Pierwsza Ławka · zestaw działowy · liceum i technikum

Dział 12

Stos, kolejka, lista, ONP; grafy

Struktury dynamiczne: stos, kolejka, lista; odwrotna notacja polska – zamiana wyrażenia i obliczanie wartości stosem; grafy jako model sytuacji problemowej, reprezentacje grafu, przeszukiwanie wszerz. 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 4 LO i technikum | Dział 12 | zakres rozszerzony (ZR)

	Karta	Rodzaj	Poziom	Zadania
1	**Stos, kolejka, lista, ONP; grafy – najważniejsze wiadomości** Teoria, przykłady krok po kroku, rozgrzewka	Najważniejsze wiadomości	ZR	4 zad. · 7 pkt
2	**Stos, kolejka, lista, ONP; grafy – ćwiczenia podstawowe** Ćwiczę umiejętności z działu	Ćwiczenia	ZR	8 zad. · 22 pkt
3	**Stos, kolejka, lista, ONP; grafy – utrwalenie** Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych	Zadania typu maturalnego	ZR	7 zad. · 13 pkt
4	**Stos, kolejka, lista, ONP; grafy – w praktyce, przy komputerze** Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt	Ćwiczenia	ZR	2 zad. · 14 pkt
5	**Kartkówka: stos, kolejka, ONP i grafy – grupa A** 15 minut · 8 pkt · bez komputera	Kartkówka	ZR	4 zad. · 8 pkt 15 min
6	**Kartkówka: stos, kolejka, ONP i grafy – grupa B** 15 minut · 8 pkt · bez komputera	Kartkówka	ZR	4 zad. · 8 pkt 15 min
7	**Sprawdzian: Stos, kolejka, lista, ONP; grafy – grupa A** 45 minut · 20 pkt · bez komputera	Sprawdzian	ZR	9 zad. · 20 pkt 45 min
8	**Sprawdzian: Stos, kolejka, lista, ONP; grafy – grupa B** 45 minut · 20 pkt · bez komputera	Sprawdzian	ZR	9 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+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

Stos, kolejka, lista, ONP; grafy – najważniejsze wiadomości

Teoria, przykłady krok po kroku, rozgrzewka

ZR </> KARTA 1/8

Imię i nazwisko: Klasa: Data: Wynik: ..... / 7 pkt

Najważniejsze wiadomości · 7 pkt

Przypomnij sobie

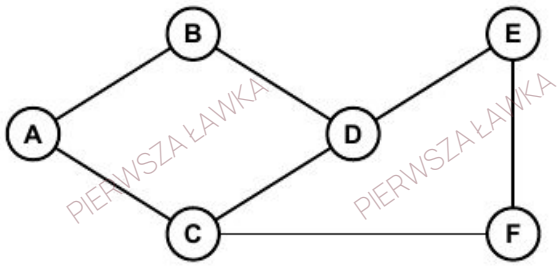
- **Stos** (ang. *stack*, LIFO – ostatni przyszedł, pierwszy wychodzi): wkładamy i zdejmujemy tylko na szczycie. W Pythonie lista: `stos.append(x)` – połóż, `stos.pop()` – zdejmij ze szczytu.
- **Kolejka** (ang. *queue*, FIFO – pierwszy przyszedł, pierwszy wychodzi): dołączamy na końcu, obsługujemy z początku. W Pythonie `collections.deque`: `append` i `popleft`. `lista.pop(0)` jest wolne – przesuwa wszystkie pozostałe elementy.
- **Lista (jednokierunkowa)**: każdy element przechowuje wartość i wskazanie na następny; zaczynamy od „głowy". Wstawienie za znanym elementem to ustawienie dwóch wskazań (nowy → następny, poprzedni → nowy), usunięcie – zmiana jednego wskazania, ale dostęp do k-tego elementu wymaga przejścia k kroków. (Lista Pythona to tablica dynamiczna – szybki dostęp po indeksie.)
- **Odwrotna notacja polska (ONP)** – zapis postfiksowy: operator stoi **za** argumentami, nawiasy są zbędne, np. (2 + 3) · 4 → `2 3 + 4 *`. To odwrócenie notacji polskiej Jana Łukasiewicza (operator przed argumentami).
- **Obliczanie ONP stosem**: czytamy symbole od lewej; liczbę kładziemy na stos; przy operatorze zdejmujemy b (szczyt), potem a i kładziemy a op b. Na końcu na stosie zostaje jedna liczba – wynik. Kolejność ma znaczenie przy − i /.
- **Zamiana na ONP** (algorytm „stacji rozrządowej" E. Dijkstry): liczby od razu na wyjście; operator – najpierw przenosimy na wyjście operatory ze stosu o priorytecie ≥ (· i / przed + i −), potem kładziemy operator; „(" na stos; „)" – przenosimy operatory do „(", a nawias usuwamy; na końcu opróżniamy stos.
- **Graf** G = (V, E): wierzchołki i krawędzie; nieskierowany lub skierowany, z wagami lub bez. **Stopień** wierzchołka – liczba krawędzi, które z niego wychodzą. W grafie nieskierowanym suma stopni = 2 · liczba krawędzi, więc wierzchołków o nieparzystym stopniu jest parzyście wiele.
- **Reprezentacja grafu**: macierz sąsiedztwa (n × n pól, M[u][v] = 1, gdy jest krawędź) albo listy sąsiedztwa (dla każdego wierzchołka lista sąsiadów – 2 · |E| wpisów w grafie nieskierowanym). Graf modeluje sieci dróg, znajomości, połączenia komputerów, zależności zadań.
- **Przeszukiwanie wszerz (BFS)** używa **kolejki** i wyznacza najmniejszą liczbę krawędzi od startu do każdego wierzchołka; **w głąb (DFS)** – stosu (lub rekurencji). Oba wyznaczają spójne składowe grafu.

