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+II.R3d** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: d) podejście zachłanne (do wydawania reszty, szukania najkrótszej drogi),
- **I+II.R3e** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: e) programowanie dynamiczne (do szukania najdłuższego wspólnego podciągu),
- **I+II.R2c** wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie,

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+II.R3d, I+II.R3e, I+II.R2c. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZR – algorytmika (metody zachłanne, programowanie dynamiczne) (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+II.R3g** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: g) struktury dynamiczne: stos, kolejka, lista (do realizacji algorytmu ONP),
- **I+II.R3h** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: h) grafy (do przedstawiania abstrakcyjnego modelu sytuacji problemowych).
- **I+II.R2d** wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: d) zamiany wyrażenia na postać w odwrotnej notacji polskiej i obliczanie jego wartości na podstawie tej postaci;

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+II.R3g, I+II.R3h, I+II.R2d. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZR – struktury danych (stos, kolejka, lista, ONP), grafy (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

PIERWSZA ŁAWKA

Wizualizacje, otwarte zasoby, myślenie komputacyjne – w praktyce, przy komputerze

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

ZR </> KARTA 4/8

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

Ćwiczenia · 13 pkt

Zadanie 1. (0–8 pkt)

Animacja piłki

Opis danych

Scena animacji ma 640 × 360 pikseli (początek układu w lewym górnym rogu, oś y w dół). Plik **ruch.txt** zawiera 480 wierszy – położenia środka piłki o promieniu 12 w kolejnych klatkach (klatki numerujemy od 0; wiersz 1 to klatka 0). W każdym wierszu są dwie liczby całkowite x i y. Piłka jest cała widoczna, gdy 12 ≤ x ≤ 628 i 12 ≤ y ≤ 348. Pierwsze wiersze pliku:

```
320 180
329 180
337 180
346 180
```

Plik **ruch_przyklad.txt** zawiera 48 klatek w tym samym formacie.

Do dyspozycji masz plik **ruch.txt** oraz mniejszy plik **ruch_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 1.1–1.4 (każda poprzedzona numerem zadania) oraz pliki z kodem źródłowym programów, np. **zadanie1.py**.

Dane fikcyjne, przygotowane do zadania.

Zadanie 1.1. (0–1 pkt) ..... / 1

Podaj, w ilu klatkach piłka jest cała widoczna. *Dla danych z pliku przykładowego odpowiedź: 18.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 1.2. (0–3 pkt) ..... / 3

Podaj numer pierwszej i ostatniej klatki najdłuższego nieprzerwanego fragmentu animacji, w którym piłka jest cała widoczna. Taki fragment jest jeden. *Dla danych z pliku przykładowego odpowiedź: 36 47.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 1.3. (0–2 pkt) ..... / 2

Podaj długość drogi przebytej przez środek piłki (suma odległości między położeniami w kolejnych klatkach) zaokrągloną do liczby całkowitej. *Dla danych z pliku przykładowego odpowiedź: 358.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 1.4. (0–2 pkt) ..... / 2

Napisz program, który zapisze plik **tor.svg** o wymiarach 640 × 360: prostokąt sceny (obrys), łamaną <polyline> przechodzącą przez położenia środka piłki ze wszystkich klatek oraz piłkę (koło o promieniu 12) w położeniu z klatki 0. W pliku wyniki1.txt zapisz trzy liczby oddzielone spacją: liczbę punktów łamanej oraz współrzędne x i y środka piłki. Tor może wychodzić poza scenę. Do oceny oddajesz plik tor.svg i plik z kodem. *Dla danych z pliku przykładowego odpowiedź: 48 590 300.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 4. (0–3 pkt) ..... / 3

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który wyznaczy długość najdłuższego spójnego podciągu niemalejącego (fragmentu złożonego z kolejnych wyrazów) ciągu A.

Przykład: dla A = [2, 2, 5, 3, 3, 4, 6, 1] wynikiem jest 4 (3, 3, 4, 6).

Uwaga: Nie używaj funkcji wbudowanych (np. max) ani dodatkowych tablic.

Specyfikacja

Dane: n – liczba całkowita, $n \geq 1$; A[1..n] – tablica liczb całkowitych

Wynik: naj – długość najdłuższego fragmentu A[i..j], w którym $A[i] \leq A[i + 1] \leq \ldots \leq A[j]$

Miejsce na odpowiedź

Zadanie 5. (0–2 pkt) ..... / 2

Automat wydaje żetony o wartościach 1, 7 i 10. Podaj najmniejszą kwotę, dla której algorytm zachłanny nie wyda najmniejszej liczby żetonów, oraz oba zestawy: zachłanny i optymalny.

Miejsce na odpowiedź

Zadanie 6. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.

