

## Liceum Ogólnokształcące Nr III w Otwocku

# Wymagania edukacyjne niezbędne do otrzymania przez ucznia poszczególnych śródrocznych i rocznych ocen klasyfikacyjnych z przedmiotu informatyka na poziomie rozszerzonym

## Klasa II

<b>Wymagania na poszczególne oceny</b>				
<b>Konieczne</b> (ocena dopuszczająca)	<b>Podstawowe</b> (ocena dostateczna)	<b>Rozszerzające</b> (ocena dobra)	<b>Dopelniające</b> (ocena bardzo dobra)	<b>Wykraczające</b> (ocena celująca)
<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>1. Algorytmy na liczbach całkowitych i tekstach</b>				
<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja,</li> <li>• wymienia sposoby reprezentacji algorytmów,</li> <li>• korzysta ze środowiska programistycznego: pisze w nim kod, kompiluje i uruchamia program, odczytuje i zapisuje pliki,</li> </ul>	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu, schematu blokowego,</li> <li>• dodaje liczby binarne,</li> <li>• konwertuje liczby między pozycyjnymi systemami liczbowymi,</li> <li>• wykonuje działania arytmetyczne na liczbach w systemach liczbowych o różnych podstawach,</li> </ul>	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• określa specyfikację algorytmu (dane, wynik),</li> <li>• pisze programy o różnym stopniu trudności, szacuje ich efektywność,</li> <li>• przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, schematu blokowego, pseudokodu,</li> <li>• dobiera typy danych do realizacji problemu,</li> </ul>	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,</li> <li>• pisze programy o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku,</li> <li>• optymalizuje rozwiązania,</li> <li>• stosuje zaawansowane funkcje środowiska i języka programowania (np. z biblioteki STL),</li> </ul>	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,</li> <li>• tworzy palindromy z napisów, dopisując minimalną liczbę znaków</li> <li>• pisze program rozkładający liczbę złożoną na dwie liczby</li> </ul>

<ul style="list-style-type: none"> <li>• pisze programy o niewielkim stopniu trudności,</li> <li>• omawia pojęcia: złożoność obliczeniowa algorytmu, algorytm naiwny, algorytm optymalny, złożoność pesymistyczna, złożoność oczekiwana (średnia),</li> <li>• korzysta z podstawowych funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych,</li> <li>• wymienia podstawowe typy danych, operacje arytmetyczne i logiczne,</li> <li>• w pisanych programach korzysta ze strukturalnych typów danych: napisów, struktur, tablic,</li> <li>• definiuje pojęcie systemów liczbowych,</li> <li>• wyjaśnia, czym jest tablica kodów ASCII,</li> <li>• wymienia systemy liczbowe używane w informatyce,</li> <li>• konwertuje liczby między systemami binarnym i decymalnym,</li> </ul>	<ul style="list-style-type: none"> <li>• przedstawia liczby w kodzie U2,</li> <li>• definiuje pojęcie zdania logicznego, charakteryzuje podstawowe operacje logiczne (koniunkcja, alternatywa, negacja) oraz operatory logiczne,</li> <li>• charakteryzuje wybrane typy zmiennych służących do zapisu liczb całkowitych w języku C++: short int, int, long int, long long int,</li> <li>• pisze programy wykonujące działania na liczbach całkowitych,</li> <li>• korzysta z biblioteki string do operacji na łańcuchach znaków,</li> <li>• wykonuje operacje na napisach, wykorzystując słowa kluczowe: size, find, substr, erase, toupper, tolower,</li> <li>• wczytuje napisy ze spacjami, wykorzystując słowo kluczowe getline,</li> <li>• tworzy algorytmy sprawdzające, czy napis jest palindromem,</li> </ul>	<ul style="list-style-type: none"> <li>• stosuje zmienne typu unsigned w tworzonych programach,</li> <li>• pisze programy konwertujące liczby między systemem dziesiętnym i binarnym,</li> <li>• implementuje w języku C++ algorytmy wykonujące działania arytmetyczne na liczbach w różnych systemach,</li> <li>• w algorytmach zamiany wykorzystuje zależności między systemami binarnym, ósemkowym i heksadecymalnym,</li> <li>• omawia sposób reprezentacji obrazów w komputerze, korzystając z takich pojęć jak: piksel, model RGB, kanał alfa,</li> <li>• wyjaśnia, na czym polega digitalizacja (dyskretyzacja) dźwięku,</li> <li>• wyjaśnia zasadę tworzenia animacji,</li> <li>• stosuje różne sposoby przekazywania parametrów do funkcji: przez wartość, referencję lub wskaźnik,</li> <li>• implementuje w języku C++ algorytmy sprawdzające, czy napis jest palindromem,</li> </ul>	<ul style="list-style-type: none"> <li>• dobiera struktury danych i metody do rodzaju problemu,</li> <li>• pisze programy konwertujące liczby między różnymi systemami pozycyjnymi,</li> <li>• w programach wykonujących działania na liczbach w różnych systemach pozycyjnych wykorzystuje bibliotekę string i strukturalne typy danych,</li> <li>• wykorzystuje rozwinięcie binarne liczby dziesiętnej w algorytmie szybkiego podnoszenia do potęgi,</li> <li>• wykonuje operacje arytmetyczne na liczbach w różnych systemach, implementuje je w języku C++,</li> <li>• stosuje różne sposoby przekazywania parametrów do funkcji, uzasadnia ich użycie,</li> <li>• pisze funkcje typu logicznego, np. sprawdzającą, czy napis jest palindromem,</li> <li>• sprawdza, czy napisy są anagramami, stosując sortowanie lub zliczanie znaków,</li> <li>• przy testowaniu liczby na pierwszość stosuje funkcję typu logicznego,</li> <li>• wyszukuje liczby bliźniacze,</li> </ul>	<ul style="list-style-type: none"> <li>• pierwsze (hipoteza Goldbacha),</li> <li>• pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere'a) i różnych kluczy,</li> <li>• wyszukuje palindromy lub anagramy w plikach tekstowych,</li> </ul>
--	--	--	--	---

