

**Wymagania edukacyjne niezbędne do otrzymania przez  
ucznia śródrocznej i rocznej oceny klasyfikacyjnej z  
przedmiotu Informatyka – zakres rozszerzony**

**Klasa 2**

Wymagania edukacyjne na poszczególne oceny				
Ocena dopuszczająca	Ocena dostateczna	Ocena dobra	Ocena bardzo dobra	Ocena celująca
<b>Na ocenę śródroczną uczeń umie/potrafi:</b>				
<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> <li>pisze programy o niewielkim stopniu trudności,</li> </ul>	<ul style="list-style-type: none"> <li>przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu,</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> <li> przedstawia liczby w kodzie U2,</li> <li> definiuje pojęcie zdania logicznego, charakteryzuje podstawowe operacje logiczne (koniunkcja,</li> </ul>	<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, pseudokodu,</li> <li> dobiera typy danych do realizacji problemu,</li> <li> stosuje zmienne typu unsigned w tworzonych programach,</li> <li> pisze programy konwertujące liczby między systemem dziesiętnym i binarnym,</li> </ul>	<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> <li> dobiera struktury danych i metody do rodzaju problemu,</li> <li> pisze programy konwertujące liczby między różnymi systemami pozycyjnymi,</li> </ul>	<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> pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych lub oznaczone trzema gwiazdkami w podręczniku,</li> <li> wyszukuje palindromy lub anagramy w plikach tekstowych,</li> <li> tworzy palindromy z napisów, dopisując minimalną liczbę znaków,</li> </ul>

<ul style="list-style-type: none"> <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> </ul>	<p>alternatywa, negacja) oraz operatory logiczne,</p> <ul style="list-style-type: none"> <li>• charakteryzuje wybrane typy zmiennych służących do zapisu liczb całkowitych w języku C++:</li> </ul> <p>short int, int, long int, long long int,</p> <ul style="list-style-type: none"> <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> </ul>	<ul style="list-style-type: none"> <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>• 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> <li>• pisze programy sprawdzające, czy dwa napisy są anagramami,</li> </ul>	<ul style="list-style-type: none"> <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</li> </ul>	<ul style="list-style-type: none"> <li>• pisze program rozkładający liczbę złożoną na dwie liczby pierwsze (hipoteza Goldbacha),</li> <li>• implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję,</li> <li>• pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere'a) i różnych kluczy,</li> <li>• implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej,</li> <li>• pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary),</li> <li>• stosuje zaawansowane algorytmy i struktury danych</li> </ul>
---	---	--	--	---

<ul style="list-style-type: none"> <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> <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> </ul>	<ul style="list-style-type: none"> <li>• tworzy algorytmy sprawdzające, czy napis jest palindromem,</li> <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> </ul>	<p>wykorzystując funkcję sort</p> <p>z biblioteki STL,</p> <ul style="list-style-type: none"> <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>	<p>sortowanie lub zliczanie znaków,</p> <ul style="list-style-type: none"> <li>• przy testowaniu liczby na pierwszość stosuje funkcję typu logicznego,</li> <li>• wyszukuje liczby bliźniacze,</li> <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>	<p>do wyszukiwania spójnych podciągów,</p> <ul style="list-style-type: none"> <li>• stosuje zaawansowane algorytmy wyszukiwania, np. najlepszego wyboru (trwałych par), stosując rekurencję,</li> </ul>
---	--	--	---	---

<ul style="list-style-type: none"><li>• omawia geometryczną interpretację algorytmu Euklidesa,</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><li>• omawia algorytm zliczania znaków w tekście,</li></ul>	<ul style="list-style-type: none"><li>• pisze program szyfrujący napis szyfrem Cezara,</li></ul>			
---	--	--	--	--

