

# WYMAGANIA EDUKACYJNE NA POSZCZEGÓLNE OCENY

## Informatyka

LO 4-letnie

### Klasa III (rozszerzona) – I półrocze

Dział / ocena	Opis wymagań
<b>1. Rozwiązywanie problemów z wykorzystaniem dynamicznych struktur danych</b>	
dopuszczający	<p>Uczeń potrafi:</p> <ul style="list-style-type: none"><li>• pisze programy o niewielkim stopniu trudności,</li><li>• wyjaśnia, co to jest notacja infiksowa, notacja prefiksowa, odwrotna notacja polska, drzewo wyrażenia algebraicznego,</li><li>• definiuje pojęcie dynamicznej struktury danych,</li><li>• definiuje dynamiczne struktury danych takie jak: stos, kolejka, lista, vector,</li><li>• wymienia rodzaje list,</li><li>• wyjaśnia, na czym polega sortowanie leksykograficzne,</li><li>• definiuje graf, wymienia elementy i rodzaje grafów, wymienia sposoby reprezentacji grafu (macierz sąsiedztwa, lista sąsiedztwa),</li></ul>
dostateczny	<ul style="list-style-type: none"><li>• wyróżnia operacje, które można wykonywać na dynamicznych strukturach danych (stosie, kolejce, liście, typie vector),</li><li>• omawia zastosowanie dynamicznych struktur danych na różnych przykładach,</li><li>• zapisuje wyrażenia algebraiczne bez użycia nawiasów, w tym w postaci odwrotnej notacji polskiej,</li><li>• oblicza wartość wyrażenia arytmetycznego zapisanego w odwrotnej notacji polskiej,</li><li>• omawia algorytmy znajdowania wyjścia z labiryntu z wykorzystaniem iteracji i rekurencji,</li><li>• symuluje problem Flawiusza,</li><li>• sortuje dane leksykograficznie,</li><li>• stosuje typ vector do reprezentacji grafu w postaci list sąsiedztwa,</li><li>• omawia algorytm przeszukiwania grafu w głąb (DFS),</li><li>• omawia algorytm przeszukiwania grafu wszerz (BFS),</li><li>• wyjaśnia, do czego służy algorytm Dijkstry,</li><li>• wyjaśnia różnicę między przekazywaniem parametrów do funkcji przez wartość i przez referencję,</li><li>• wykorzystuje pliki tekstowe do wczytywania danych i zapisywania wyników,</li></ul>

dobry	<ul style="list-style-type: none"> <li>• pisze programy o różnym stopniu trudności, szacuje ich efektywność,</li> <li>• dobiera typy danych do rozwiązania problemu,</li> <li>• do przeglądania grafu stosuje algorytm przeszukiwania w głąb (DFS) oraz algorytm przeszukiwania grafu wszerz (BFS),</li> <li>• omawia algorytm Dijkstry,</li> </ul>
bardzo dobry	<ul style="list-style-type: none"> <li>• charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,</li> <li>• pisze programy o podwyższonym stopniu trudności: rozwiązuje zadania oznaczone trzema gwiazdkami w podręczniku,</li> <li>• optymalizuje rozwiązania,</li> <li>• stosuje zaawansowane funkcje środowiska i języka programowania,</li> <li>• dobiera struktury danych i metody do rodzaju problemu,</li> <li>• szacuje złożoność algorytmów,</li> <li>• implementuje algorytmy grafowe – BFS, DFS, algorytm Dijkstry,</li> </ul>
celujący	<ul style="list-style-type: none"> <li>• charakteryzuje skomplikowane sytuacje algorytmiczne,</li> <li>• proponuje optymalne rozwiązanie sytuacji problemowej</li> <li>• stosuje złożone struktur danych,</li> </ul>
<b>2. Rozwiązywanie problemów z wykorzystaniem dynamicznych struktur danych</b>	
dopuszczający	<ul style="list-style-type: none"> <li>• omawia różnice między stałoprzecinkową a zmiennoprzecinkową reprezentacją liczb rzeczywistych w komputerze,</li> <li>• wymienia rodzaje błędów w obliczeniach komputerowych, rozróżnia błąd względny i bezwzględny,</li> <li>• znajduje wartość wielomianu algorytmem naiwnym,</li> <li>• wie, na czym polegają podstawowe metody obliczeń przybliżonych,</li> <li>• zna proste algorytmy badające własności geometryczne (np. położenie punktu względem prostej),</li> <li>• wyjaśnia, co to jest fraktal, wskazuje przykłady struktur fraktalnych występujących w przyrodzie,</li> </ul>
dostateczny	<ul style="list-style-type: none"> <li>• omawia algorytm znajdujący rozwinięcie binarne nieskracalnego ułamka właściwego,</li> <li>• zapisuje liczby w postaci znormalizowanej,</li> <li>• definiuje liczby pojedynczej precyzji i liczby podwójnej precyzji,</li> </ul>