<ul style="list-style-type: none"> <li>• dodaje pisemnie liczby binarne,</li> <li>• wyjaśnia, czym są palindrom i anagram, podaje przykłady,</li> <li>• podaje definicje liczby pierwszej i liczby złożonej,</li> <li>• implementuje w języku C++ algorytm zliczający dzielniki danej liczby,</li> <li>• definiuje pojęcia: kryptologia, kryptografia, kryptoanaliza, tekst jawny, klucz, szyfrogram,</li> <li>• rozróżnia szyfry podstawieniowe i przestawieniowe,</li> <li>• omawia szyfr Cezara jako przykład szyfru podstawieniowego i szyfr kolumnowy jako przykład szyfru przestawieniowego,</li> <li>• wyjaśnia, na czym polega łamanie szyfru</li> </ul>	<ul style="list-style-type: none"> <li>• przedstawia w postaci algorytmu problem wyszukiwania anagramów,</li> <li>• przy pisaniu programów stosuje własne funkcje różnych typów, w tym funkcję typu void,</li> <li>• wyjaśnia różnicę między parametrami formalnym i aktualnym, a także między zmiennymi lokalną i globalną,</li> <li>• implementuje w języku C++ algorytm naiwny sprawdzający, czy liczba jest pierwsza,</li> <li>• implementuje w języku C++ algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem,</li> <li>• pisze program szyfrujący napis szyfrem Cezara,</li> </ul>	<ul style="list-style-type: none"> <li>• pisze programy sprawdzające, czy dwa napisy są anagramami, wykorzystując funkcję sort z biblioteki STL,</li> <li>• implementuje w języku C++ i optymalizuje algorytm sprawdzający, czy liczba jest pierwsza,</li> <li>• pisze program rozkładający liczby na czynniki pierwsze,</li> <li>• stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW,</li> <li>• wykorzystuje algorytm Euklidesa do działań na ułamkach,</li> <li>• szyfruje dane wczytane z pliku tekstowego,</li> </ul>	<ul style="list-style-type: none"> <li>• wykorzystuje algorytm Euklidesa do działań na ułamkach, stosując struktury lub pary (typ pair),</li> <li>• szyfruje dane wczytane z pliku z uwzględnieniem polskich znaków diakrytycznych,</li> <li>• pisze program odczytujący informację ukrytą za pomocą szyfru Cezara z wykorzystaniem analizy częstości znaków w tekście</li> </ul>	
--	--	--	---	--

