metody szyfru kolumnowego o kluczu 10. Otrzymano szyfrogram:

NKI_ATE_USGACYOKZZ_YYSJTCWEKI_SAEMTRLE_P

Rozszyfruj ten cytat.

Rozwiązanie

Do rozszyfrowania tego cytatu potrzebować będziemy podobnej tabeli, do tej przedstawionej w treści zadania. Podany cytat jest w postaci szyfrogramu, dlatego jego znakami wypełniamy po kolei następujące po sobie wiersze (będzie ich dziesięć, ponieważ taka jest też wartość klucza).

N	K	I	_
A	T	E	_
U	S	G	A
C	Y	O	K
Z	Z	_	Y
Y	S	J	T
C	W	E	K
I	_	S	A
E	M	T	R
L	E	_	P

Odczytajmy teraz tekst jawny, stosując opisane w treści zadania wytyczne. W rezultacie otrzymujemy:

NAUCZYCIELEM_WSZYSTKIEGO_JEST_PRAKTYKA

Jest to rozwiązanie naszego zadania.

Podpunkt 2

W wybranym przez siebie języku programowania, za pomocą pseudokodu lub w postaci listy kroków, napisz algorytm deszyfrujący tekst, który został zakodowany szyfrem kolumnowym.

Specyfikacja:

Dane:

- `k` – klucz, liczba całkowita większa od 0
- `n` – liczba znaków w tekście zaszyfrowanym, n jest wielokrotnością k
- `S[1..n]` – ciąg znaków (tekst do odszyfrowania)

Wynik:

- `T[1..n]` – ciąg znaków (tekst odszyfrowany)

Rozwiązanie

Pierwszym krokiem naszego algorytmu będzie zdefiniowanie odpowiednich zmiennych. Należeć do nich będą (przyjęte oznaczenia mogą być inne):

- `P` – liczba kolumn w tabeli,
- `W` – pozycja w szyfrogramie litery aktualnie dopisywanej do tekstu jawnego,
- `z` – różnica pomiędzy pozycjami kolejnych liter, odczytywanych z szyfrogramu.

Początkowe wartości zdefiniowanych zmiennych:

```
P = n/k
```

```
W = 0
```

```
z = P
```

Mimo że operować będziemy na ciągu znaków, musimy zachować wyobrażenie utworzonej do szyfrowania tabeli. W naszym algorytmie wykorzystamy dwie pętle:

- pierwszą zawierającą operacje odczytywania liter z rozpatrywanej kolumny (liczba powtórzeń tej pętli równa jest kluczowi `k`),
- drugą odpowiedzialną za operacje przejścia pomiędzy parzystymi i nieparzystymi iteracjami pierwszej pętli (`P` powtórzeń).

Zależność operacji od parzystości powtórzenia pętli wynika z konstrukcji tabeli – parzyste kolumny zapisywane są od dołu do góry, natomiast nieparzyste od góry do dołu.

Wewnątrz pętli odpowiedzialnej za odczytanie liter wykonywać będziemy dwie operacje – najpierw dopisanie litery szyfrogramu z pozycji `W` do ciągu znaków tekstu jawnego, a następnie dodanie wartości zmiennej `z` do aktualnej wartości zmiennej `W`.

Przed każdym wykonaniem tej pętli musimy dodać do `W` liczbę jeden – przed pierwszą iteracją w celu ustalenia odczytu litery z pierwszej pozycji w szyfrogramie, natomiast w przypadku kolejnych iteracji ze względu na fakt, że dwie sąsiednie litery tekstu jawnego, znajdujące się w dwóch różnych kolumnach tabeli, w szyfrogramie również ze sobą sąsiadują. Operację tę zamieścimy oczywiście w pętli odpowiedzialnej za przejścia pomiędzy kolejnymi kolumnami (parzystymi i nieparzystymi).