	<ul style="list-style-type: none"> <li>• wykonuje działania na liczbach zmiennoprzecinkowych,</li> <li>• wskazuje różnice między algorytmem stabilnym a algorytmem niestabilnym,</li> <li>• znajduje pierwiastki równania kwadratowego algorytmem stabilnym i algorytmem niestabilnym,</li> <li>• implementuje algorytm obliczający wartość wielomianu z zastosowaniem schematu Hornera,</li> <li>• stosuje w algorytmach numerycznych metody: bisekcji, Newtona–Raphsona, trapezów, prostokątów,</li> <li>• omawia algorytmy badające własności geometryczne – położenie punktu względem prostej, przecinania się odcinków, przynależności punktu do figury,</li> <li>• podaje przykłady fraktali (zbiór Cantora, drzewo binarne, dywan Sierpińskiego, płatek Kocha), wyjaśnia sposób tworzenia tych fraktali,</li> </ul>
dobry	<ul style="list-style-type: none"> <li>• znajduje reprezentację liczby zapisanej w systemie dziesiętnym jako liczby pojedynczej i liczby podwójnej precyzji,</li> <li>• świadomie używa typów <code>float</code> i <code>double</code> w zadaniach,</li> <li>• stosuje schemat Hornera do zamiany liczby w systemie pozycyjnym o wybranej podstawie na liczbę dziesiętną,</li> <li>• stosuje metodę Monte Carlo w obliczeniach przybliżonych,</li> <li>• w algorytmach badających własności geometryczne wykorzystuje macierz oraz regułę Sarrusa do obliczania wyznacznika macierzy,</li> </ul>
bardzo dobry	<ul style="list-style-type: none"> <li>• w reprezentacji liczb rzeczywistych w komputerze stosuje reprezentację stało- lub zmiennoprzecinkową zgodnie ze specyfikacją algorytmu, minimalizując błędy w obliczeniach,</li> <li>• stosuje schemat Hornera do szybkiego podnoszenia do potęgi,</li> <li>• implementuje algorytmy numeryczne: znajdowania miejsc zerowych funkcji oraz obliczania pierwiastka kwadratowego metodą bisekcji, obliczania pierwiastka kwadratowego metodą Newtona–Raphsona, obliczania pola obszaru zamkniętego metodą prostokątów i metodą trapezów, znajdowania przybliżenia liczby pi oraz symulacja ruchów Browna metodą Monte Carlo,</li> <li>• implementuje algorytmy badające własności geometryczne,</li> <li>• implementuje w języku JavaScript algorytmy generujące fraktale danego stopnia,</li> <li>• stosuje metodę IFS do tworzenia fraktali w arkuszu kalkulacyjnym,</li> </ul>

