Pierwsza Ławka · zestaw działowy · liceum i technikum

Dział 4

# Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW

Badanie pierwszości liczby, zamiana reprezentacji liczb między systemami pozycyjnymi (2, 8, 10, 16 i inne), algorytm Euklidesa, NWW i działania na ułamkach – na papierze i w Pythonie. Osiem kart: teoria z przykładami, ćwiczenia, utrwalenie w formule CKE, zadanie przy komputerze z plikami danych (ZIP), kartkówka i sprawdzian w grupach A i B – z kluczem i zasadami oceniania.

Informatyka | klasa 2 LO i technikum | Dział 4 | zakres podstawowy (ZP)

| | Karta | Rodzaj | Poziom | Zadania |
|---|---|---|---|---|
| 1 | **Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – najważniejsze wiadomości**<br>Teoria, przykłady krok po kroku, rozgrzewka | Najważniejsze wiadomości | ZP | 4 zad. · 8 pkt |
| 2 | **Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – ćwiczenia podstawowe**<br>Ćwiczę umiejętności z działu | Ćwiczenia | ZP | 7 zad. · 21 pkt |
| 3 | **Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – utrwalenie**<br>Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych | Zadania typu maturalnego | ZP | 6 zad. · 11 pkt |
| 4 | **Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – w praktyce, przy komputerze**<br>Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt | Ćwiczenia | ZP | 2 zad. · 13 pkt |
| 5 | **Kartkówka: systemy pozycyjne i liczby pierwsze – grupa A**<br>15 minut · 8 pkt · bez komputera | Kartkówka | ZP | 4 zad. · 8 pkt<br>15 min |
| 6 | **Kartkówka: systemy pozycyjne i liczby pierwsze – grupa B**<br>15 minut · 8 pkt · bez komputera | Kartkówka | ZP | 4 zad. · 8 pkt<br>15 min |
| 7 | **Sprawdzian: Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – grupa A**<br>45 minut · 20 pkt · bez komputera | Sprawdzian | ZP | 9 zad. · 20 pkt<br>45 min |
| 8 | **Sprawdzian: Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – grupa B**<br>45 minut · 20 pkt · bez komputera | Sprawdzian | ZP | 9 zad. · 20 pkt<br>45 min |

PIERWSZA ŁAWKA

## Jak korzystać z zestawu

- Karta 1 przypomina najważniejsze wiadomości i pokazuje przykłady rozwiązane krok po kroku (z kodem w Pythonie albo opisem kroków w aplikacji) – od niej warto zacząć.
- Karta 2 to ćwiczenia podstawowe, karta 3 – utrwalenie w formule zadań maturalnych CKE (ramka „Jak to sprawdza CKE").
- Karta 4 to praca przy komputerze: pliki danych są w archiwum ZIP dołączonym do zestawu, a odpowiedzi kontrolne dla plików *_przyklad (pliki o nazwie z dopiskiem „_przyklad") podano przy zadaniach. Wyniki zapisuje się w plikach wynikiN.txt – jak na maturze.
- Karty 5–6 to kartkówka (15 minut), karty 7–8 – sprawdzian (45 minut), obie w grupach A i B: te same typy zadań i ta sama punktacja, inne dane. Kartkówka i sprawdzian nie wymagają komputera.
- Klucz zawiera odpowiedzi, przykładowe programy, zasady oceniania w stylu CKE i numery wymagań podstawy. Osobny plik „Powtórka" zbiera teorię, typowe błędy i test „Sprawdź się".