Pozostaje nam dodać jeszcze dwie operacje w tej samej pętli, tym razem po wykonaniu odczytu z kolumn. Pierwsza z nich to odjęcie od zmiennej `W` wartości zmiennej `z` – w ostatniej iteracji pętli odczytującej litery z kolumn osiągamy wartość przekraczającą liczbę liter w wiadomości. Drugą operacją natomiast jest zmiana znaku zmiennej `z` wynikająca z różnego sposobu zapisu kolumn zależnego od parzystości. W ten sposób otrzymujemy gotowy algorytm odszyfrowujący.

Przykładowy algorytm będący poprawnym rozwiązaniem zadania:

```
P ← n / k
W ← 0
z ← P
    dla i = 1, 2, ..., P wykonuj
    W ← W + 1
    dla j = 1, 2, ..., k wykonuj
        T ← T + S[W]
        W ← W + z
    W ← W - z
    z ← z * (-1)
```

Schemat oceniania

Numer podpunktu	Punktacja za część zadania	Maksymalna punktacja za część zadania
1.	Za poprawne odszyfrowanie: **2 punkty**. Za odpowiedź z jednym błędem: **1 punkt**.	**2**

Numer podpunktu	Punktacja za część zadania	Maksymalna punktacja za część zadania
2.	Za poprawny algorytm: **4 punkty**, w tym: – za poprawne przestawianie wskaźnika elementu początkowego dla pętli wewnętrznych (po wykonaniu pętli wewnętrznej +1): **1 punkt**, – za poprawne indeksowanie komórek przy nieparzystej iteracji (zwiększanie o P): **1 punkt**, – za poprawne indeksowanie komórek przy nieparzystej iteracji (zmniejszanie o P): **1 punkt**, – za poprawny kierunek indeksowania pętli odczytujących: **1 punkt**.	**4**

Słownik

kryptografia

dziedzina zajmująca się zabezpieczaniem danych przed dostępem do nich osób do tego nieupoważnionych

tekst jawny

informacja zapisana w sposób umożliwiający jej odczyt i poznanie, bez przeprowadzania procesu deszyfracji

Prezentacja multimedialna

Zadanie 2

Szyfr Vigenère'a to jeden z szyfrów należących do grupy szyfrów polialfabetycznych. Do szyfrowania za pomocą szyfru Vigenère'a wykorzystywana jest następująca struktura:

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
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	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	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

W każdym kolejnym wierszu znajdują się permutacje alfabetu łacińskiego (kolejność liter jest przesunięta o jeden, względem poprzedniego).

Jeżeli chcemy zaszyfrować przykładowy ciąg znaków „GRACZ” kluczem „KLUCZ” za pomocą szyfru Vigenere'a, postępujemy zgodnie z listą kroków:

1. Znajdź przecięcie pionowego i poziomego rzędu, w którym w przypadku poziomego rzędu pierwszą literą jest „G” (pierwsza litera ciągu do zaszyfrowania), a w przypadku pionowego rzędu pierwszą literą rzędu jest „K” (pierwsza litera klucza).
2. Zaszyfrowaną literą jest – znalezione w punkcie 1 – przecięcie rzędów.
3. Weź kolejną literę słowa do zaszyfrowania oraz kolejną literę klucza i wróć do kroku nr 1 (w przypadku gdy nie ma kolejnych liter klucza, należy wziąć pierwszą literę klucza).

Gdy wykonamy kroki algorytmu dla wszystkich znaków słowa, do zaszyfrowania „GRACZ”, wykorzystując klucz „KLUCZ”, otrzymamy zaszyfrowane słowo:

```
QCUEY
```

Podpunkt 1

W pliku `dane.txt` znajdują się ciągi znaków, które pracownik agencji wywiadowczej ma za zadanie zaszyfrować za pomocą szyfru Vigenere'a. Każde słowo jest zapisane w osobnej linii.

