

Zasady oceniania z **informatyki** (zakres rozszerzony) - klasa IV Technikum ZST w Kłodzku

Ocenę niedostateczną otrzymuje uczeń, który:	Ocenę dopuszczającą otrzymuje uczeń, który:	Ocenę dostateczną otrzymuje uczeń, który:	Ocenę dobrą otrzymuje uczeń, który:	Ocenę bardzo dobrą otrzymuje uczeń, który:	Ocenę celującą otrzymuje uczeń, który:
<ul style="list-style-type: none"> nie opanował podstawowych wiadomości i umiejętności niezbędnych do dalszego zdobywania wiedzy, nie rozwiązuje najprostszyc zadań z pomocą nauczyciela, nie wykazuje zainteresowania treściami prezentowanymi na lekcjach, nie rozwiązuje ćwiczeń, zadań domowych, otrzymuje cząstkowe oceny niedostateczne, których nie poprawia. 	<ul style="list-style-type: none"> definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja, wymienia sposoby reprezentacji algorytmów, korzysta ze środowiska programistycznego: pisze w nim kod, kompiluje i uruchamia program, odczytuje i zapisuje pliki, pisze programy o niewielkim stopniu trudności, omawia pojęcia: złożoność obliczeniowa algorytmu, algorytm naiwny, algorytm optymalny, złożoność pesymistyczna, złożoność oczekiwana (średnia), korzysta z podstawowych 	<ul style="list-style-type: none"> przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu, schematu blokowego, dobdaje liczby binarne, konwertuje liczby między pozycyjnymi systemami liczbowymi, wykonuje działania arytmetyczne na liczbach w systemach liczbowych o różnych podstawach, przedstawia liczby w kodzie U2, definiuje pojęcie zdania logicznego, charakteryzuje podstawowe operacje logiczne (koniunkcja, alternatywa, negacja) oraz operatory logiczne, charakteryzuje wybrane typy zmiennych służących do zapisu liczb całkowitych w języku 	<ul style="list-style-type: none"> określa specyfikację algorytmu (dane, wynik), pisze programy o różnym stopniu trudności, szacuje ich efektywność, przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, schematu blokowego, pseudokodu, dobiera typy danych do realizacji problemu, stosuje zmienne typu unsigned w tworzonych programach, pisze programy konwertujące liczby między systemem dziesiętnym i binarnym, implementuje w języku C++ algorytmy wykonujące działania arytmetyczne na 	<ul style="list-style-type: none"> charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania, pisze programy o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku, optymalizuje rozwiązania, stosuje zaawansowane funkcje środowiska i języka programowania (np. z biblioteki STL), dobiera struktury danych i metody do rodzaju problemu, pisze programy konwertujące liczby między różnymi systemami pozycyjnymi, w programach wykonujących działania na liczbach w różnych systemach 	<ul style="list-style-type: none"> charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++, pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych lub oznaczone trzema gwiazdkami w podręczniku, wyszukuje palindromy lub anagramy w plikach tekstowych, tworzy palindromy z napisów, dopisując minimalną liczbę znaków, implementuje w języku C++ algorytm wyszukiwania

Zasady oceniania z **informatyki** (zakres rozszerzony) - klasa IV Technikum ZST w Kłodzku

	<p>funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych,</p> <ul style="list-style-type: none"> wymienia podstawowe typy danych, operacje arytmetyczne i logiczne, w pisanych programach korzysta ze strukturalnych typów danych: napisów, struktur, tablic, definiuje pojęcie systemów liczbowych, wyjaśnia, czym jest tablica kodów ASCII, wymienia systemy liczbowe używane w informatyce, konwertuje liczby między systemami binarnym i decymalnym, dodaje pisemnie liczby binarne, wyjaśnia, czym są palindrom i anagram, podaje przykłady, 	<p>C++: short int, int, long int, long long int,</p> <ul style="list-style-type: none"> pisze programy wykonujące działania na liczbach całkowitych, korzysta z biblioteki string do operacji na łańcuchach znaków, wykonuje operacje na napisach, wykorzystując słowa kluczowe: size, find, substr, erase, toupper, tolower, wczytuje napisy ze spacjami, wykorzystując słowo kluczowe getline, tworzy algorytmy sprawdzające, czy napis jest palindromem, przedstawia w postaci algorytmu problem wyszukiwania anagramów, przy pisaniu programów stosuje własne funkcje różnych typów, w tym funkcję typu void, wyjaśnia różnicę między parametrami formalnym i aktualnym, a także 	<p>liczbach w różnych systemach,</p> <ul style="list-style-type: none"> w algorytmach zamiany wykorzystuje zależności między systemami binarnym, ósemkowym i heksadecymalnym, omawia sposób reprezentacji obrazów w komputerze, korzystając z takich pojęć jak: piksel, model RGB, kanał alfa, wyjaśnia, na czym polega digitalizacja (dyskretyzacja) dźwięku, wyjaśnia zasadę tworzenia animacji, stosuje różne sposoby przekazywania parametrów do funkcji: przez wartość, referencję lub wskaźnik, implementuje w języku C++ algorytmy sprawdzające, czy napis jest palindromem, pisze programy sprawdzające, czy dwa napisy są anagramami, 	<p>pozycyjnych wykorzystuje bibliotekę string i strukturalne typy danych,</p> <ul style="list-style-type: none"> wykorzystuje rozwinięcie binarne liczby dziesiętnej w algorytmie szybkiego podnoszenia do potęgi, wykonuje operacje arytmetyczne na liczbach w różnych systemach, implementuje je w języku C++, stosuje różne sposoby przekazywania parametrów do funkcji, uzasadnia ich użycie, pisze funkcje typu logicznego, np. sprawdzającą, czy napis jest palindromem, sprawdza, czy napisy są anagramami, stosując sortowanie lub zliczanie znaków, przy testowaniu liczby na pierwszość stosuje funkcję typu logicznego, 	<p>binarnego w wersji rekurencyjnej,</p> <ul style="list-style-type: none"> pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary), bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca, w projektach zespołowych przyjmuje rolę lidera.
--	---	--	---	--	---

