8 **Sprawdzian: Algorytmy na tekstach: porównywanie, wyszukiwanie wzorca, szyfr Cezara – grupa B**
45 minut · 20 pkt · bez komputera

Sprawdzian | ZP | 8 zad. · 20 pkt · 45 min

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.2b** 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.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.2b, I.3, II.1. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZP – część III, rozdz. „Algorytmika i programowanie" (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

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.2c** stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: c) porządkowania ciągu liczb przez wstawianie i metodą bąbelkową,
- **I.2d** stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: d) obliczania wartości elementów ciągu metodą iteracyjną w tym wartości elementów ciągu Fibonacciego;
- **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.2c, I.2d, I.3, II.1. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZP – część III, rozdz. „Algorytmika i programowanie" (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

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)

- **IV.1** aktywnie uczestniczy w realizacji projektów rozwiązujących problemy z różnych dziedzin;
- **II.2** do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki;
- **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.

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: IV.1, II.2, I.1, I.3. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZP – część III, rozdz. „Algorytmika i programowanie" (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

Algorytmy na tekstach: porównywanie, wyszukiwanie wzorca, szyfr Cezara – ćwiczenia podstawowe

ZP </> KARTA 2/8

Ćwiczę umiejętności z działu

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 16 pkt

Ćwiczenia · 16 pkt

Zadanie 1. (0–2 pkt) ..... / 2

Metoda naiwna szuka wzorca w = "KAK" w tekście t = "KAKAKO". Uzupełnij tabelę: wpisz fragment tekstu przyłożony do wzorca na pozycji i oraz to, czy jest zgodny ze wzorcem (tak/nie).

i	fragment t[i..i+2]	zgodny?
0		
1		
2		
3		

Zadanie 2. (0–2 pkt) ..... / 2

Szyfr Cezara.

1. SZKOLA zaszyfrowane kluczem k = 5: _____
2. HFJXFW odszyfrowane (klucz szyfrowania k = 5): _____

Zadanie 3. (0–4 pkt) ..... / 4

Oceń prawdziwość zdań o porównywaniu napisów w Pythonie.

1. "KOT" < "KOTEK"	P	F
2. "Kot" < "kot"	P	F
3. "kra" < "kot"	P	F
4. "10" < "9"	P	F

PIERWSZA ŁAWKA

Przykład 2. **Posortuj przez wstawianie ciąg 4, 1, 3, 2. Zapisz stan po każdym wstawieniu.**

i = 1: x = 1, przesuwamy 4 → 1, 4, 3, 2. i = 2: x = 3, przesuwamy 4 → 1, 3, 4, 2. i = 3: x = 2, przesuwamy 4 i 3 → 1, 2, 3, 4. Razem 4 przesunięcia.

```
def wstawianie(a):
    for i in range(1, len(a)):
        x = a[i]
        j = i - 1
        while j >= 0 and a[j] > x:
            a[j + 1] = a[j]
            j = j - 1
        a[j + 1] = x
    return a
```

Przykład 3. **Oblicz iteracyjnie F_{10}.**

krok	1	2	3	4	5	6	7	8	9	10
a	1	1	2	3	5	8	13	21	34	55
b	1	2	3	5	8	13	21	34	55	89

Start: a = 0, b = 1; po 10 krokach a = F_{10} = 55.

```
def fib(n):
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
    return a

print(fib(10))   # 55
```

Zadania

Zadanie 1. (0–3 pkt) ..... / 3

Uzupełnij wyrazy ciągu Fibonacciego.

1. F_7 = _____ 2. F_8 = _____ 3. F_{12} = _____

Zadanie 2. (0–1 pkt) ..... / 1

Po pierwszym przebiegu sortowania bąbelkowego ciągu 6, 3, 8, 2 otrzymamy

A. 3, 2, 6, 8 B. 3, 6, 2, 8 C. 3, 6, 8, 2 D. 2, 3, 6, 8

Zadanie 3. (0–3 pkt) ..... / 3

Ciąg: $a_1 = 4$, $a_{n+1} = 3 \cdot a_n - 2$. Uzupełnij.

