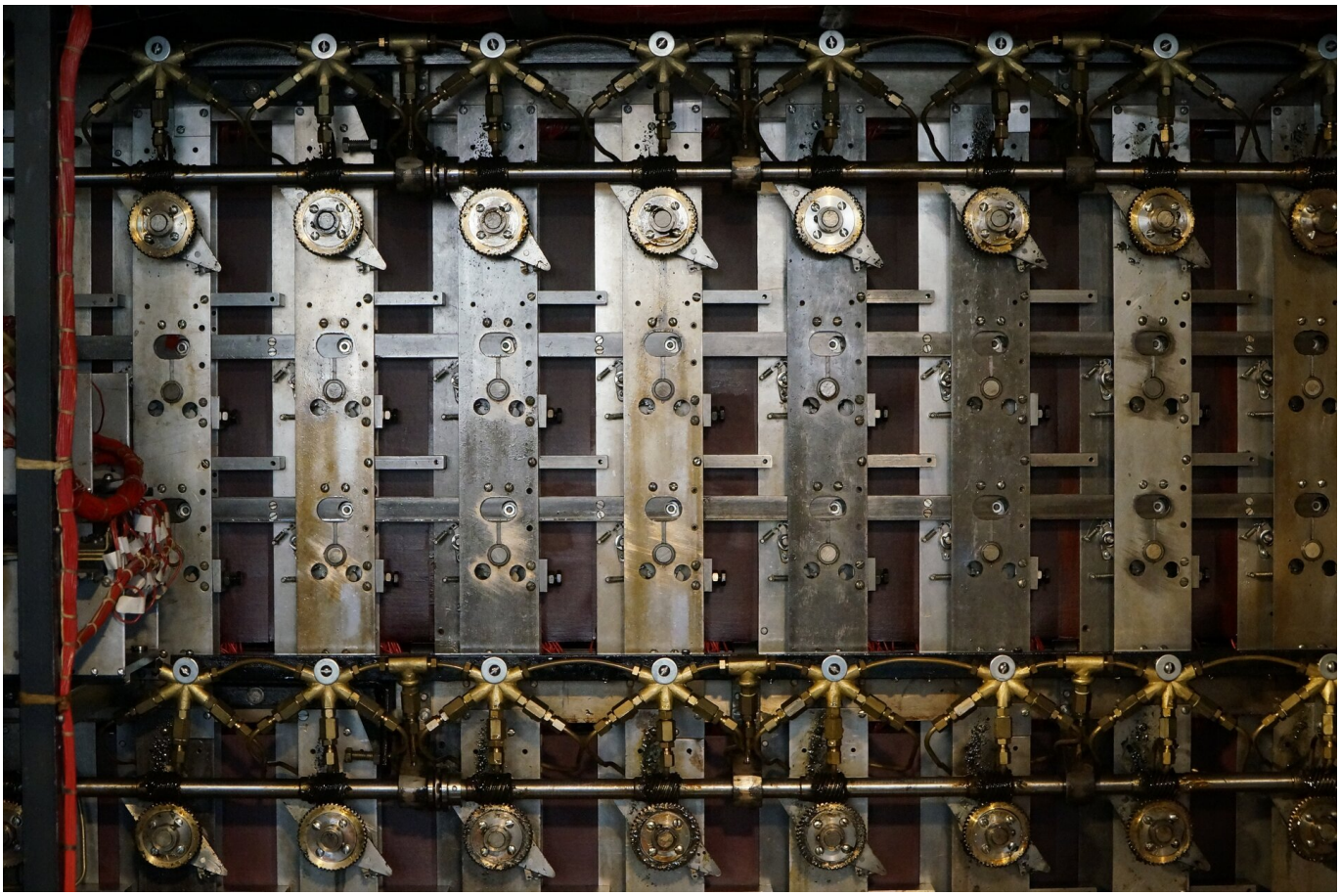


Szyfry symetryczne i asymetryczne – zadania maturalne

Bibliografia:

- Źródło: oprac. własne.