Przykład 2. **Zamień na ONP wyrażenie (8 − 3) · (2 + 4) / 5.**

symbol	wyjście	stos operatorów
(		(
8	8	(
−	8	(−
3	8 3	(−
)	8 3 −	pusty
*	8 3 −	*
(	8 3 −	* (
2	8 3 − 2	* (
+	8 3 − 2	* (+
4	8 3 − 2 4	* (+
)	8 3 − 2 4 +	*
/	8 3 − 2 4 + *	/
5	8 3 − 2 4 + * 5	/
koniec	8 3 − 2 4 + * 5 /	pusty

Wynik: **8 3 − 2 4 + * 5 /** (wartość 30 / 5 = 6). Przy „/” na stosie jest „*” o tym samym priorytecie – przenosimy go na wyjście (łączność lewostronna).

Przykład 3. **Wyznacz najmniejszą liczbę krawędzi z A do każdego wierzchołka grafu (BFS, sąsiedzi w kolejności alfabetycznej).**



Kolejka: [A] → zdejmujemy A, dodajemy B, C (odl. 1) → zdejmujemy B, dodajemy D (2) → zdejmujemy C, dodajemy F (2) → zdejmujemy D, dodajemy E (3) → F, E. Odległości: B = 1, C = 1, D = 2, F = 2, E = 3.

```
from collections import deque

def bfs(g, s):                  # g[u] – lista sąsiadów u
    odl = {s: 0}
    q = deque([s])
    while q:
        u = q.popleft()
        for v in g[u]:
            if v not in odl:
                odl[v] = odl[u] + 1
                q.append(v)
    return odl
```

Przykład 4. **Do pustego stosu i do pustej kolejki wstawiamy kolejno 1, 2, 3, 4, a potem dwa razy usuwamy element. Co zostanie?**

Stos: zdejmujemy 4, potem 3 – zostają 1, 2. Kolejka: obsługujemy 1, potem 2 – zostają 3, 4.

```
from collections import deque

kolejka = deque()
kolejka.append("Ala")          # dołącz na koniec
kolejka.append("Bartek")
kolejka.append("Celina")
pierwszy = kolejka.popleft()   # obsłuż pierwszego -> "Ala"
```

Zadania

Zadanie 1. (0–2 pkt) ..... / 2

Na pustym stosie wykonano: włóż 3, włóż 8, usuń, włóż 5, włóż 1, usuń, usuń.

1. Zdjęte elementy (po kolei): _____ 2. Na stosie zostało: _____

Zadanie 2. (0–2 pkt) ..... / 2

Na pustej kolejce wykonano te same operacje: włóż 3, włóż 8, usuń, włóż 5, włóż 1, usuń, usuń.

1. Usunięte elementy (po kolei): _____ 2. W kolejce zostało: _____

Zadanie 3. (0–2 pkt) ..... / 2

Oblicz wartości wyrażeń zapisanych w ONP.

1. 4 6 + 3 * = _____ 2. 9 2 3 * − = _____

Zadanie 4. (0–1 pkt) ..... / 1

Stopnie wierzchołków grafu nieskierowanego to 3, 2, 2, 3, 2. Ile krawędzi ma ten graf?