1. a_2 = _____ 2. a_3 = _____ 3. a_4 = _____

Zadanie 4. (0–2 pkt) ..... / 2

Oceń prawdziwość zdań.

1. W ciągu 3, 1, 2 są dokładnie 2 inwersje.	P	F
2. Sortowanie przez wstawianie ciągu już uporządkowanego nie wykonuje żadnego porównania.	P	F

Zadanie 5. (0–3 pkt) ..... / 3

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który obliczy pole końcowe robota. Robot zaczyna na (0, 0), zwrócony na północ (w kierunku rosnących y). Polecenia: P – jedno pole naprzód, L – obrót o 90° w lewo, R – obrót o 90° w prawo.

Przykład: dla PPRPLP wynikiem jest (1, 3).

Specyfikacja

Dane: n – liczba poleceń; p[0], p[1], …, p[n − 1] – polecenia (znaki P, L, R)

Wynik: x, y – współrzędne pola końcowego

Miejsce na odpowiedź

Zadanie 6. (0–3 pkt) ..... / 3

Zespół testuje robota, który ma zatrzymać się 10 cm przed ścianą. W 3 z 10 prób robot uderzył w ścianę przy szybkiej jeździe. Zaproponuj dwie zmiany (w programie lub w konstrukcji) i opisz, jak sprawdzić, czy pomogły.

Miejsce na odpowiedź

PIERWSZA ŁAWKA

PIERWSZA ŁAWKA · POWTÓRKA · INFORMATYKA · KLASA 3 LO I TECHNIKUM · ZAKRES PODSTAWOWY · ALGORYTMY NA TEKSTACH: PORÓWNYWANIE, WYSZUKIWANIE WZORCA, SZYFR CEZARA

Algorytmy na tekstach – powtórka

Porównywanie napisów, wyszukiwanie wzorca metodą naiwną i szyfr Cezara – z kodem w Pythonie, typowymi błędami i testem „Sprawdź się". Klasa 3 LO i technikum, zakres podstawowy.

1. Kody znaków i porównywanie napisów

- ord("A") = 65 … ord("Z") = 90, ord("a") = 97 … ord("z") = 122, ord("0") = 48; chr(66) = "B".
- Porządek leksykograficzny: decyduje pierwsza różnica; jeśli jeden napis jest początkiem drugiego – krótszy jest mniejszy.
- "Zebra" < "ala" (90 < 97); "10" < "9" (porównujemy znaki, nie liczby).

```
def porownaj(a, b):
    i = 0
    while i < len(a) and i < len(b):
        if a[i] != b[i]:
            if ord(a[i]) < ord(b[i]):
                return -1
            return 1
        i = i + 1
    if len(a) < len(b):
        return -1
    if len(a) > len(b):
        return 1
    return 0
```

2. Wyszukiwanie wzorca metodą naiwną

- Wzorzec długości m przykładamy do pozycji 0, 1, …, n − m tekstu długości n – razem n − m + 1 pozycji.
- Wystąpienia mogą zachodzić na siebie: "ANA" w "ANANAS" – pozycje 0 i 2; "ANANAS".count("ANA") = 1.

```
def szukaj(t, w):
    wynik = []
    for i in range(len(t) - len(w) + 1):
        j = 0
        while j < len(w) and t[i + j] == w[j]:
            j = j + 1
        if j == len(w):
            wynik.append(i)
    return wynik
```

PIERWSZA ŁAWKA · POWTÓRKA · INFORMATYKA · KLASA 3 LO I TECHNIKUM · ZAKRES PODSTAWOWY · SORTOWANIE (WSTAWIANIE, BĄBELKOWE) I CIĄGI LICZONE ITERACYJNIE (FIBONACCI)

Sortowanie i ciągi iteracyjne – powtórka

Sortowanie bąbelkowe i przez wstawianie, ciąg Fibonacciego i inne ciągi liczone w pętli – z kodem w Pythonie, typowymi błędami i testem „Sprawdź się”. Klasa 3 LO i technikum, zakres podstawowy.