Źródło: awsloley, domena publiczna.

Poznaliśmy już definicje szyfrów symetrycznych i asymetrycznych. Wiemy, czym różnią się te typy szyfrów, jak się dzielą oraz jakie jest ich zastosowanie.

W tym e-materiale zapoznamy się z przykładowymi zadaniami maturalnymi dotyczącymi tego zagadnienia.

Implementacje algorytmów w poszczególnych językach programowania zostały omówione w e-materiałach:

- Szyfry symetryczne i asymetryczne w języku C++,
- Szyfry symetryczne i asymetryczne w języku Java,
- Szyfry symetryczne i asymetryczne w języku Python.

Twoje cele

- Zastosujesz w praktyce swoją wiedzę na temat szyfrów symetrycznych i asymetrycznych.
- Rozwiążesz przykładowe zadania typu maturalnego z zakresu szyfrów symetrycznych i asymetrycznych.
- Powtórzysz wiadomości dotyczące szyfrów symetrycznych i asymetrycznych.

Przeczytaj

Zadanie 1. Szyfr

Rozważamy szyfrowanie przestawieniowe, w którym kluczem jest `n`-elementowa tablica zawierająca różne liczby całkowite z przedziału `[1, n]`. Kluczem 5-elementowym może być np. tablica `[3,2,5,4,1]`.

Szyfrowanie napisu `A` (o długości co najmniej `n`) kluczem `n`-elementowym `P[1..n]` odbywa się w następujący sposób:

- pierwsza litera słowa `A` zamieniana jest miejscem z literą na pozycji `P[1]`,
- następnie druga litera słowa `A` zamieniana jest z literą na pozycji `P[2]`,
- i tak dalej.

Uzyskane na końcu słowo jest szyfrem napisu `A` z kluczem `P`.

Jeśli napis `A` ma więcej niż `n` liter, to po `n`-tym kroku algorytmu kolejną literę znów zamieniamy z literą na pozycji `P[1]` i tak dalej. Oznacza to, że w `i`-tym kroku zamieniamy litery na pozycjach `i` oraz `P[1 + (i - 1) mod n]`.

Przykład:

Tabela ilustruje szyfrowanie słowa `INFORMATYKA` kluczem `P` równym `[3,2,5,4,1]`:

i	1	2	3	4	5	6	7	8	9	10	11
P[1 + (i - 1) mod n]	3	2	5	4	1	3	2	5	4	1	3
Słowo	I	N	F	O	R	M	A	T	Y	K	A
Krok 1	**F**	N	**I**	O	R	M	A	T	Y	K	A
Krok 2	F	**N**	I	O	R	M	A	T	Y	K	A
Krok 3	F	N	**R**	O	**I**	M	A	T	Y	K	A
Krok 4	F	N	R	**O**	I	M	A	T	Y	K	A
Krok 5	**I**	N	R	O	**F**	M	A	T	Y	K	A
Krok 6	I	N	**M**	O	F	**R**	A	T	Y	K	A
Krok 7	I	**A**	M	O	F	R	**N**	T	Y	K	A
Krok 8	I	A	M	O	**T**	R	N	**F**	Y	K	A
Krok 9	I	A	M	**Y**	T	R	N	F	**O**	K	A
Krok 10	**K**	A	M	Y	T	R	N	F	O	**I**	A
Krok 11	K	A	**A**	Y	T	R	N	F	O	I	**M**

Napis `KAAYTRNFOIM` jest zatem szyfrem tekstu `INFORMATYKA` przy użyciu klucza `[3,2,5,4,1]`.

Zadanie 1.1

W pliku `szyfr1.txt` dane są:

- w wierszach nr 1–6 – napisy złożone z 50 liter alfabetu łacińskiego,
- w wierszu nr 7 – klucz 50-elementowy; liczby oddzielone są pojedynczym znakiem odstępu.

Napisz program, który zaszyfruje wszystkie napisy zawarte w udostępnionym pliku zgodnie z metodą opisaną na początku sekcji. Wynik, czyli zaszyfrowane teksty, zapisz w osobnych wierszach w pliku `wyniki_szyfr1.txt`.