Plik o rozmiarze 98.00 B w języku polskim

Napisz program, który zaszyfruje kolejne ciągi znaków i wyniki zaszyfrowań zapisze do pliku `zaszyforwane.txt` (każde zaszyfrowane słowo w osobnej linii). Każde słowo ma zostać zaszyfrowane za pomocą klucza „TAJNE”.

> Rozwiązanie przedstawimy przy użyciu pseudokodu ze względu na fakt, że zadania maturalne, dotyczące programowania, można rozwiązać w wybranym języku programowania: C++, Java lub Python.
> **Praca domowa**
>
> Dokonaj implementacji powyższego zadania lokalnie na swoim komputerze, w języku, w którym programujesz. Zadbaj o prawidłowe wczytanie danych z pliku tekstowego do swojego programu. Odpowiedź do zadania, dla danych z pliku, znajdziesz na samym końcu zadania.
> **Polecenie 1**
>
> Przeanalizuj prezentację, która przedstawia rozwiązanie krok po kroku w postaci pseudokodu, a następnie rozwiąż zadanie w dowolnym języku programowania.

1

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

W tej prezentacji zostanie przedstawione, krok po kroku, rozwiązanie przykładowego zadania maturalnego. Pokażemy poprawny pseudokod, realizujący założenia zadania.

2

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

Pierwszym etapem zadania, który należy wykonać, jest utworzenie tabeli kodowań

i umieszczenie jej w odpowiedniej zmiennej.

3

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

Tabela kodowań zostanie zapisana w tablicy dwuwymiarowej o rozmiarze 26 na 26.

W tablicy jednowymiarowej `alfabet` będą się znajdowały kolejne litery alfabetu łacińskiego.

4

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

W funkcji `wypelnij()` tabela kodowań zostanie wypełniona znakami alfabetu łacińskiego.

Pierwszym krokiem, jaki zostanie wykonany po wejściu w blok funkcji `wypelnij()`, będzie wypełnienie tablicy `alfabet` kolejnymi literami alfabetu łacińskiego.

Funkcja `ascii()` jako parametr przyjmuje dodatnią liczbę całkowitą i zwraca odpowiadający jej znak z tabeli ASCII.

5

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

Następnie przejdźmy do wypełniania tablicy `tabelaKodowan`.

Przeanalizujmy trzy pętle:

Pierwsza z nich iteruje po kolejnych wierszach tabeli kodowań, druga po kolejnych jej komórkach, natomiast trzecia z nich ma za zadanie dokonywać kolejnych permutacji alfabetu łacińskiego. Funkcja `zamien()` dokonuje zamiany miejscami dwóch zadanych elementów tablicy.

6

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

Oto cały blok gotowej funkcji `wypelnij()`

7

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

Następnie utworzymy kolejną funkcję – `szyfrowanie()`, która dokona szyfrowania zadanego ciągu znaków, za pomocą określonego klucza.

Funkcja ta będzie przyjmować dwa parametry – ciąg znaków i klucz. Oba będą łańcuchami znaków.

8

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

W poniższej funkcji dokonywana jest operacja znajdowania wyniku szyfrowania – punktu przecięcia dwóch rzędów, rozpoczynających się od danego znaku klucza i słowa do zaszyfrowania.

Funkcja `asciiDec()` zwraca wartość dziesiętną (z tabeli ASCII) przypisaną do zadanego w parametrze znaku.

9

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

Należy jeszcze pamiętać o przypadku, w którym klucz jest krótszy niż słowo do zaszyfrowania.

10

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

Gdy obie funkcje są gotowe, wykorzystajmy do zaszyfrowania słów z pliku tekstowego `dane.txt`.

11

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/PsfaUPdII

Oto całe rozwiązanie zadania w pseudokodzie:

Schemat oceniania

Numer podpunktu	Punktacja za część zadania	Maksymalna punktacja za podpunkt
1	Za poprawny algorytm: **4 punkty**, w tym: – za poprawne utworzenie tabeli kodowań: **2 punkty** - za poprawne iterowanie po kolejnych komórkach tabeli kodowań: **1 punkt** – za poprawne tworzenie zaszyfrowanego słowa**: 1 punkt**	4 p.