<ul style="list-style-type: none"> <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> </ul>				
<b>Ocena roczna obejmuje zakres wiedzy i umiejętności z pierwszego półrocza oraz:</b>				
<ul style="list-style-type: none"> <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</li> </ul>	<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 wartownikiem,</li> </ul>	<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</li> </ul>	<ul style="list-style-type: none"> <li>- pisze program wyszukujący 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> </ul>	<ul style="list-style-type: none"> <li>• pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,</li> <li>• stosuje w programach algorytmy sortowania inne niż omawiane na lekcjach (np. heapsort),</li> <li>• bierze udział w olimpiadach i konkursach,</li> </ul>

<p>metodę sita Eratostenesa,</p> <ul style="list-style-type: none"> <li>• wyszukuje w ciągu liczb spójne podciągi (nierosnący, niemalejący, stały), wskazuje najdłuższe, oblicza ich sumę,</li> <li>• definiuje pojęcia iteracji i rekurencji,</li> <li>• omawia zasadę złotego podziału,</li> <li>• opisuje rozszerzony algorytm Euklidesa,</li> <li>• omawia metody zachłanne na przykładzie problemu kasjera, harmonogramu sali i wyszukiwania drogi,</li> <li>• porównuje metody zachłanną i dynamiczną,</li> </ul>	<p>porównuje ich efektywność,</p> <ul style="list-style-type: none"> <li>• przedstawia w postaci listy kroków 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 największej sumie elementów,</li> <li>• implementuje w języku C++ algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi,</li> </ul>	<p>zastosowaniem funkcji typu void,</p> <ul style="list-style-type: none"> <li>• stosuje algorytmy sortowania szybkiego i przez scalanie,</li> <li>• pisze program realizujący algorytm sita Eratostenesa,</li> <li>• implementuje w języku C++ algorytmy wyszukujące spójne podciągi o różnych cechach,</li> <li>• porównuje algorytmy iteracyjne i rekurencyjne (liczbę wykonywanych operacji), szacuje ich złożoność czasową,</li> <li>• zapisuje w postaci programu rozszerzony algorytm Euklidesa, wyjaśnia jego działanie i zastosowanie,</li> </ul>	<ul style="list-style-type: none"> <li>• wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach,</li> <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 obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,</li> <li>• do implementacji rozszerzonego algorytmu Euklidesa stosuje zarówno iterację, jak i rekurencję,</li> </ul>	<p>zajmując punktowane miejsca,</p> <ul style="list-style-type: none"> <li>• w projektach zespołowych przyjmuje rolę lidera.</li> </ul>
---	---	--	--	---

<ul style="list-style-type: none"> <li>• uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania o niewielkim stopniu trudności.</li> </ul>	<ul style="list-style-type: none"> <li>• omawia rozszerzony algorytm Euklidesa,</li> <li>• formułuje algorytm wydawania reszty minimalną liczbą monet, harmonogramu wykorzystania sali, znajdowania drogi metodami zachłanną i dynamiczną,</li> <li>• uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania.</li> </ul>	<ul style="list-style-type: none"> <li>• stosuje metodę zachłanną w programach – problem kasjera, harmonogram wykorzystania sali, wyszukiwanie drogi,</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>• stosuje metody zachłanną i dynamiczną w problemach kasjera, harmonogramu wykorzystania sali i wyszukiwania drogi, wskazuje wady i zalety obu metod, szacuje złożoność czasową,</li> <li>• aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy.</li> </ul>	
--	--	--	--	--

Uwagi:

I. Ocenę wyższą otrzymuje uczeń spełniający łącznie wymagania edukacyjne określone dla ocen niższych np. ocenę dobrą otrzymuje uczeń spełniający wymagania edukacyjne na ocenę dopuszczającą, dostateczną oraz dobrą. W ramach przedmiotu: “Pracownia Aplikacji Mobilnych” uczeń powinien wykazać się umiejętnością wykorzystania wiedzy teoretycznej w praktyce.

II. Ocenę niedostateczną otrzymuje uczeń, który nie spełnia wymagań na poszczególne pozytywne oceny.



III. W przypadku niezrealizowania tematów lekcji (zagadnień) w I okresie będą one realizowane po klasyfikacji śródrocznej. W tym przypadku obowiązują również wymagania edukacyjne dla tych tematów (zagadnień).