Plik o rozmiarze 452.00 B w języku polskim

Do oceny oddajesz:

- plik `wyniki_szyfr1.txt` z odpowiedzią (zaszyfrowane napisy zapisane każdy w osobnym wierszu),
- plik(i) z komputerową realizacją zadania (kodem programu).

> Rozwiąż zadanie dla załączonych danych. W tym celu zastosuj jeden z języków programowania: C++, Java lub Python. Odpowiedź do zadania znajdziesz w osobnym pliku, umieszczonym po omówieniu przykładowego rozwiązania.
> Zadanie zostało opracowane na podstawie zadania nr 76, które pojawiło się w zbiorze zadań z informatyki autorstwa Centralnej Komisji Egzaminacyjnej.
> **Ważne!**
>
> Do rozwiązania zadania przydatne jest zapamiętanie klucza (z wiersza nr 7) w tablicy 5-elementowej. Niech `P[0]`, `P[1]`, ..., `P[49]` oznaczają kolejne liczby klucza. W zadaniu należy zaszyfrować 6 danych napisów za pomocą klucza `P` – w tym celu trzeba napisać funkcję, a następnie uruchomić ją dla kolejnych napisów.
>
> Niech `A[0..49]` będzie napisem, który chcemy zaszyfrować kluczem `P[0..49]`. Podkreślmy, że w tablicy `P` znajdują się wartości od 1 do 50, co oznacza, że trzeba zachować ostrożność przy implementacji procedury szyfrującej – za każdym razem należy zamieniać literę `A[i]` z literą `A[P[i]-1]`. Procedurę tę warto zaprogramować w pętli, która wykona operacje przestawiania liter w takiej samej kolejności, jak to zostało podane w treści zadania, czyli od pierwszej do ostatniej litery słowa `A`. Klucz jest tej samej długości co słowo `A`, dlatego wystarczy kolejno zamieniać literę `A[i]` z literą `A[P[i]-1]` dla kolejnych indeksów `i = 0, 1, ..., 49`. Na końcu otrzymamy zaszyfrowane słowo wyjściowe.
> **Uwaga:** powyższy komentarz do zadania pochodzi ze zbioru przygotowanego przez CKE.

Przykładowe rozwiązanie

Najpierw wczytujemy dane z pliku. W tym celu będziemy potrzebować dwóch tablic: jednej dla napisów do zaszyfrowania, drugiej dla klucza.

```
napisy_do_zaszyfrowania[0..5] - wczytaj napisy do zaszyfrowania z
klucz[0..49] - wczytaj klucz z pliku - jak w przypadku napisów po
```

Mamy zaszyfrować 6 napisów zapisanych w tablicy. W tym celu utwórzmy pętlę umożliwiającą wykonanie szyfrowania dla kolejnych napisów zapisanych w tablicy `napisy_do_zaszyfrowania[]`.

```
napisy_do_zaszyfrowania[0..5] - wczytaj napisy do zaszyfrowania z
klucz[0..49] - wczytaj klucz z pliku - jak w przypadku napisów po

dla i = 0, 1, ... 5 wykonuj
```

Następnym krokiem jest zaszyfrowanie napisów odczytanych wcześniej z pliku. W tym celu potrzebna jest pętla, która przeiteruje po wszystkich literach, z których składa się napis, i zaszyfruje je zgodnie ze sposobem opisanym na początku sekcji. W tym celu wyciągamy literę pod indeksem równym numerowi kroku pętli (kroku szyfrowania - 1, przez indeksowanie od 0) do zmiennej pomocniczej. Następnie na miejsce tej litery zapisujemy literę z miejsca wyznaczonego przez liczbę zapisaną w odpowiadającym literze miejscu w kluczu (liczba z klucza pomniejszona o 1 ze względu na indeksowanie tablic od 0). Następnie w miejscu litery z indeksu wyznaczonego kluczem zapisujemy literę przechowywaną w zmiennej pomocniczej.

```
napisy_do_zaszyfrowania[0..5] - wczytaj napisy do zaszyfrowania z
klucz[0..49] - wczytaj klucz z pliku - jak w przypadku napisów po

dla i = 0, 1, ... 5 wykonuj
    dla j = 0, 1, ..., 49 wykonuj
        pomocnicza ← napisy_do_zaszyfrowania[i][j]
        napisy_do_zaszyfrowania[i][j] ← napisy_do_zaszyfrowania[i
        napisy_do_zaszyfrowania[i][klucz[1 + (j-1) mod 50] - 1]
```