## Oznaczenia

| | |
|---|---|
| ZP | zakres podstawowy |
| ZR | zakres rozszerzony (treści tylko dla ZR) |
| **(0–2 pkt)** | liczba punktów możliwych do zdobycia za zadanie |
| **Zadanie 4.1.** | wiązka zadań do wspólnego materiału źródłowego |
| **Jak to sprawdza CKE** | typ zadania i typowy błąd ze sprawozdań CKE |
| **Na maturze** | wymagania z podstawy programowej, których dotyczy temat |

## Wymagania z podstawy programowej (Dz.U. 2024 poz. 1019)

- **I.2a** stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: badania pierwszości liczby, zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi, działań na ułamkach z wykorzystaniem NWD i NWW,
- **I.1** planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
- **I.3** sprawdza poprawność działania algorytmów dla przykładowych danych.
- **II.1** projektuje i programuje rozwiązania problemów z różnych dziedzin, stosując: 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;

Materiał oryginalny – żadne zadanie nie jest kopią zadania CKE ani podręcznika; dane w plikach są fikcyjne. Podstawa programowa LO i technikum w brzmieniu z 2024 r. (Dz.U. 2024 poz. 1019). Wymagania: I.2a, I.1, I.3, II.1. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZP – część II, rozdz. „Algorytmika i programowanie" (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

# Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – najważniejsze wiadomości

Teoria, przykłady krok po kroku, rozgrzewka

ZP </> KARTA 1/8

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 8 pkt

Najważniejsze wiadomości · 8 pkt

## Przypomnij sobie

- **Liczba pierwsza** to liczba naturalna większa od 1, która ma dokładnie dwa dzielniki: 1 i samą siebie. Liczby 0 i 1 nie są ani pierwsze, ani złożone. Liczba złożona n ma dzielnik d, dla którego $2 \le d \le \sqrt{n}$ – bo jeśli $n = a \cdot b$ i $a \le b$, to $a \cdot a \le n$.
- **Test pierwszości**: sprawdzamy kolejne d = 2, 3, 4, … dopóki $d \cdot d \le n$. Jeśli któraś reszta n % d jest równa 0 – liczba jest złożona; jeśli żadna – pierwsza. Warunek $d \cdot d \le n$ (a nie $d < n$) skraca pracę: dla n = 1 000 003 wystarczy 999 sprawdzeń zamiast prawie miliona.
- **System pozycyjny o podstawie p** ma cyfry 0, 1, …, p − 1 (powyżej 9: A = 10, B = 11, …, F = 15). Wartość zapisu $c_k \ldots c_1 c_0$ to $c_k \cdot p^k + \ldots + c_1 \cdot p + c_0$, np. $1101_2 = 8 + 4 + 0 + 1 = 13$, $2F_{16} = 2 \cdot 16 + 15 = 47$.
- **Z systemu p na dziesiętny** – metoda kolejnych mnożeń: w = 0, a potem dla każdej cyfry od lewej w = w · p + cyfra. **Z dziesiętnego na system p**: dzielimy n przez p z resztą, zapisujemy resztę i zastępujemy n ilorazem, aż iloraz będzie równy 0; reszty czytamy **od ostatniej do pierwszej**.
- **Systemy 2, 8 i 16**: jedna cyfra ósemkowa to 3 bity, jedna szesnastkowa – 4 bity, np. $1100\ 1000_2 = C8_{16}$, $11\ 001\ 000_2 = 310_8$. Potęgi dwójki: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024.
- **NWD – algorytm Euklidesa** (znany ze szkoły podstawowej): dopóki b ≠ 0, zastępujemy parę (a, b) parą (b, a mod b); gdy b = 0, NWD = a. **NWW**(a, b) = a · b / NWD(a, b) – w programie lepiej a // NWD(a, b) * b (mniejsze liczby po drodze).
- **Ułamki**: ułamek a/b skracamy, dzieląc licznik i mianownik przez NWD(a, b) – wtedy jest nieskracalny. Przy dodawaniu a/b + c/d wspólnym mianownikiem jest NWW(b, d); wynik na końcu skracamy.
- **W Pythonie**: % – reszta z dzielenia, // – dzielenie całkowite, int('2F', 16) – zamiana zapisu w systemie 16 na liczbę, bin(n) – zapis dwójkowy z przedrostkiem 0b, math.gcd – NWD. Przy komputerze wolno z nich korzystać, ale gdy polecenie każe „zapisać algorytm” i zabrania funkcji wbudowanych (częste na maturze), piszesz pętlę sam.
- **Testowanie (wymaganie I.3)**: sprawdzaj program na przykładzie z treści i na przypadkach brzegowych: n = 0, 1, 2, kwadrat liczby pierwszej (49, 169), duża liczba pierwsza, liczba 0 przy zamianie systemu.