A. 5 B. 12 C. 7 D. 6

Stos, kolejka, lista, ONP; grafy – ćwiczenia podstawowe

Ćwiczę umiejętności z działu

ZR </> KARTA 2/8

Imię i nazwisko: Klasa: Data: Wynik: ..... / 22 pkt

Ćwiczenia · 22 pkt

Zadanie 1. (0–3 pkt) ..... / 3

Oblicz stosem wartość wyrażenia w ONP: 5 3 4 + * 6 2 * −. Uzupełnij tabelę (szczyt stosu z prawej).

symbol	stos po symbolu
5	5
3	5 3
4	
+	
*	
6	
2	
*	
−	

Zadanie 2. (0–4 pkt) ..... / 4

Zapisz wyrażenia w ONP.

1. $a + b \cdot c \rightarrow$ _____
2. $(a + b) \cdot c \rightarrow$ _____
3. $a - b - c \rightarrow$ _____
4. $(a + b) \cdot (c - d) \rightarrow$ _____

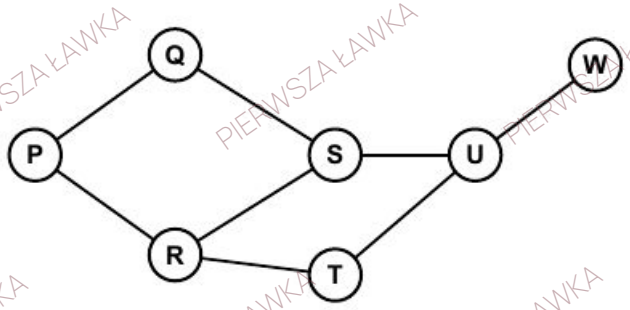
Zadanie 3. (0–3 pkt) ..... / 3

Zapisz wyrażenia w ONP w zwykłej postaci (z nawiasami tylko tam, gdzie są potrzebne).

1. 2 3 4 * + → _____
2. 2 3 + 4 * → _____
3. 8 2 − 3 − → _____

Zadanie 6. (0–4 pkt) ..... / 4

Przeszukiwaniem wszerz (BFS) wyznacz najmniejszą liczbę krawędzi od P.



1. S: _____ 2. T: _____ 3. U: _____ 4. W: _____

Zadanie 7. (0–2 pkt) ..... / 2

Lista jednokierunkowa w tablicy: głowa ma indeks 1, „następny = 0" oznacza koniec listy.

indeks	wartość	następny
1	K	4
2	O	0
3	T	2
4	R	3

1. Wartości w kolejności listy: _____
2. Dopisano element 5 z wartością A między R a T. Jakie wartości „następny" mają teraz elementy 4 i 5?

_____

Zadanie 8. (0–2 pkt) ..... / 2

Uzupełnij luki w funkcji obliczającej wartość wyrażenia w ONP.

```
def wartosc_onp(wyrazenie):
    stos = []
    for s in wyrazenie.split():
        if s in "+-*":
            b = stos.pop()              # najpierw zdejmujemy PRAWY argument
            a = stos.pop()
            if s == "+":
                stos.append(a + b)
            elif s == "-":
                stos.append(luka 2)
            else:
                stos.append(a * b)
        else:
            luka 1
    return stos.pop()
```

1. luka 1: _____
2. luka 2: _____

Stos, kolejka, lista, ONP; grafy – utrwalenie

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

ZR </> KARTA 3/8

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

Zadania typu maturalnego · 13 pkt

Jak to sprawdza CKE

Grafy na maturze: 2026 zad. 4 „Korporacja" – hierarchia pracowników (drzewo zapisane jako numery przełożonych): 4.1 bez komputera (0–1), 4.2–4.4 przy komputerze (0–2, 0–2, 0–3); zad. 4.2 rozwiązało 28% zdających, a 4.4 było najtrudniejszym zadaniem arkusza (sprawozdanie CKE 2026). ONP i struktury stos/kolejka w arkuszach 2023–2026 nie wystąpiły jako osobne zadania – ćwiczymy je w formułach CKE: P/F (1 pkt za komplet), „Dokończ zdanie", „Uzupełnij tabelę", zapis algorytmu ze *Specyfikacją*.