IV. Tryb uzyskania oceny wyższej niż przewidywana: uczeń powinien spełnić wymagania określone w statucie szkoły:

*„1. Uczeń może starać się o uzyskanie wyższej niż przewidywana rocznej oceny klasyfikacyjnej z zajęć edukacyjnych w przypadku, gdy zgłosi w formie pisemnej Dyrektorowi liceum chęć uzyskania z danych zajęć edukacyjnych rocznej oceny klasyfikacyjnej wyższej niż przewidywana, wskazując tę ocenę.*

*2. Uczeń ubiegający się o wyższą ocenę z zajęć edukacyjnych niż przewidywana ocena roczna musi spełnić następujące warunki:*

- 1) uzyskał co najmniej 50% ocen bieżących takich, o jaką się ubiega (z prac pisemnych i odpowiedzi ustnych, a z wychowania fizycznego, plastyki, technologii informacyjnej – z wykonanych ćwiczeń);*
- 2) przystąpił do wszystkich sprawdzianów, kartkówek, prac pisemnych i uzyskał z nich ocenę pozytywną,*
- 3) doświadczył w br. szkolnym szczególnych sytuacji losowych (śmierć rodzica, wypadek, tragedia rodzinna, itp.),*
- 4) był obecny na 90% przeprowadzonych zajęć w ciągu roku oraz nie ma nieusprawiedliwionych nieobecności na danych zajęciach edukacyjnych.*

*3. Tryb postępowania:*

- 1) uczeń lub jego rodzic składa do nauczyciela przedmiotu, w terminie do 2 dni po otrzymaniu informacji o przewidywanej rocznej ocenie klasyfikacyjnej, pisemny wniosek wraz z uzasadnieniem i wskazaniem, o jaką ocenę się ubiega;*
- 2) nauczyciel w ciągu 1 dnia po otrzymaniu wniosku zobowiązany jest sprawdzić, czy uczeń spełnia powyższe warunki;*
- 3) w przypadku niespełnienia przez ucznia warunków nauczyciel pisemnie informuje, że nie ma podstaw do ustalenia wyższej oceny niż przewidywana (zapisuje decyzję na wniosku ucznia);*
- 4) w przypadku spełnienia przez ucznia w/w warunków nauczyciel:*
  - a) przypomina uczniowi wymagania na ocenę, o jaką się on ubiega,*

- b) przygotowuje zestaw zadań zgodnie z wymaganiami edukacyjnymi,
- c) uzgadnia termin przeprowadzenia sprawdzianu z uczniem i jego rodzicem (nie później niż w terminie 2 dni od dnia zgłoszenia zastrzeżeń),
- d) przygotowuje sprawdzian obejmujący zakres materiału podstawy programowej danej klasy i ma formę pisemną, a w przypadku języka polskiego i języków obcych sprawdzian ma formę pisemną i ustną oraz musi się odbyć przed klasyfikacyjnym posiedzeniem Rady Pedagogicznej,
- e) informuje, że w czasie trwania sprawdzianu w sali może być obecny rodzic lub inny nauczyciel danego przedmiotu w charakterze obserwatora,
- f) poprawia pracę w tym samym dniu, ogłasza wynik, informując ucznia czy uzyskał on ocenę o jaką się ubiegał, czy nie,
- g) ze sprawdzianu sporządza protokół, do którego dołącza prace pisemne, krótką informację na temat ustnych odpowiedzi ucznia/ wykonania przez ucznia zadań praktycznych;
- h) pełną dokumentację przechowuje do początku następnego roku szkolnego.

4. Sprawdzian wiadomości i umiejętności ucznia z informatyki, technologii informacyjnej i wychowania fizycznego ma przede wszystkim formę zadań praktycznych.

5. Uczeń uzyskuje z danych zajęć edukacyjnych roczną ocenę klasyfikacyjną wyższą niż przewidywana, jeżeli w wyniku sprawdzianu zaprezentował opanowanie wszystkich wiadomości i umiejętności odpowiadających wymaganiom edukacyjnym na daną ocenę, uwzględnionym na sprawdzianie przez nauczyciela.”