Praca domowa

Odpowiedź do zadania dla danych zawartych w pliku tekstowym:

```
ZRJSP
WAU
DOANP
KAON
LKXETBOW
ZEWRDT
EOTV
DRJW
LEA
MATGCDA
VHVHVT
IOPBHT
KESF
LTJGID
```

```
MRJXXHR
FAAZYK
```

Sprawdź się

Zadanie 2.

Podpunkt 2.

Zaszyfrowane wiadomości z pierwszej części zadania okazały się bardzo łatwe do odszyfrowania. W związku z tym pracownik agencji wywiadowczej dostał zadanie, aby oprócz zaszyfrowania słów za pomocą szyfru Vigenère'a zastosował dodatkowe szyfrowanie.

Pracownik wymyślił, że słowa zostaną najpierw zaszyfrowane za pomocą szyfru Vigenère'a , a następnie za pomocą szyfru Cezara.

W pliku `dane.txt` znajdują się słowa do zaszyfrowania, a obok każdego słowa jest napisana liczba, która oznacza wartość przesunięcia (klucz) szyfru Cezara.

Plik o rozmiarze 138.00 B w języku polskim

Napisz program, który zaszyfruje zadane w pliku ciągi znaków za pomocą szyfru Vigenère'a , kluczem „KLUCZ”, a następnie za pomocą szyfru Cezara, według podanej w pliku wartości klucza. Wyniki należy zapisać do osobnego pliku.

Przykładowy ciąg znaków „GRAF” i klucz równy 4.

```
GRAF 4
```

Wynik szyfrowania:

```
UGYL
```

Dane zawarte w pliku tekstowym zostały umieszczone w tablicach i znajdują się w poniższych ćwiczeniach. Pamiętaj jednak, że rozwiązując zadania maturalne, musisz dokonać prawidłowego wczytania danych z pliku tekstowego do swojego programu.

Pokaż ćwiczenia:

Ćwiczenie 1

Napisz program w języku C++, który zaszyfruje zadane w pliku ciągi znaków za pomocą szyfru Vigenère'a, kluczem „KLUCZ”, a następnie za pomocą szyfru Cezara, według podanej w pliku wartości klucza.

Specyfikacja:

Dane:

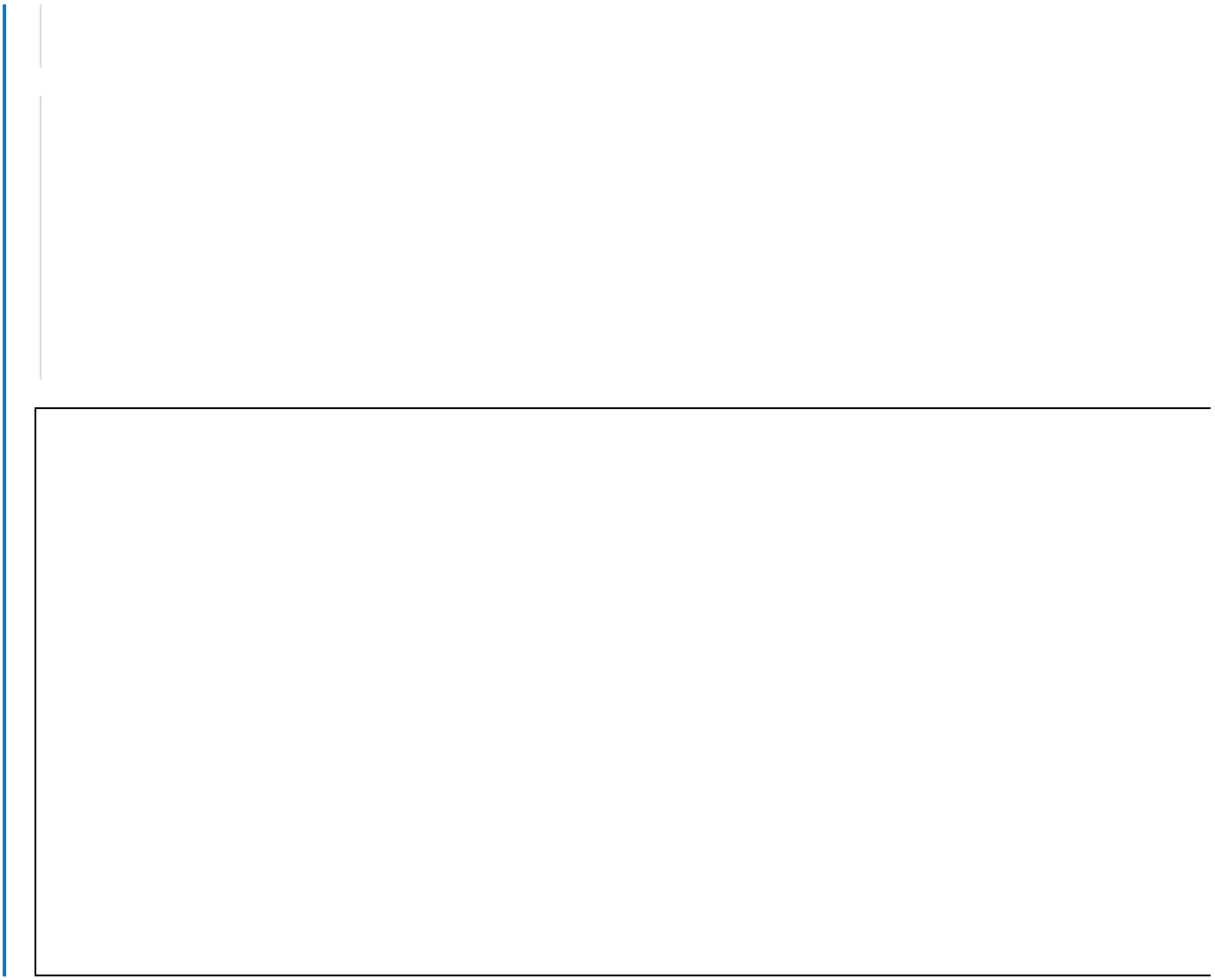
- `alfabet, tabela` – tablica znaków
- `dane` – tablica łańcuchów znaków
- `klucze` – tablica elementów typu *int*

Wynik:

Program na wyjściu standardowym zwróci zakodowane za pomocą szyfru Cezara łańcuchy znaków.

Twoje zadania

1. Program ma wypisać poprawnie zakodowane słowa (zgodnie z poleceniem zadania) z pliku `dane.txt`. Dane umieszczone są w tablicy `dane`, a klucze w `klucze`.



Ćwiczenie 2

Napisz program w języku Java, który zaszyfruje zadane w pliku ciągi znaków za pomocą szyfru Vigenère'a, kluczem „KLUCZ”, a następnie za pomocą szyfru Cezara, według podanej w pliku wartości klucza.

Specyfikacja:

Dane:

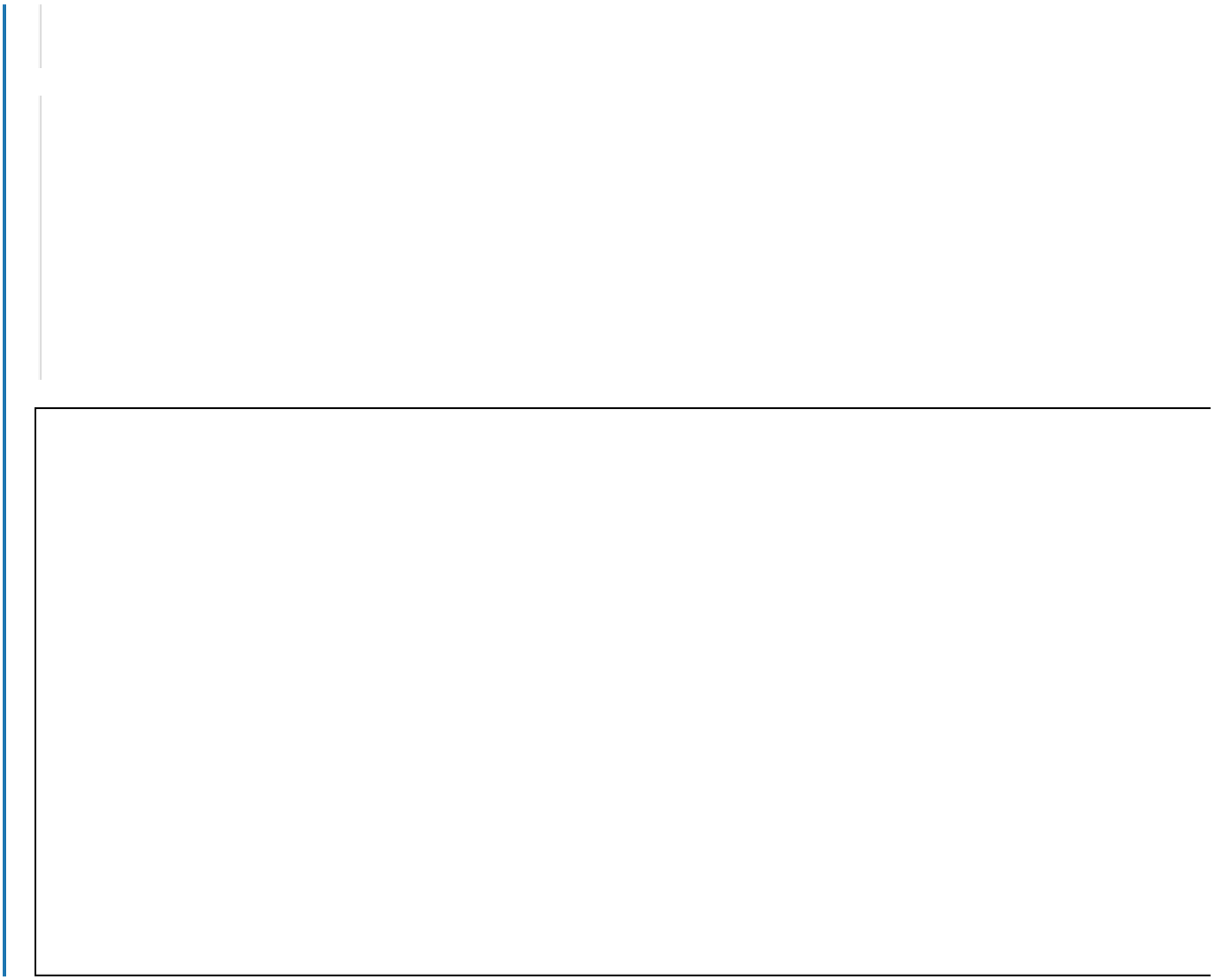
- `alfabetLacinski, tabela` – tablica znaków
- `dane` – tablica łańcuchów znaków
- `klucze` – tablica elementów typu *int*

Wynik:

Program na wyjściu standardowym zwróci zakodowane za pomocą szyfru Cezara łańcuchy znaków.

Twoje zadania

1. Program ma wypisać poprawnie zakodowane słowa (zgodnie z poleceniem zadania) z pliku `dane.txt`. Dane umieszczone są w tablicy `dane`, a klucze w `klucze`.



Ćwiczenie 3

Napisz program w języku Python, który zaszyfruje zadane w pliku ciągi znaków za pomocą szyfru Vigenère'a, kluczem „KLUCZ”, a następnie za pomocą szyfru Cezara, według podanej w pliku wartości klucza.