Zadanie 1. (0–1 pkt) ..... / 1

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

1. Element włożony na stos jako ostatni zostanie zdjęty jako pierwszy.	**P**	**F**
2. Wartość wyrażenia zapisanego w ONP 3 4 2 * − jest równa 14.	**P**	**F**
3. W grafie nieskierowanym liczba wierzchołków o nieparzystym stopniu może być nieparzysta.	**P**	**F**

Zadanie 2. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Wyrażenie (7 − 2) · 3 + 8 / 4 zapisane w ONP to

A. + * − 7 2 3 / 8 4
B. 7 2 3 − * 8 4 / +
C. 7 2 − 3 * 8 4 / +
D. 7 2 − 3 8 4 * / +

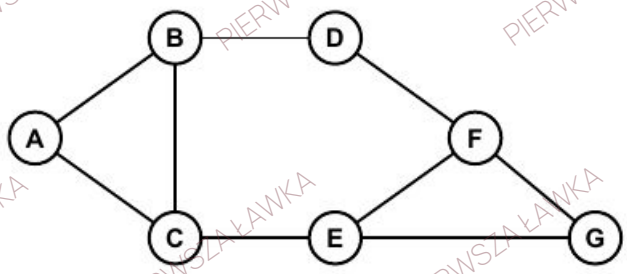
Zadanie 3. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Podczas obliczania stosem wyrażenia w ONP 2 3 4 5 + * 6 − + największa liczba elementów na stosie jest równa

A. 2
B. 4
C. 5
D. 3

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

Stos, kolejka, lista, ONP; grafy – w praktyce, przy komputerze

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

ZR </> KARTA 4/8

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 14 pkt

Ćwiczenia · 14 pkt

Zadanie 1. (0–7 pkt)

Kalkulator ONP

Opis danych

Plik **wyrazenia.txt** zawiera 300 wierszy. W każdym jest ciąg symboli oddzielonych spacjami: liczby od 1 do 9 i operatory +, –, * (w pliku: znaki + - *). Część wierszy nie jest poprawnym wyrażeniem w ONP (np. brakuje argumentu operatora albo na końcu na stosie zostaje więcej niż jedna liczba). Pierwsze wiersze pliku:

```
8 4 - 7 6 - *
4 5 - 4 + 2 8 - 1 9 - - * 3 3 - 2 + 9 3 9 - + + +
9 1 + 9 7 +
9 5 6 + 1 5 + - *
```

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

Do dyspozycji masz plik **wyrazenia.txt** oraz mniejszy plik **wyrazenia_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 wierszy pliku to poprawne wyrażenia w ONP. *Dla danych z pliku przykładowego odpowiedź: 14.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 1.2. (0–2 pkt) ..... / 2

Podaj największą wartość poprawnego wyrażenia i numer wiersza, w którym ono jest (oddziel spacją). *Dla danych z pliku przykładowego odpowiedź: 352 10.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 1.3. (0–3 pkt) ..... / 3

Podczas obliczania wyrażenia stosem liczba elementów na stosie się zmienia. Podaj największą liczbę elementów na stosie, jaka wystąpi przy obliczaniu poprawnych wyrażeń z pliku, i liczbę wyrażeń, w których ją osiągnięto. *Dla danych z pliku przykładowego odpowiedź: 5 2.*

Odpowiedź zapisz w pliku wyniki1.txt

CO JESZCZE JEST W ŚRODKU

Stos, kolejka, lista, ONP; grafy – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 4 LO i technikum (zakres rozszerzony)

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

- Karta 1: Stos, kolejka, lista, ONP; grafy – najważniejsze wiadomości – 4 zadania, 7 pkt
- Karta 2: Stos, kolejka, lista, ONP; grafy – ćwiczenia podstawowe – 8 zadań, 22 pkt
- Karta 3: Stos, kolejka, lista, ONP; grafy – utrwalenie – 7 zadań, 13 pkt
- Karta 4: Stos, kolejka, lista, ONP; grafy – w praktyce, przy komputerze – 2 zadania, 14 pkt
- Karta 5: Kartkówka: stos, kolejka, ONP i grafy – grupa A – 4 zadania, 8 pkt
- Karta 6: Kartkówka: stos, kolejka, ONP i grafy – grupa B – 4 zadania, 8 pkt
- Karta 7: Sprawdzian: Stos, kolejka, lista, ONP; grafy – grupa A – 9 zadań, 20 pkt
- Karta 8: Sprawdzian: Stos, kolejka, lista, ONP; grafy – grupa B – 9 zadań, 20 pkt Karta 1 przypomina teorię i pokazuje przykłady rozwiązane krok po kroku, dalej ćwiczenia, utrwalen

Kup pełną wersję – plik od razu na e-mail
pierwszalawka.pl/material/informatyka-liceum-klasa-4-zr-dzial-12-stos-kolejka-lista-onp-grafy