Obliczenie długości najdłuższego wspólnego podciągu napisów o długościach m i n metodą programowania dynamicznego (tablica T) wymaga liczby operacji rzędu

A. $m + n$ B. 2^{m+n} C. $m \cdot n$ D. $\log(m \cdot n)$

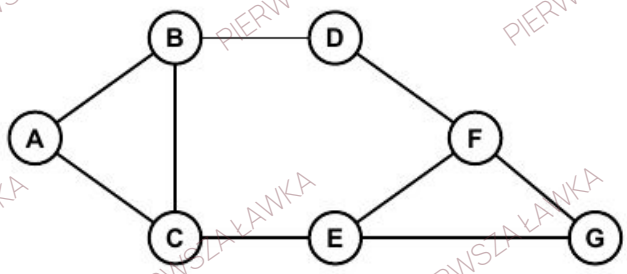
Zadanie 7. (0–2 pkt) ..... / 2

Podaj największą sumę spójnego fragmentu ciągu 4, −6, 3, 5, −2, 6, −9, 2 oraz ten fragment.

Miejsce na odpowiedź

Zadanie 4. (0–3 pkt) ..... / 3

Graf z rysunku przeszukujemy wszerz (BFS) od wierzchołka A; sąsiadów odwiedzamy w kolejności alfabetycznej. Uzupełnij tabelę: najmniejsza liczba krawędzi od A i miejsce wierzchołka w kolejności odwiedzania (A – 1).



wierzchołek	B	C	D	E	F	G
odległość od A						
miejsce w kolejności						

Zadanie 5. (0–4 pkt) ..... / 4

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który obliczy wartość poprawnego wyrażenia zapisanego w ONP. Użyj stosu zrealizowanego w tablicy.

Przykład: dla X = 7, 2, 3, *, +, 4, − wynikiem jest 9.

Uwaga: Stos realizujesz w tablicy S z licznikiem t; nie używaj funkcji eval ani gotowych struktur bibliotecznych.

Specyfikacja

Dane: n – liczba symboli, n ≥ 1; X[1..n] – symbole wyrażenia w ONP: liczby całkowite i operatory +, −, * (wyrażenie jest poprawne)

Wynik: wartość wyrażenia

Miejsce na odpowiedź

PIERWSZA ŁAWKA

Zadanie 7. (0–3 pkt) ..... / 3

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który dla obrazu w odcieniach szarości obliczy, ile pikseli ma jasność nie mniejszą od progu p.

Przykład: dla p = 128 i obrazu 2 × 3 o jasnościach 120, 30, 200 / 255, 90, 128 wynikiem jest 3

Uwaga: Nie używaj funkcji wbudowanych (np. max, count).

Specyfikacja

Dane: w, k, p – liczby całkowite dodatnie (p ≤ 255); J[1..w][1..k] – jasności pikseli od 0 do 255

Wynik: ile – liczba pikseli o jasności ≥ p

Miejsce na odpowiedź

Zadanie 8. (0–2 pkt) ..... / 2

Uczennica tworzy darmową, niekomercyjną stronę klubu szachowego. Chce użyć przerobionej ikony na CC BY-ND 4.0 i zdjęcia na CC BY-NC 4.0 bez zmian. Oceń, czy może użyć obu zasobów, i uzasadnij.

Miejsce na odpowiedź

Zadanie 9. (0–2 pkt) ..... / 2

Uczennica wybiera między informatyką a kierunkiem analiza danych. Podaj dwa źródła lub dokumenty, w których sprawdzi warunki przyjęcia, oraz dwie informacje, których w nich szuka.

Miejsce na odpowiedź

PIERWSZA ŁAWKA

PIERWSZA ŁAWKA · POWTÓRKA · INFORMATYKA · KLASA 4 LO I TECHNIKUM · ZAKRES ROZSZERZONY · PODEJŚCIE ZACHŁANNE I PROGRAMOWANIE DYNAMICZNE; PODCIĄGI

Podejście zachłanne, programowanie dynamiczne i podciągi – powtórka

Wydawanie reszty i najkrótsza droga (metoda zachłanna), programowanie dynamiczne (najmniej monet, najdłuższy wspólny podciąg), spójne podciągi o zadanej własności. Klasa 4 LO i technikum, zakres rozszerzony.

1. Podejście zachłanne

- W każdym kroku najlepszy wybór „na teraz”, bez cofania. Szybkie, ale poprawność trzeba uzasadnić.
- Wydawanie reszty: największa moneta ≤ kwota. Polskie monety – zawsze optymalnie. Nominały 1, 3, 4 i kwota 6: 4 + 1 + 1 zamiast 3 + 3.
- Dijkstra: zatwierdzamy wierzchołek o najmniejszym d, poprawiamy sąsiadów d[v] = min(d[v], d[u] + w). Wagi ≥ 0.

```
def reszta(kwota, nominaly):
    # nominaly uporządkowane malejąco, np. [500, 200, 100, 50, 20, 10, 5, 2, 1]
    wydane = []
    for m in nominaly:
        while kwota >= m:
            wydane.append(m)
            kwota = kwota - m
    return wydane
```

2. Programowanie dynamiczne

- Podproblemy się powtarzają → liczymy je raz, od najmniejszych, wyniki w tablicy.
- Reszta: ile[0] = 0, ile[k] = min(ile[k − m] + 1). Żetony 1, 4, 6: ile[0..10] = 0, 1, 2, 3, 1, 2, 1, 2, 2, 3, 2.
- NWP: T[i][j] = T[i−1][j−1] + 1 przy równych literach, inaczej max(T[i−1][j], T[i][j−1]). Czas rzędu m · n.