PIERWSZA ŁAWKA

## Przykłady z rozwiązaniem

**Przykład 1.** **Czy 221 jest liczbą pierwszą? Ile reszt obliczy funkcja pierwsza(n)?**

```
def pierwsza(n):
    if n < 2:
        return False
    d = 2
    while d * d <= n:
        if n % d == 0:
            return False
        d = d + 1
    return True
```

Sprawdzamy d = 2, 3, …, 14 (14 · 14 = 196 ≤ 221, ale 15 · 15 = 225 > 221). Dla d = 13: 221 % 13 = 0, bo 221 = 13 · 17 – funkcja zwraca False po 12 resztach (d = 2, …, 13). Liczba 221 jest złożona.

**Przykład 2.** **Zamień $1A3_{16}$ na system dziesiętny, a wynik – na dwójkowy i ósemkowy.**

Kolejne mnożenia: w = 1; w = 1 · 16 + 10 = 26; w = 26 · 16 + 3 = 419.
419 : 2 = 209 r. 1; 209 : 2 = 104 r. 1; 104 : 2 = 52 r. 0; 52 : 2 = 26 r. 0; 26 : 2 = 13 r. 0; 13 : 2 = 6 r. 1; 6 : 2 = 3 r. 0; 3 : 2 = 1 r. 1; 1 : 2 = 0 r. 1.
Reszty od ostatniej: $110100011_2$. Grupy po 3 bity od prawej: 110 100 011 → $643_8$.

**Przykład 3.** **Oblicz NWD i NWW liczb 462 i 1078, a potem skróć ułamek 462/1078.**

Euklides: (462, 1078) → (1078, 462) → (462, 154), bo 1078 mod 462 = 154 → (154, 0), bo 462 mod 154 = 0. NWD = 154.
NWW = 462 // 154 · 1078 = 3 · 1078 = 3234. Ułamek: 462/1078 = (462 : 154)/(1078 : 154) = 3/7.

**Przykład 4.** **Dodaj ułamki 5/14 + 4/21 tak, jak robi to program.**

NWD(14, 21) = 7, więc NWW(14, 21) = 14 // 7 · 21 = 42. 5/14 = 15/42, 4/21 = 8/42, suma 23/42; NWD(23, 42) = 1 – ułamek jest nieskracalny.

```
def nwd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

def dodaj(a, b, c, d):          # a/b + c/d
    m = b // nwd(b, d) * d      # NWW(b, d)
    l = a * (m // b) + c * (m // d)
    g = nwd(l, m)
    return l // g, m // g

print(dodaj(5, 14, 4, 21))      # (23, 42)
```

## Zadania

**Zadanie 1.** (0–3 pkt) ..... / 3

Uzupełnij.

1. 45 = _____$_2$  2. $1111_2$ = _____  3. $2F_{16}$ = _____

**Zadanie 2.** (0–1 pkt) ..... / 1

Która z liczb jest liczbą pierwszą?

A. 87  B. 97  C. 91  D. 51

# Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – ćwiczenia podstawowe

Ćwiczę umiejętności z działu

ZP </> KARTA 2/8

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 21 pkt

Ćwiczenia · 21 pkt

## Zadanie 1. (0–3 pkt) ..... / 3

Wykonaj algorytm Euklidesa dla a = 270 i b = 192. Uzupełnij tabelę i podaj NWD(270, 192).

| a | b | a mod b |
|---|---|---|
| 270 | 192 | |
| | | |
| | | |
| | | 0 |

Miejsce na odpowiedź

## Zadanie 2. (0–4 pkt) ..... / 4

Zamień liczby.