1. Sortowanie bąbelkowe

- Porównuj sąsiadów, zamieniaj, gdy lewy jest większy. Po każdym przebiegu kolejny największy element jest na swoim miejscu.
- Porównań $n(n-1)/2$, zamian tyle, ile inwersji. Przykład 5, 2, 4, 1 → 6 porównań, 5 zamian.

```
def babelkowe(a):
    n = len(a)
    for i in range(n - 1):
        for j in range(n - 1 - i):
            if a[j] > a[j + 1]:
                a[j], a[j + 1] = a[j + 1], a[j]
    return a
```

2. Sortowanie przez wstawianie

- Lewa część jest uporządkowana; x = a[i] wstawiamy, przesuwając większe elementy w prawo.
- 4, 1, 3, 2 → 1, 4, 3, 2 → 1, 3, 4, 2 → 1, 2, 3, 4.

```
def wstawianie(a):
    for i in range(1, len(a)):
        x = a[i]
        j = i - 1
        while j >= 0 and a[j] > x:
            a[j + 1] = a[j]
            j = j - 1
        a[j + 1] = x
    return a
```

PIERWSZA ŁAWKA · POWTÓRKA · INFORMATYKA · KLASA 3 LO I TECHNIKUM · ZAKRES PODSTAWOWY · PROJEKT ZESPOŁOWY I ROBOTYKA

Projekt zespołowy i robotyka – powtórka

Etapy projektu zespołowego, harmonogram, budowa robota (czujniki, sterownik, elementy wykonawcze) i symulacja ruchu robota – z kodem w Pythonie, typowymi błędami i testem „Sprawdź się". Klasa 3 LO i technikum, zakres podstawowy.

1. Projekt krok po kroku

- Etapy: określenie problemu i celu → plan (zadania, podział pracy, harmonogram) → realizacja → testowanie → podsumowanie.
- Najkrótszy czas projektu = najdłuższy ciąg zadań zależnych; zadania niezależne wykonują różne osoby równocześnie.

Zadanie	Czas	Po zadaniu	Koniec
A	2	–	2
B	3	A	5
C	2	–	2
D	2	B, C	7

2. Budowa robota

- Czujniki (wejście) → sterownik (mikrokontroler, program) → elementy wykonawcze (wyjście).
- Czujnik ultradźwiękowy: s = v · t / 2, v ≈ 343 m/s. t = 2 ms → ok. 34 cm.

```
while True:
    odleglosc = zmierz_cm()          # odczyt czujnika (wejście)
    if odleglosc < 20:
        zatrzymaj()                      # element wykonawczy (wyjście)
        skrec_w_prawo()
    else:
        jedz_naprzod()
```

CO JESZCZE JEST W ŚRODKU

Informatyka, klasa 3 LO i technikum (zakres podstawowy) – pakiet 3 zestawów: kartkówki, sprawdziany, karty pracy i powtórki

Podgląd pokazał 9 z 89 stron. Pełna wersja: 89 stron A4, PDF do druku i DOCX do edycji.

- Algorytmy na tekstach: porównywanie, wyszukiwanie wzorca, szyfr Cezara – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 3 LO i...
- Sortowanie (wstawianie, bąbelkowe) i ciągi liczone iteracyjnie (Fibonacci) – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 3 LO...
- Projekt zespołowy i robotyka – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 3 LO i technikum (zakres podstawowy)

Kup pełną wersję – plik od razu na e-mail
pierwszalawka.pl/material/informatyka-liceum-klasa-3-zakres-podstawowy-pakiet