Specyfikacja:

Dane:

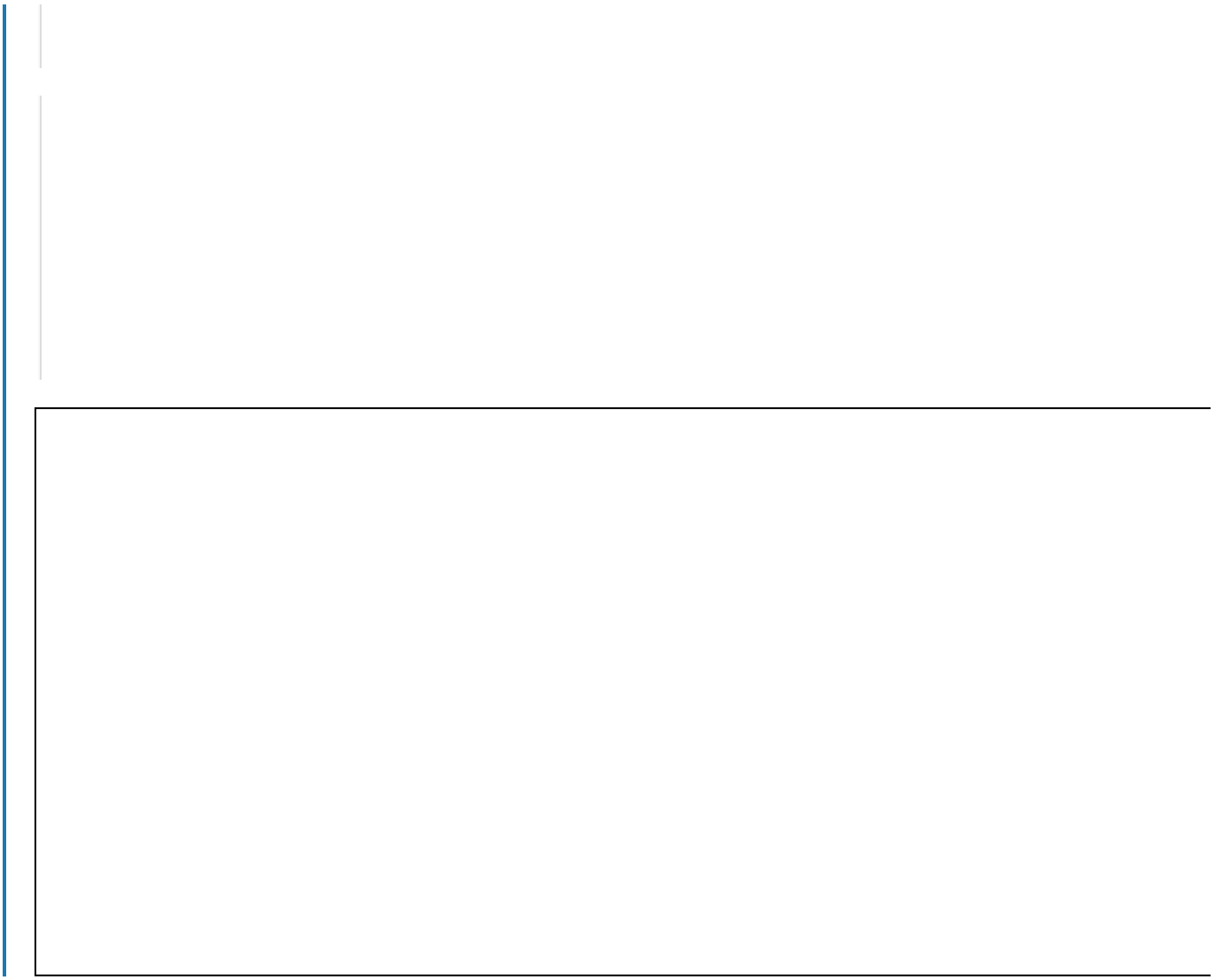
- `alfabet, tabela` – tablica znaków
- `dane` – tablica łańcuchów znaków
- `klucze` – tablica elementów typu *int*

Wynik:

Program na wyjściu standardowym zwróci zakodowane za pomocą szyfru Cezara łańcuchy znaków.