1. $200 = _____{}_{2}$
2. $200 = _____{}_{8}$
3. $200 = _____{}_{16}$
4. $2101_3 = _____$

## Zadanie 3. (0–4 pkt) ..... / 4

Połącz zapis liczby z jej wartością dziesiętną. Jedna wartość jest zbędna.

1. $1100100_2$
2. $75_8$
3. $3B_{16}$
4. $1210_3$

a) 48
b) 59
c) 61
d) 64
e) 100

1 ____ 2 ____ 3 ____ 4 ____

## Zadanie 4. (0–3 pkt) ..... / 3

Wykonaj działania. Wynik podaj w postaci ułamka nieskracalnego.

1. 84/126 = _____
2. 7/15 + 4/9 = _____
3. 5/6 − 3/10 = _____

# Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – utrwalenie

ZP </> KARTA 3/8

Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 11 pkt

Zadania typu maturalnego · 11 pkt

**Jak to sprawdza CKE**

Na maturze z informatyki (poziom rozszerzony) te umiejętności wracają co roku: systemy pozycyjne bez komputera (2023 zad. 2 „Liczby binarne" – zapis algorytmu, 3 pkt; 2026 zad. 5 – systemy piątkowy i trójkowy, 2 pkt), liczby pierwsze i dzielniki przy komputerze (2024 zad. 4 „Liczby", 10 pkt). Zapis algorytmu zawsze ze *Specyfikacją* (Dane, Wynik) i z *Uwagą*, czego nie wolno używać – sprawozdania CKE 2023–2026: zdający często pomijają tę uwagę. Zadanie P/F – 1 pkt za komplet poprawnych ocen.

## Zadanie 1. (0–1 pkt) ..... / 1

Oceń prawdziwość podanych zdań. Zaznacz P, jeśli zdanie jest prawdziwe, albo F – jeśli jest fałszywe.

```
def pierwsza(n):
    if n < 2:
        return False
    d = 2
    while d * d <= n:
        if n % d == 0:
            return False
        d = d + 1
    return True
```

| | | |
|---|---|---|
| 1. Liczba $1000000_2$ jest większa od liczby $77_8$. | P | F |
| 2. Zapis dziesiętnej liczby 256 w systemie ósemkowym ma cztery cyfry. | P | F |
| 3. Aby stwierdzić, że 323 nie jest liczbą pierwszą, funkcja pierwsza(n) obliczy 16 reszt z dzielenia. | P | F |

## Zadanie 2. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Suma $10110_2 + 1101_2$ jest równa

A. $100011_2$ B. $11011_2$ C. $100001_2$ D. $100101_2$

## Zadanie 3. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Najmniejszą liczbą pierwszą większą od 200 jest

A. 203 B. 201 C. 211 D. 209

**Zadanie 4.** (0–3 pkt) ..... / 3

Przeanalizuj funkcję f(a, b). Uzupełnij tabelę – podaj wynik funkcji (a) oraz wartość k.

```
def f(a, b):
    k = 0
    while b != 0:
        a, b = b, a % b
        k = k + 1
    return a, k
```

| a | b | zwrócone a | k |
|---|---|---|---|
| 91 | 35 | | |
| 120 | 84 | | |
| 13 | 34 | | |

**Zadanie 5.** (0–3 pkt) ..... / 3

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który dla danych liczb n i p obliczy sumę cyfr zapisu liczby n w systemie o podstawie p.