Po zakończeniu obiegów obu pętli otrzymujemy tablicę zaszyfrowanych napisów. Pozostaje zapisać je do pliku wynikowego.

```
napisy_do_zaszyfrowania[0..5] - wczytaj napisy do zaszyfrowania z
```

```
klucz[0..49] - wczytaj klucz z pliku - jak w przypadku napisów po

dla i = 0, 1, ... 5 wykonuj
    dla j = 0, 1, ..., 49 wykonuj
        pomocnicza ← napisy_do_zaszyfrowania[i][j]
        napisy_do_zaszyfrowania[i][j] ← napisy_do_zaszyfrowania[i
        napisy_do_zaszyfrowania[i][klucz[1 + (j-1) mod 50] - 1]

dla i = 0, 1, ... 5 wykonuj
    zapisz napisy_do_zaszyfrowania[i] do pliku "wyniki_szyfr1.txt
```

Odpowiedź do zadania

Odpowiedź dla danych zapisanych w pliku `szyfr1.txt` znajduje się w załączniku.

Plik o rozmiarze 312.00 B w języku polskim

Słownik

klucz

w kryptografii: informacja niezbędna do przeprowadzenia procesu szyfrowania lub deszyfrowania

klucz prywatny

rodzaj klucza w szyfrach asymetrycznych służący do deszyfrowania szyfrogramów; dostęp do niego powinien mieć tylko odbiorca zaszyfrowanych wiadomości

klucz publiczny

rodzaj klucza w szyfrach asymetrycznych służący do szyfrowania tekstu jawnego; nie musi być chroniony ani utrzymywany w tajemnicy, może nawet zostać udostępniony publicznie

szyfr asymetryczny

rodzaj szyfru, w którym operuje się dwoma osobnymi kluczami – szyfrującym (publicznym) oraz deszyfrującym (prywatnym)

szyfr symetryczny

rodzaj szyfru charakteryzujący się zastosowaniem jednego klucza zarówno do szyfrowania, jak i deszyfrowania

Prezentacja multimedialna

Zadanie 2. Podpis elektroniczny

Bajtek otrzymał od przyjaciela 500 jawnych wiadomości. Do każdej z nich przyjaciel dołączył podpis elektroniczny, będący zaszyfrowanym skrótem wiadomości. Skróty utworzył, przekształcając wiadomość w 8-znakowy napis. Następnie zaszyfrował ten skrót za pomocą algorytmu A o tylko sobie znanym kluczu prywatnym `(e,n)`.

Bajtek chciałby odszyfrować wszystkie otrzymane skróty z użyciem algorytmu A, za pomocą ogólnie znanego klucza publicznego `(d,n)` o wartościach `(3,200)`.

W pliku `podpisy.txt` znajduje się 500 wierszy, z których każdy zawiera 8 liczb całkowitych z przedziału [22, 162], stanowiących elementy podpisu elektronicznego jednej wiadomości. Liczby w wierszu oddzielone są pojedynczymi znakami odstępu. Kolejność wierszy podpisów jest zgodna z kolejnością wierszy wiadomości (pierwszy wiersz podpisów odpowiada pierwszej wiadomości, drugi — drugiej i tak dalej).

Algorytm A deszyfrowania kluczem publicznym `(d,n)`

Deszyfrowanie polega na wykonaniu operacji:

$$x = (y \cdot d \; mod \; n)$$

gdzie za `y` należy przyjąć kolejne liczby tworzące podpis elektroniczny. Tekst wynikowy można otrzymać, składając w jeden napis reprezentacje graficzne kolejnych liczb `x` zgodnie ze standardem ASCII.