PIERWSZA ŁAWKA · POWTÓRKA · INFORMATYKA · KLASA 4 LO I TECHNIKUM · ZAKRES ROZSZERZONY · STOS, KOLEJKA, LISTA, ONP; GRAFY

Stos, kolejka, lista, ONP i grafy – powtórka

Struktury dynamiczne (stos, kolejka, lista), odwrotna notacja polska – obliczanie i zamiana, grafy jako model sytuacji problemowej i ich przeszukiwanie. Klasa 4 LO i technikum, zakres rozszerzony.

1. Stos, kolejka, lista

- Stos – LIFO: append / pop. Kolejka – FIFO: deque.append / deque.popleft (nie lista.pop(0) – wolne).
- Lista jednokierunkowa: wartość + wskazanie na następny; wstawienie = zmiana dwóch wskazań.
- Stos – ONP, DFS, cofanie zmian; kolejka – BFS, symulacje; lista – częste wstawianie w środek.

2. Odwrotna notacja polska

- Operator za argumentami: (2 + 3) · 4 → 2 3 + 4 *. Nawiasy niepotrzebne.
- Obliczanie: liczba na stos; operator zdejmuje b, potem a, odkłada a op b.
- Zamiana: liczby na wyjście; operator wypycha ze stosu operatory o priorytecie ≥; „(” na stos; „)” zdejmuje do „(”.
- Kod obliczania – funkcja wartosc_onp z karty 1 (przykład 1).

3. Grafy

- Suma stopni = 2 · liczba krawędzi. Graf pełny o n wierzchołkach ma n(n − 1)/2 krawędzi.
- Macierz sąsiedztwa: n^2 pól; listy sąsiedztwa: 2|E| wpisów (graf nieskierowany).
- BFS (kolejka) – najmniejsza liczba krawędzi; DFS (stos) – przejście w głąb; oba – spójne składowe.

PIERWSZA ŁAWKA · POWTÓRKA · INFORMATYKA · KLASA 4 LO I TECHNIKUM · ZAKRES ROZSZERZONY · WIZUALIZACJE, OTWARTE ZASOBY, MYŚLENIE KOMPUTACYJNE

Wizualizacje, otwarte zasoby, myślenie komputacyjne – powtórka

Grafika i animacja, SVG z programu, wykresy, licencje otwartych zasobów, myślenie komputacyjne, dalsza nauka. Klasa 4 LO i technikum, ZR.

1. Grafika i animacja

- Rozmiar bitmapy: szer. · wys. · bity / 8 bajtów. 1920 × 1080 × 24 bity = 6 220 800 B.

format	kompresja	animacja	do czego
PNG	bezstratna	nie (APNG – tak)	schematy, zrzuty, przezroczystość
JPEG	stratna	nie	zdjęcia
GIF	bezstratna, ≤ 256 kolorów	tak	proste animacje
SVG	– (wektor, XML)	możliwa	logo, ikony, wykresy

- Klatki = fps · czas. Interpolacja liniowa: $x(k) = x0 + (x1 - x0) \cdot k / n$.
- SVG: (0, 0) w lewym górnym rogu, y w dół; słupek na linii y = H ma y = H − h. Wykres: typ do celu, tytuł, opisy osi.

2. Otwarte zasoby i licencje

- BY – podaj autora; SA – utwór zależny na tej samej licencji; NC – tylko niekomercyjnie; ND – bez przeróbek; CC0 – bez warunków.
- Opis TASL: tytuł, autor, źródło (link), licencja (link).

3. Myślenie komputacyjne i dalsza droga

- Abstrakcja, reprezentacja danych, dekompozycja, redukcja, myślenie rekurencyjne, heurystyka – rozpoznawaj je w opisie rozwiązania.
- Matura z informatyki: 210 min, 50 pkt, praca przy komputerze. Zasady przyjęć – uchwała rekrutacyjna senatu wybranej uczelni.

CO JESZCZE JEST W ŚRODKU

Informatyka, klasa 4 LO i technikum (zakres rozszerzony) – pakiet 3 zestawów: kartkówki, sprawdziany, karty pracy i powtórki

Podgląd pokazał 9 z 95 stron. Pełna wersja: 95 stron A4, PDF do druku i DOCX do edycji.

- Podejście zachłanne i programowanie dynamiczne; podciągi – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 4 LO i technikum…
- Stos, kolejka, lista, ONP; grafy – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 4 LO i technikum (zakres rozszerzony)
- Wizualizacje, otwarte zasoby, myślenie komputacyjne – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 4 LO i technikum (zakres…

Kup pełną wersję – plik od razu na e-mail
pierwszalawka.pl/material/informatyka-liceum-klasa-4-zakres-rozszerzony-pakiet