**Przykład:** dla n = 100 i p = 3 wynikiem jest 4, bo $100 = 10201_3$.

**Uwaga:** Twój algorytm może używać tylko zmiennych przechowujących pojedyncze liczby całkowite, operacji arytmetycznych (dodawanie, odejmowanie, mnożenie, dzielenie całkowite i reszta z dzielenia), porównań, operatorów logicznych, instrukcji warunkowych, pętli, przypisań i własnych funkcji. Nie wolno używać funkcji wbudowanych, tablic ani list ani zamieniać liczby na napis.

**Specyfikacja**
**Dane:** n – liczba całkowita, $n \geq 1$; p – liczba całkowita, $2 \leq p \leq 10$
**Wynik:** s – suma cyfr zapisu liczby n w systemie o podstawie p

Miejsce na odpowiedź

PIERWSZA ŁAWKA

# Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – w praktyce, przy komputerze

ZP </> KARTA 4/8

Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 13 pkt

Ćwiczenia · 13 pkt

## Zadanie 1. (0–7 pkt)

Etykiety w bibliotece

**Opis danych**

Każda książka w szkolnej bibliotece ma etykietę z numerem zapisanym w systemie szesnastkowym (cyfry 0–9 i wielkie litery A–F, bez zer wiodących). Plik **etykiety.txt** zawiera 500 wierszy – w każdym jeden numer o wartości od 16 do 65 535. Pierwsze wiersze pliku:

```
F6D8
1EDB
8A1E
B77E
```

Plik **etykiety_przyklad.txt** zawiera 20 numerów w tym samym formacie.

Do dyspozycji masz plik **etykiety.txt** oraz mniejszy plik **etykiety_przyklad.txt**. Odpowiedzi dla danych przykładowych podano przy zadaniach – najpierw sprawdź na nich swój program.

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi zapisz w pliku **wyniki1.txt**, a każdą z nich poprzedź numerem odpowiedniego zadania.

Do oceny oddajesz: plik tekstowy **wyniki1.txt** z odpowiedziami do zadań 1.1–1.3 (każda poprzedzona numerem zadania) oraz pliki z kodem źródłowym programów, np. **zadanie1.py**.

*Dane fikcyjne, przygotowane do zadania.*

### Zadanie 1.1. (0–2 pkt) ..... / 2

Podaj, ile numerów z pliku (po zamianie na system dziesiętny) jest liczbami pierwszymi. *Dla danych z pliku przykładowego odpowiedź: 9.*

Odpowiedź zapisz w pliku wyniki1.txt

### Zadanie 1.2. (0–2 pkt) ..... / 2

Podaj numer o największej wartości – w zapisie szesnastkowym (tak jak w pliku) i w zapisie dwójkowym; oba zapisy oddziel spacją. *Dla danych z pliku przykładowego odpowiedź: DD0D 1101110100001101.*

Odpowiedź zapisz w pliku wyniki1.txt

### Zadanie 1.3. (0–3 pkt) ..... / 3

Podaj, ile numerów ma w zapisie dwójkowym (bez zer wiodących) więcej jedynek niż zer. *Dla danych z pliku przykładowego odpowiedź: 11.*

Odpowiedź zapisz w pliku wyniki1.txt

PIERWSZA ŁAWKA

## Zadanie 2. (0–6 pkt)

Proporcje

**Opis danych**

Plik **proporcje.txt** zawiera 300 wierszy. W każdym są dwie liczby całkowite l i m ($1 \le l < m \le 999$) oddzielone spacją – licznik i mianownik ułamka l/m. Pierwsze wiersze pliku:

```
330 575
428 754
5 33
```

Plik **proporcje_przyklad.txt** zawiera 15 wierszy w tym samym formacie.

Do dyspozycji masz plik **proporcje.txt** oraz mniejszy plik **proporcje_przyklad.txt**. Odpowiedzi dla danych przykładowych podano przy zadaniach – najpierw sprawdź na nich swój program.

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi zapisz w pliku **wyniki2.txt**, a każdą z nich poprzedź numerem odpowiedniego zadania.