Twoje zadania

1. Program ma wypisać poprawnie zakodowane słowa (zgodnie z poleceniem zadania) z pliku `dane.txt`. Dane umieszczone są w liście `dane`, a klucze w `klucze`.



Dla nauczyciela

Autor: Maurycy Gast
Przedmiot: Informatyka

Temat: Szyfrowanie i deszyfrowanie – zadania maturalne

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

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,

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

I + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) objaśnia, a także porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przy tym przykłady problemów i algorytmów, w szczególności:

f) metodę szyfrowania z kluczem publicznym i jej zastosowanie w podpisie elektronicznym,

IV. Rozwijanie kompetencji społecznych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) analizuje i charakteryzuje wpływ trendów w historycznym rozwoju pojęć, metod informatyki oraz technologii na możliwości rozwiązywania problemów teoretycznych i praktycznych;

V. Przestrzeganie prawa i zasad bezpieczeństwa.

Zakres podstawowy. Uczeń:

3) stosuje dobre praktyki w zakresie ochrony informacji wrażliwych (np. hasła, pin), danych i bezpieczeństwa systemu operacyjnego, objaśnia rolę szyfrowania informacji;

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) objaśnia rolę technik uwierzytelniania, kryptografii i podpisu elektronicznego w ochronie i dostępie do informacji;

2) omawia znaczenie algorytmów szyfrowania i składania podpisu elektronicznego.

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):

- Sprawdzisz w praktyce wiedzę dotyczącą szyfrowania i deszyfrowania informacji.

- Rozwiążesz przykładowe zadania typu maturalnego z zakresu szyfrowania i deszyfrowania informacji.
- Przeanalizujesz schemat oceniania rozwiązywanych zadań.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium i ćwiczeń interaktywnych;
- burza mózgów;
- ćwiczenia praktyczne.

Formy pracy:

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

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. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Szyfrowanie i deszyfrowanie – zadania maturalne". Uczniowie zapoznają się z treściami w sekcji „Przeczytaj".

Faza wstępna:

1. Nauczyciel prosi, by metodą burzy mózgów uczniowie przypomnieli najważniejsze informacje dotyczące szyfrowania i deszyfrowania. Chętna lub wybrana osoba zapisuje je na tablicy w formie mapy myśli. W razie potrzeby nauczyciel uzupełnia propozycje uczniów.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie analizują treści z sekcji „Przeczytaj” wyświetlone na tablicy. Uczniowie zapoznają się zadaniem 1 i powtarzają zaprezentowane rozwiązanie na swoim komputerze.
2. **Praca z multimedium.** Uczniowie, pracując w parach, zapoznają się z treścią zadania z sekcji „Prezentacja multimedialna”, a następnie analizują prezentację, w której omówiono jego rozwiązanie za pomocą pseudokodu. W kolejnym kroku nauczyciel dzieli klasę na grupy, w których uczniowie opracowują rozwiązanie omawianego zadania w wybranych językach programowania. Jedna osoba z grupy prezentuje rozwiązanie na forum klasy.
3. **Ćwiczenie umiejętności.** Liga zadaniowa - uczniowie wykonują indywidualnie na czas ćwiczenia nr 1-3 z sekcji „Sprawdź się”, a następnie omawiają zadania na forum.

Faza podsumowująca:

1. Na koniec zajęć nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak…”.

Praca domowa:

1. Uczniowie wykonują pracę domową dołączoną do zadania i polecenia w sekcji „Prezentacja multimedialna”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Prezentacja multimedialna”, „Sprawdź się” jako materiał do lekcji powtórkowej.