Zasady oceniania z **informatyki** (zakres rozszerzony) - klasa IV Technikum ZST w Kłodzku

	<ul style="list-style-type: none"> • podaje definicje liczby pierwszej i liczby złożonej, • implementuje w języku C++ algorytm zliczający dzielniki danej liczby, • omawia geometryczną interpretację algorytmu Euklidesa, • definiuje pojęcia: kryptologia, kryptografia, kryptoanaliza, tekst jawny, klucz, szyfrogram, • rozróżnia szyfry podstawieniowe i przestawieniowe, • omawia szyfr Cezara jako przykład szyfru podstawieniowego i szyfr kolumnowy jako przykład szyfru przestawieniowego, • wyjaśnia, na czym polega łamanie szyfru, • omawia algorytm zliczania znaków w tekście, • wyjaśnia, na czym polega metoda „dziel i zwyciężaj”, 	<p>między zmiennymi lokalną i globalną,</p> <ul style="list-style-type: none"> • implementuje w języku C++ algorytm naiwny sprawdzający, czy liczba jest pierwsza, • implementuje w języku C++ algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem, • pisze program szyfrujący napis szyfrem Cezara, • omawia algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic, • implementuje w języku C++ algorytmy wyszukiwania liniowego i liniowego z wartownikiem, porównuje ich efektywność, • przedstawia w postaci listy kroków lub schematu blokowego algorytmy sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez 	<p>wykorzystując funkcję sort z biblioteki STL,</p> <ul style="list-style-type: none"> • implementuje w języku C++ i optymalizuje algorytm sprawdzający, czy liczba jest pierwsza, • pisze program rozkładający liczby na czynniki pierwsze, • stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW, • wykorzystuje algorytm Euklidesa do działań na ułamkach, • szyfruje dane wczytane z pliku tekstowego, • implementuje w języku C++ algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic, • stosuje algorytm wyszukiwania binarnego i oszacowuje jego złożoność czasową, • pisze programy sortujące metodami prostymi z 	<ul style="list-style-type: none"> • wyszukuje liczby bliźniacze, • wykorzystuje algorytm Euklidesa do działań na ułamkach, stosując struktury lub pary (typ pair), • szyfruje dane wczytane z pliku z uwzględnieniem polskich znaków diakrytycznych, • pisze program odczytujący informację ukrytą za pomocą szyfru Cezara z wykorzystaniem analizy częstości znaków w tekście, • pisze program wyszukujący jednocześnie minimum i maksimum w zbiorze z wykorzystaniem metody „dziel i zwyciężaj” oraz podaje wzór na liczbę wykonywanych operacji, • szacuje złożoność obliczeniową programów sortujących, modyfikuje funkcje 	
--	---	---	---	---	--

Zasady oceniania z **informatyki** (zakres rozszerzony) - klasa IV Technikum ZST w Kłodzku

	<ul style="list-style-type: none"> wczytuje dane z pliku tekstowego, zapisuje wyniki w pliku, omawia algorytmy wyszukiwania liczby w zbiorach uporządkowanym i nieuporządkowanym, stosuje funkcję losującą w tworzonych programach, omawia metody sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie na przykładowych danych, wypisuje liczby pierwsze z zadanego przedziału, stosując metodę sita Eratostenesa. 	<p>scalanie, określa operacje dominujące,</p> <ul style="list-style-type: none"> omawia algorytm sita Eratostenesa, przedstawia algorytmy znajdowania spójnych podciągów, wyznaczania najdłuższego z nich oraz podciągu o największej sumie elementów, omawia algorytm znajdowania idola i lidera w zbiorze, implementuje w języku C++ algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi. 	<p>zastosowaniem funkcji typu void,</p> <ul style="list-style-type: none"> stosuje algorytmy sortowania szybkiego i przez scalanie, pisze program realizujący algorytm sita Eratostenesa, implementuje w języku C++ algorytmy wyszukujące spójne podciągi o różnych cechach, stosuje w programach algorytmy wyszukiwania lidera i idola w zbiorze. 	<p>sortując, zmieniając porządek sortowania,</p> <ul style="list-style-type: none"> wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach, optymalizuje program realizujący algorytm sita Eratostenesa i szacuje jego złożoność czasową, wyszukuje spójne podciągi w plikach tekstowych, stosując optymalne algorytmy (w tym programowanie dynamiczne), wyjaśnia ich działanie, pisze programy wyszukujące lidera i idola w zbiorze, optymalizuje je, szacuje złożoność czasową, aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy. 	
--	--	--	--	--	--