

Wymaganie edukacyjne z informatyki – klasa III, zakres rozszerzony

Ocenę niedostateczną otrzymuje uczeń, który:	Ocenę dopuszczającą otrzymuje uczeń, który:	Ocenę dostateczną otrzymuje uczeń, który spełnia kryteria oceny dopuszczającej, a ponadto:	Ocenę dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:	Ocenę bardzo dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:	Ocenę celującą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:
<ul style="list-style-type: none"> • nie opanował podstawowych wiadomości i umiejętności, co uniemożliwia zdobywanie dalszej wiedzy, • nie jest w stanie scharakteryzować podstawowych pojęć (algorytm, warunek, iteracja, rekurencja), • nie zna prostych algorytmów, • nie rozwiązuje najprostszyc zadań, • nie bierze czynnego udziału w lekcjach, nie wykonuje zadań, nie pisze 	<ul style="list-style-type: none"> • omawia zasadę złotego podziału, • opisuje rozszerzony algorytm Euklidesa, • omawia metody zachłanne na przykładzie problemu kasjera, harmonogramu sali i wyszukiwania drogi, • porównuje metody zachłanną i dynamiczną, • pisze programy o niewielkim stopniu trudności, 	<ul style="list-style-type: none"> • omawia rozszerzony algorytm Euklidesa, • formułuje algorytm wydawania reszty minimalną liczbą monet, harmonogramu wykorzystania sali, znajdowania drogi metodami zachłanną i dynamiczną, • zapisuje wyrażenia algebraiczne bez użycia nawiasów, w tym w postaci odwrotnej notacji polskiej, • oblicza wartość wyrażenia arytmetycznego zapisanego w odwrotnej notacji polskiej, • omawia algorytmy znajdowania wyjścia z labiryntu z wykorzystaniem iteracji i rekurencji, • stosuje typ vector do reprezentacji grafu w postaci list sąsiedztwa, • omawia algorytm przeszukiwania grafu w głąb (DFS), • omawia algorytm przeszukiwania grafu wszerz (BFS), 	<ul style="list-style-type: none"> • zapisuje w postaci programu rozszerzony algorytm Euklidesa, wyjaśnia jego działanie i zastosowanie, • stosuje metodę zachłanną w programach – problem kasjera, harmonogram wykorzystania sali, wyszukiwanie drogi, • pisze programy o różnym stopniu trudności, szacuje ich efektywność, • dobiera typy danych do rozwiązania problemu, • do przeglądania grafu stosuje algorytm 	<ul style="list-style-type: none"> • do implementacji rozszerzonego algorytmu Euklidesa stosuje zarówno iterację, jak i rekurencję, • stosuje metody zachłanną i dynamiczną w problemach kasjera, harmonogramu wykorzystania sali i wyszukiwania drogi, wskazuje wady i zalety obu metod, szacuje złożoność czasową. • charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania, 	<ul style="list-style-type: none"> • stosuje w programach algorytmy sortowania inne niż omawiane na lekcjach (np. heapsort), • bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca, • charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych, • bierze udział w konkursach i olimpiadach informatycznych i zajmuje punktowane

<p>programów, nie odrabia prac domowych.</p>		<ul style="list-style-type: none"> • wyjaśnia, do czego służy algorytm Dijkstry, • wykorzystuje pliki tekstowe do wczytywania danych i zapisywania wyników, • omawia algorytm znajdujący rozwinięcie binarne nieskracalnego ułamka właściwego, • zapisuje liczby w postaci znormalizowanej, • definiuje liczby pojedynczej precyzji i liczby podwójnej precyzji, • wykonuje działania na liczbach zmiennoprzecinkowych, • wskazuje różnice między algorytmem stabilnym a algorytmem niestabilnym, • znajduje pierwiastki równania kwadratowego algorytmem stabilnym i algorytmem niestabilnym, • implementuje algorytm obliczający wartość wielomianu z zastosowaniem schematu Hornera, 	<p>przeszukiwania w głąb(DFS) oraz algorytm przeszukiwania grafu wszerz (BFS),</p> <ul style="list-style-type: none"> • omawia algorytm Dijkstry, • znajduje reprezentację liczby zapisanej w systemie dziesiętnym jako liczby pojedynczej i liczby podwójnej precyzji, • świadomie używa typów <code>float</code> i <code>double</code> w zadaniach, • stosuje schemat Hornera do zamiany liczby w systemie pozycyjnym o wybranej podstawie na liczbę dziesiętną, 	<ul style="list-style-type: none"> • pisze programy o podwyższonym stopniu trudności: rozwiązuje zadania oznaczone trzema gwiazdkami w podręczniku, • optymalizuje rozwiązania, • stosuje zaawansowane funkcje środowiska i języka programowania, • dobiera struktury danych i metody do rodzaju problemu, • szacuje złożoność algorytmów, 	<p>miejsca,</p> <ul style="list-style-type: none"> • pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych, • optymalizuje programy, szacuje ich efektywność,
--	--	--	--	---	--