Algorytm A szyfrowania kluczem prywatnym `(e,n)`

Zaszyfrowanie za pomocą algorytmu A polega na wykonywaniu operacji:

$$y = (x \cdot e \; mod \; n)$$

Za `x` należy podstawić kody ASCII kolejnych znaków tekstu źródłowego. Uzyskany w ten sposób ciąg liczb jest podpisem elektronicznym wiadomości. Gdyby ktoś chciał złamać szyfr A, czyli wyznaczyć nieznany element `e` klucza prywatnego, musiałby znaleźć taką wartość `e`, względnie pierwszą z `d`, że:

$$e \cdot d \; mod \; n = 1$$

Uzasadnienia szukaj w prawach arytmetyki modularnej.

> **Polecenie 1**
>
> Napisz program, który odszyfruje skróty wiadomości ze wszystkich podpisów elektronicznych umieszczonych w pliku `podpisy.txt`, stosując algorytm A z kluczem publicznym `(d,n) = (3,200)`. Wyniki zapisz do pliku `wynik2.txt`, w takim samym formacie jak w pliku wejściowym (każdy znak oddzielony pojedynczym znakiem odstępu), zachowując oryginalne linijki.

Plik o rozmiarze 13.35 KB w języku polskim

Do oceny oddajesz:

- plik `wynik2.txt` zawierający odpowiedź do zadania (odszyfrowane skróty wiadomości ze wszystkich podpisów elektronicznych),
- plik(i) z komputerową realizacją zadania (kodem programu).

Polecenie 2

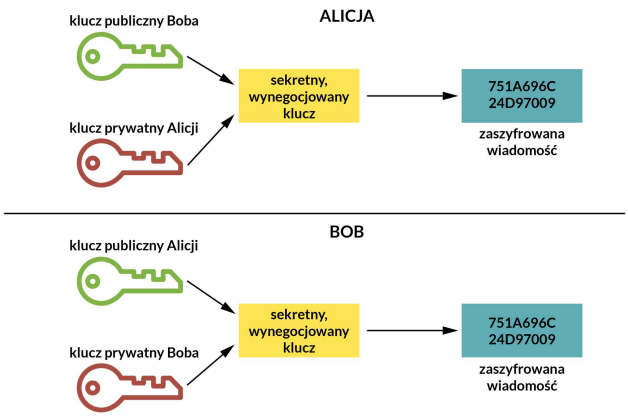
Przedstaw rozwiązanie w wybranym języku programowania: C++, Java lub Python. Odpowiedź dla danych z pliku znajdziesz pod prezentacją omawiającą kolejne kroki rozwiązania.

Zadanie zostało opracowane na podstawie zadania nr 78, które pojawiło się w zbiorze zadań z informatyki przygotowanego przez Centralną Komisję Egzaminacyjną.

Rozwiązanie

Rozwiązanie zadania przedstawimy w postaci pseudokodu.

1



Kryptografia klucza prywatnego – jeden z używanych kluczy jest dostępny publicznie. Każdy użytkownik może użyć tego klucza do zaszyfrowania wiadomości, ale tylko posiadacz drugiego, tajnego klucza może odszyfrować taką wiadomość.

Materiał audio dostępny pod adresem:

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

Do rozwiązania tego zadania posłużymy się równością $x = (y \cdot d \ mod \ n)$, gdzie x to odszyfrowana liczba, a y to kolejne liczby tworzące zaszyfrowany napis. Jest ona dostarczona w opisie algorytmu. Na początku wczytujemy dane z pliku `podpisy.txt`:

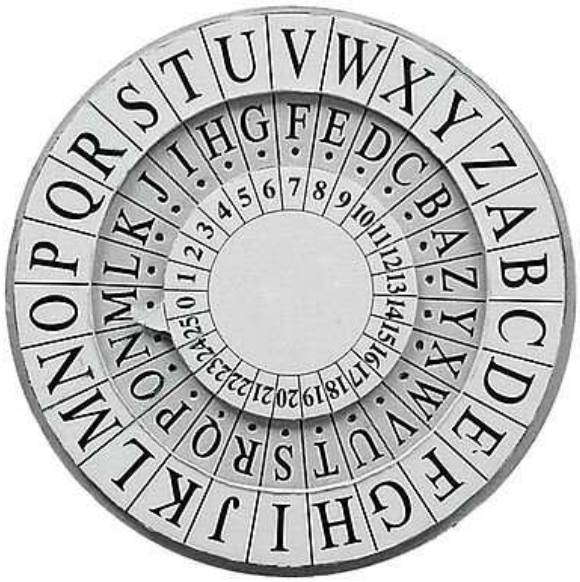
2

Materiał audio dostępny pod adresem:

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

Następnie przypisujemy zmiennym `N` oraz `D` wartości podane w zadaniu:

3



Urządzenie kryptograficzne Franka S.

Materiał audio dostępny pod adresem:

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

Tworzymy pierwszą pętlę, która będzie przechodzić przez kolejne zaszyfrowane skróty:

4

Materiał audio dostępny pod adresem:

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

Następnie tworzymy pętlę wewnętrzną, która będzie po kolei rozszyfrowywać liczby do odpowiednich liter:

5

Materiał audio dostępny pod adresem:

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

Teraz możemy przejść do rozszyfrowania każdej liczby po kolei. Tworzymy zmienną pomocniczą `a`, do której będziemy zapisywać odpowiednią wartość z tablicy `tabSkrot`:

6

Materiał audio dostępny pod adresem:

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

Zgodnie ze wzorem podanym w treści zadania musimy tę liczbę pomnożyć przez D:

7

Materiał audio dostępny pod adresem:

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

Następnie należy z niej wyciągnąć, za pomocą operacji mod, resztę z dzielenia przez N:

8

Materiał audio dostępny pod adresem:

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

Zapisujemy literę odpowiadającą rozszyfrowanej liczbie na koniec obecnej linii w pliku `wynik2.txt`:

9

Materiał audio dostępny pod adresem:

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

Przechodzimy do kolejnej linii w pliku `wynik2.txt`, by wpisywać tam rozszyfrowany kolejny skrót:

> **Ważne!**
>
> W pseudokodzie wykorzystaliśmy operator `mod` oraz funkcję `znak()`:
>
> - `mod` – operator modulo, służy do obliczania reszty z dzielenia,
> - `znak(a)` – wypisuje znak odpowiadający liczbie `a` w kodzie ASCII.

Odpowiedź do zadania

Poprawna odpowiedź dla danych z pliku `podpisy.txt` zapisana jest w pliku `wynik2.txt`.

Plik o rozmiarze 8.30 KB w języku polskim

> **Polecenie 3**
>
> Dodaj do napisanego przez siebie programu komentarze, aby nawet osoba, która nie potrafi programować, mogła go zrozumieć.

Sprawdź się

Zadanie 3

W pliku `słowa.txt` znajduje się 1000 wierszy. W każdym z nich znajduje się liczba całkowita z przedziału [1, 25], następnie oddzielone spacją słowo o długości od 2 do 20 znaków, złożone z liter alfabetu łacińskiego. Litery w słowie mogą być zarówno wielkie, jak i małe.

> **Przykład 1**
>
> Przykładowy fragment z pliku `słowa.txt`:

```
2 aiojK
1 KGUauGsF
3 aAaAaAaAaAaAaAaAaAaA
5 SB
4 OPsT
```

Plik o rozmiarze 15.38 KB w języku polskim

Napisz program, który każde słowo z pliku `słowa.txt` zaszyfruje w taki w sposób, że każdą literę wyrazu zamieni na literę znajdującą się w alfabecie na pozycji oddalonej od pozycji wyjściowej litery o liczbę podaną na początku wiersza. Przesunięć dokonujemy, idąc w prawą stronę. Wielkość liter powinna zostać zachowana. Wyniki zapisz do pliku `wynik3.txt`, każde słowo w oddzielnej linijce, zachowując kolejność ich występowania w oryginalnym pliku.