Do oceny oddajesz: plik tekstowy **wyniki2.txt** z odpowiedziami do zadań 2.1–2.3 (każda poprzedzona numerem zadania) oraz pliki z kodem źródłowym programów, np. **zadanie2.py**.

*Dane fikcyjne, przygotowane do zadania.*

### Zadanie 2.1. (0–1 pkt) ..... / 1

Podaj, ile ułamków z pliku jest nieskracalnych. *Dla danych z pliku przykładowego odpowiedź: 8.*

Odpowiedź zapisz w pliku wyniki2.txt

### Zadanie 2.2. (0–2 pkt) ..... / 2

Podaj ułamek o największej wartości. Zapisz go w postaci nieskracalnej l/m. *Dla danych z pliku przykładowego odpowiedź: 29/30.*

Odpowiedź zapisz w pliku wyniki2.txt

### Zadanie 2.3. (0–3 pkt) ..... / 3

Oblicz sumę wszystkich ułamków z pliku, których mianownik jest dzielnikiem liczby 360. Wynik zapisz w postaci ułamka nieskracalnego l/m. *Dla danych z pliku przykładowego odpowiedź: 233/120.*

Odpowiedź zapisz w pliku wyniki2.txt

# Kartkówka: systemy pozycyjne i liczby pierwsze – grupa A

15 minut · 8 pkt · bez komputera

ZP </> KARTA 5/8

Imię i nazwisko: Klasa: Data: Wynik: ..... / 8 pkt

Kartkówka · 15 minut · 8 pkt

## Zadanie 1. (0–2 pkt) ..... / 2

Zamień liczby.

1. $77 = _____{}_2$

2. $1011011_2 = _____$

## Zadanie 2. (0–2 pkt) ..... / 2

Zamień liczby.

1. $3C_{16} = _____$

2. $150 = _____{}_8$

## Zadanie 3. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.

Liczbą pierwszą jest

A. 133 B. 111 C. 119 D. 127

## Zadanie 4. (0–3 pkt) ..... / 3

Przeanalizuj funkcję pierwsza(n). Uzupełnij tabelę: podaj, ile razy funkcja obliczy resztę n % d, oraz wynik funkcji.

```
def pierwsza(n):
    if n < 2:
        return False
    d = 2
    while d * d <= n:
        if n % d == 0:
            return False
        d = d + 1
    return True
```

| n | liczba obliczonych reszt | wynik |
|---|---|---|
| 169 | | |
| 101 | | |

CO JESZCZE JEST W ŚRODKU

# Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 2 LO i...

Podgląd pokazał 10 z 31 stron. Pełna wersja: 31 stron A4, PDF do druku i DOCX do edycji.

- Karta 1: Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – najważniejsze wiadomości – 4 zadania, 8 pkt
- Karta 2: Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – ćwiczenia podstawowe – 7 zadań, 21 pkt
- Karta 3: Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – utrwalenie – 6 zadań, 11 pkt
- Karta 4: Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – w praktyce, przy komputerze – 2 zadania, 13 pkt
- Karta 5: Kartkówka: systemy pozycyjne i liczby pierwsze – grupa A – 4 zadania, 8 pkt
- Karta 6: Kartkówka: systemy pozycyjne i liczby pierwsze – grupa B – 4 zadania, 8 pkt
- Karta 7: Sprawdzian: Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – grupa A – 9 zadań, 20 pkt
- Karta 8: Sprawdzian: Algorytmy na liczbach: liczby pierwsze, systemy pozycyjne, NWD i NWW – grupa B – 9 zadań, 20 pkt Karta 1 przypomina teorię i pokazuje przykłady rozwiązane krok

Kup pełną wersję – plik od razu na e-mail

pierwszalawka.pl/material/informatyka-liceum-klasa-2-dzial-04-algorytmy-na-liczbach-liczby-pierwsze-systemy