celujący	<ul style="list-style-type: none"> <li>• optymalizuje programy, szacuje ich efektywność,</li> <li>• wykorzystuje poznane algorytmy do rozwiązywania złożonych problemów obliczeniowych</li> </ul>
<b>3. Zaawansowane algorytmy i techniki programistyczne</b>	
dopuszczający	<ul style="list-style-type: none"> <li>• wyszukuje wzorec w tekście algorytmem naiwnym,</li> <li>• rozumie działanie funkcji haszującej,</li> <li>• wskazuje różnice między kryptografią symetryczną i kryptografią asymetryczną, definiuje pojęcia klucz publiczny i klucz prywatny,</li> <li>• wyjaśnia, do czego służy algorytm RSA, i wyróżnia główne etapy tego algorytmu (generowanie kluczy, szyfrowanie z kluczem publicznym oraz deszyfrowanie z kluczem prywatnym),</li> <li>• definiuje programowanie strukturalne,</li> <li>• definiuje programowanie obiektowe i podstawowe pojęcia z nim związane,</li> </ul>
dostateczny	<ul style="list-style-type: none"> <li>• implementuje algorytm naiwny wyszukiwania wzorca w tekście,</li> <li>• wyjaśnia metodę haszowania,</li> <li>• wyjaśnia, jak generuje się klucze publiczny i prywatny oraz szyfruje i deszyfruje informacje w algorytmie RSA,</li> <li>• wyjaśnia, na czym polegają metoda zstępująca i metoda wstępująca,</li> <li>• w programowaniu obiektowym definiuje własne klasy, korzystając ze specyfikatorów dostępu,</li> </ul>
dobry	<ul style="list-style-type: none"> <li>• omawia algorytm Karpa–Rabina do wyszukiwania wzorca w tekście z zastosowaniem funkcji haszującej,</li> <li>• w programowaniu obiektowym stosuje hierarchię klas, wyjaśnia, na czym polega hermetyzacja danych i jakie jest zastosowanie operatora zasięgu,</li> </ul>
bardzo dobry	<ul style="list-style-type: none"> <li>• stosuje programowanie obiektowe, definiując własne klasy, obiekty, atrybuty i metody, deklaruje konstruktory w klasach, wyjaśnia, na czym polega polimorfizm i czym są metody wirtualne,</li> <li>• pisze program generujący klucz prywatny i klucz publiczny w algorytmie RSA,</li> </ul>
celujący	<ul style="list-style-type: none"> <li>• stosuje funkcję haszującą oraz algorytm Karpa–Rabina w programach wyszukujących wzorec w tekście,</li> </ul>

	<ul style="list-style-type: none"> <li>• pisze programy szyfrujące i deszyfrujące informacje w algorytmie RSA,</li> </ul>
--	---

## II półrocze

4. Relacyjne bazy danych	
dopuszczający	<ul style="list-style-type: none"> <li>• zna podstawowe pojęcia dotyczące relacyjnych baz danych,</li> <li>• wie, co to jest język SQL, zna podstawowe klauzule tego języka,</li> <li>• zna zasady tworzenia zapytań do bazy z wykorzystaniem języka SQL,</li> <li>• wyróżnia etapy pracy nad aplikacją internetową, rozróżnia technologie back-end i front-end,</li> </ul>
dostateczny	<ul style="list-style-type: none"> <li>• wyszukuje informacje w bazach danych, tworzy formularze, kwerendy i raporty,</li> <li>• wykorzystuje język SQL do tworzenia i usuwania baz danych, dodawania tabel do baz danych, usuwania tabel z baz, dodawania rekordów do tabel, importowania danych do tabel, edycji rekordów,</li> <li>• tworzy konta użytkowników i przydziela im uprawnienia do wybranej bazy, używając języka SQL,</li> <li>• formułuje zapytania zwracające określone dane, sortuje wyniki zapytań,</li> <li>• wyjaśnia, na czym polega praca nad aplikacją internetową, instaluje i konfiguruje niezbędne oprogramowanie, przygotowuje bazę danych na potrzeby projektu,</li> </ul>
dobry	<ul style="list-style-type: none"> <li>• projektuje i tworzy proste bazy danych,</li> <li>• przy tworzeniu aplikacji internetowej projektuje i tworzy interfejs użytkownika, zapewnia komunikację aplikacji z bazą danych,</li> </ul>
bardzo dobry	<ul style="list-style-type: none"> <li>• projektuje zaawansowane relacyjne bazy danych, zarządza nimi, tworzy tabele pomostowe, formularze, kwerendy i raporty,</li> <li>• formułuje zapytania w języku SQL, stosując selekcję, sortowanie, projekcję oraz agregowanie danych,</li> </ul>
celujący	<ul style="list-style-type: none"> <li>• tworzy aplikacje internetowe z przejrzystym interfejsem użytkownika korzystające z sieciowej bazy danych, testuje je i wprowadza poprawki,</li> <li>• projektuje rozbudowane relacyjne bazy danych, zarządza nimi, wykorzystując zaawansowane narzędzia oraz klauzule języka SQL,</li> </ul>