W sytuacji, gdy dotrzemy do końca alfabetu, wracamy na jego początek – czyli jeżeli chcemy zamienić literę „z” na literę oddaloną o jedno miejsce w prawo, otrzymujemy „a”.

> **Przykład 2**
>
> Dla przykładowych pięciu linii z pliku `słowa.txt`:

```
2 aiojK
1 KGUauGsF
3 aAaAaAaAaAaAaAaAaAaA
5 SB
4 OPsT
```

> plik `wynik3.txt` powinien wyglądać następująco:

```
ckqlM
LHVbvHtG
dDdDdDdDdDdDdDdDdDdD
XG
STwX
```

Do oceny oddajesz:

- plik `wynik3.txt` zawierający odpowiedź do zadania (zaszyfrowane słowa, każde zapisane w osobnym wierszu),
- plik(i) z komputerową realizacją rozwiązania (kodem programu).

> Przedstaw rozwiązanie zadania w postaci programu w języku C++, Java lub Python. Odpowiedź do zadania dla danych z pliku znajdziesz pod sekcją ćwiczeń.

Pokaż ćwiczenia:

Ćwiczenie 1

JĘZYK C++

Twoje zadania

1. Program w każdej linii wypisuje słowo zaszyfrowane według warunków podanych w zadaniu.

Ćwiczenie 2

JĘZYK JAVA

Twoje zadania

1. Program w każdej linii wypisuje słowo zaszyfrowane według warunków podanych w zadaniu.

Ćwiczenie 3

JĘZYK PYTHON

Twoje zadania

1. Program w każdej linii wypisuje słowo zaszyfrowane według warunków podanych w zadaniu.

Odpowiedź do zadania

Poprawna odpowiedź dla danych z pliku `słowa.txt` zapisana jest w pliku `wynik3.txt`:

Plik o rozmiarze 12.82 KB w języku polskim

Dla nauczyciela

Autor: Maurycy Gast
Przedmiot: Informatyka

Temat: Szyfry symetryczne i asymetryczne – zadania maturalne

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

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

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

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:

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 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,

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;

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):

- Zastosujesz w praktyce swoją wiedzę na temat szyfrów symetrycznych i asymetrycznych.
- Rozwiążesz przykładowe zadania typu maturalnego z zakresu szyfrów symetrycznych i asymetrycznych.
- Powtórzysz wiadomości dotyczące szyfrów symetrycznych i asymetrycznych.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

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

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;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Szyfry symetryczne i asymetryczne – zadania maturalne". Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj".

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć zawarte w sekcji „Wprowadzenie". Prosi uczniów, by na podstawie wiadomości zdobytych przed lekcją zaproponowali kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji oraz w odniesieniu do poznanych języków programowania.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie w parach analizują zadanie 1 z sekcji „Przeczytaj". Następnie implementują algorytm w wybranym języku programowania. Porównują swoje rozwiązanie z programem innej grupy.
2. **Praca z multimedium.** Wybrany uczeń czyta treść zadania nr 2 z sekcji „Prezentacja multimedialna". Uczniowie wspólnie analizują przedstawione w formie pseudokodu rozwiązanie. Nauczyciel wyjaśnia niezrozumiałe kwestie.
3. **Ćwiczenia umiejętności.** Uczniowie indywidualnie wykonują zadanie nr 3 z sekcji „Sprawdź się". Ćwiczenia realizują w jednym z dostępnych języków programowania. Nauczyciel prosi wybrane osoby o zaprezentowanie rozwiązania.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie". W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się".

Praca domowa:

1. Uczniowie zapisują rozwiązanie zadania z sekcji „Prezentacja multimedialna" w wybranym języku programowania.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Prezentację wykorzystać można do dyskusji na temat tego, jak działa podpis cyfrowy.