<b>Wymagania na poszczególne oceny</b>				
<b>Konieczne</b> (ocena dopuszczająca)	<b>Podstawowe</b> (ocena dostateczna)	<b>Rozszerzające</b> (ocena dobra)	<b>Dopelniające</b> (ocena bardzo dobra)	<b>Wykraczające</b> (ocena celująca)
<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>2. Rozwiązywanie problemów z wykorzystaniem struktur danych</b>				
<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• omawia algorytm zliczania znaków w tekście,</li> <li>• wyjaśnia, na czym polega metoda „dziel i zwyciężaj”,</li> <li>• wczytuje dane z pliku tekstowego, zapisuje wyniki w pliku,</li> <li>• omawia algorytmy wyszukiwania liczby w zbiorach uporządkowanym i nieuporządkowanym,</li> <li>• stosuje funkcję losującą w tworzonych programach,</li> <li>• omawia metody sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie na przykładowych danych,</li> <li>• wypisuje liczby pierwsze z danego przedziału, stosując metodę sita Eratostenesa,</li> <li>• wyszukuje w ciągu liczb spójne podciągi (nierosnący,</li> </ul>	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• omawia algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic,</li> <li>• implementuje w języku C++ algorytmy wyszukiwania liniowego i liniowego z wartością, porównuje ich efektywność,</li> <li>• przedstawia w postaci listy kroków lub schematu blokowego algorytmy sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie, określa operacje dominujące,</li> <li>• omawia algorytm sita Eratostenesa,</li> <li>• przedstawia algorytmy znajdowania spójnych podciągów, wyznaczania najdłuższego z nich oraz podciągu o</li> </ul>	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• implementuje w języku C++ algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic,</li> <li>• stosuje algorytm wyszukiwania binarnego i oszacowuje jego złożoność czasową,</li> <li>• pisze programy sortujące metodami prostymi z zastosowaniem funkcji typu void,</li> <li>• stosuje algorytmy sortowania szybkiego i przez scalanie,</li> <li>• pisze program realizujący algorytm sita Eratostenesa,</li> <li>• implementuje w języku C++ algorytmy wyszukiujące spójne podciągi o różnych cechach,</li> </ul>	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• pisze programy o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku,</li> <li>• optymalizuje rozwiązania,</li> <li>• pisze program wyszukiujący jednocześnie minimum i maksimum w zbiorze z wykorzystaniem metody „dziel i zwyciężaj” oraz podaje wzór na liczbę wykonywanych operacji,</li> <li>• szacuje złożoność obliczeniową programów sortujących, modyfikuje funkcje sortujące, zmieniając porządek sortowania,</li> <li>• wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach,</li> </ul>	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary),</li> <li>• stosuje zaawansowane algorytmy i struktury danych do wyszukiwania spójnych podciągów</li> <li>• w projektach zespołowych przyjmuje rolę lidera</li> <li>• pisze programy o wysokim stopniu trudności: rozwiązując zadania z olimpiad przedmiotowych, konkursów informatycznych lub oznaczone trzema gwiazdkami w podręczniku</li> </ul>

<p>niemalejący, stały), wskazuje najdłuższe, oblicza ich sumę,</p> <ul style="list-style-type: none"> <li>wskazuje idola i lidera w zbiorze danych,</li> <li>uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania o niewielkim stopniu trudności</li> </ul>	<p>największej sumie elementów,</p> <ul style="list-style-type: none"> <li>omawia algorytm znajdowania idola i lidera w zbiorze,</li> <li>uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania</li> </ul>	<ul style="list-style-type: none"> <li>stosuje w programach algorytmy wyszukiwania lidera i idola w zbiorze,</li> <li>uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej</li> </ul>	<ul style="list-style-type: none"> <li>optymalizuje program realizujący algorytm sita Eratostenesa i szacuje jego złożoność czasową,</li> <li>wyszukuje spójne podciągi w plikach tekstowych, stosując optymalne algorytmy (w tym programowanie dynamiczne), wyjaśnia ich działanie,</li> <li>pisze programy wyszukiujące lidera i idola w zbiorze, optymalizuje je, szacuje złożoność czasową,</li> <li>aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy</li> </ul>	
---	---	--	---	--

**Ocenę niedostateczną** otrzymuje uczeń, który:

- 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 programów, nie odrabia prac domowych.