

**Rozkład materiału nauczania z Informatyki
oraz wymagania edukacyjne
w zakresie rozszerzonym dla liceum ogólnokształcącego i technikum – Informatyka na czasie, część 2
KLASA2**

W związku z uszczupleniem przez MEN podstawy programowej, w rozkładzie materiału zmniejszyła się liczba godzin na realizację obowiązkowych zagadnień. Uzyskane w ten sposób dodatkowe godziny pozostają do dyspozycji nauczyciela w trakcie roku szkolnego. Zgodnie z założeniami MEN: *Ograniczony zakres treści nauczania – wymagań szczegółowych – da nauczycielom i uczniom więcej czasu na spokojniejszą i bardziej dogłębną realizację programów nauczania.*

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
Rozdział 1. Algorytmy na liczbach całkowitych i tekstach					
1	Od problemu do programu	3	I.1, I.2, I.3, RI.3, II.1, RII.2, RI.1.1	<p>definiuje pojęcie specyfikacja algorytmu, określa dane i wyniki</p> <p>planuje kolejne kroki rozwiązania problemu</p> <p>omawia różne sposoby przedstawiania algorytmów (opis słowny, lista kroków, pseudokod)</p> <p>programuje i testuje rozwiązanie problemu</p> <p>sprawdza działanie algorytmów dla różnych danych</p> <p>tworzy algorytmy działania na liczbach całkowitych</p> <p>stosuje w języku C++ podstawowe konstrukcje programistyczne (operacje wejścia i wyjścia, instrukcja warunkowa, operatory matematyczne i logiczne)</p> <p>tworzy w języku C++ programy wykonujące działania na liczbach całkowitych</p>	<p>dobiera struktury i typy danych do rodzaju problemu</p> <p>wyszukuje optymalne rozwiązania problemów</p> <p>ocenia efektywność algorytmu</p> <p>objaśnia dobrany do danego problemu algorytm, uzasadnia jego poprawność i wybór</p>

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
2	Systemy liczbowe i reprezentacja danych w komputerze	3	I.1, I.3, RI.7, RI.8, II.1, RII.2	<p>definiuje pojęcie pozycyjnego systemu liczbowego</p> <p>wymienia systemy liczbowe stosowane w informatyce</p> <p>definiuje pojęcia bit i bajt</p> <p>dokonuje konwersji między pozycyjnymi systemami liczbowymi, wykorzystując przy tym zależności między systemami binarnym i ósemkowym oraz binarnym i heksadecymalnym</p> <p>omawia sposób reprezentowania liczb całkowitych w komputerze</p> <p>wymienia typy danych służące do zapisu liczb całkowitych (short int, int, long int, long long int, unsigned), stosuje je w pisanych programach</p> <p>opisuje, jak w komputerze reprezentowane są znaki i napisy (char, string), odwołuje się do znaku w napisie za pomocą indeksu</p> <p>wyjaśnia, czym jest tablica kodów ASCII</p> <p>omawia działanie operacji logicznych i reprezentację ich wyników w komputerze (wynik może przyjmować wartość prawda – 1 lub fałsz – 0, co zajmuje 1 bajt pamięci)</p>	wykonuje zadania o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku, z arkuszy maturalnych z lat poprzednich lub konkursów i olimpiad informatycznych
3	Algorytmy zamiany reprezentacji liczb między systemami liczbowymi	3	I.1, I.2, I.2a, I.3, RI.5, RI.6, II.1, RII.2, RIiII.1, RIiII.1i	tworzy programy do konwersji między liczbami w systemach binarnym i decymalnym	pisze programy zamieniające liczby z systemu decymalnego na system heksadecymalny

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
				<p>pisze programy konwertujące liczbę dziesiętną na liczbę w podanym systemie pozycyjnym</p> <p>posługuje się środowiskiem programistycznym, strukturami danych oraz językiem programowania w stopniu umożliwiającym implementację omawianych algorytmów</p> <p>stosuje binarną reprezentację liczby w algorytmie szybkiego podnoszenia do potęgi</p>	<p>pisze programy o podwyższonym stopniu trudności z wykorzystaniem algorytmów zamiany: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych</p> <p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p> <p>do rozwiązania problemu dobiera optymalny algorytm i struktury danych</p> <p>uzasadnia poprawność zaproponowanego rozwiązania</p> <p>korzysta z dostępnych bibliotek w tworzonych przez siebie programach</p> <p>tworzy własne funkcje rozwiązujące problemy</p>
4	Czy to jest palindrom?	2	I.1, I.2b, I.3, RI.1, II.1, RII.2	<p>definiuje pojęcie palindromu</p> <p>określa, czy dany napis lub liczba są palindromami</p> <p>wykonuje operacje na napisach (wczytywanie napisów ze spacjami, sprawdzanie długości napisu, zamiana liter dużych na małe i odwrotnie, porównywanie znaków, znajdowanie oraz usuwanie fragmentów napisów)</p>	<p>pisze programy dotyczące palindromów o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych</p> <p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p> <p>optymalizuje algorytmy i ocenia ich efektywność</p>

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
				<p>definiuje własne funkcje w języku C++, wyjaśnia celowość ich stosowania, rozróżnia parametry formalne i aktualne</p> <p>realizuje w języku C++ algorytmy sprawdzające, czy dany napis jest palindromem, oraz wyszukujące palindromy w zdaniach</p> <p>opisuje popularne funkcje oraz metody stosowane dla zmiennych typu string (toupper, tolower, size, substr, erase)</p>	
5	Czy ta liczba jest pierwsza?	3	I.1, I.2, I.2a, I.3, RI.2, RI.3, RI.5, II.1, RII.2, RI.1, RI.2a	<p>wymienia podstawowe własności liczb pierwszych</p> <p>sprawdza, czy dana liczba jest pierwsza, stosując algorytm naiwny</p> <p>rozkłada liczbę złożoną na czynniki pierwsze</p> <p>wyznacza liczby bliźniacze</p>	<p>implementuje algorytmy dotyczące liczb pierwszych o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych</p> <p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p> <p>stosuje optymalny algorytm sprawdzający, czy liczba jest pierwsza, wykorzystując funkcję logiczną; uzasadnia jego efektywność</p> <p>pisze program rozkładający liczbę złożoną na sumę dwóch liczb pierwszych (hipoteza Goldbacha)</p>
6	Działania na liczbach w systemach innych niż dziesiętny	3	I.1, I.2, I.3, RI.3, RI.6, RI.8, II.1, RII.2, RI.1, RI.2b	wykonuje działania arytmetyczne na liczbach w różnych systemach pozycyjnych	wykonuje działania o podwyższonym stopniu trudności

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
				<p>wykonuje obliczenia na dowolnie dużych liczbach, wykorzystując napisy</p> <p>wyjaśnia różnicę między operacjami na liczbach o podstawie od 1 do 9 i większej od 10</p> <p>stosuje odejmowanie w dzieleniu pisemnym liczb binarnych</p> <p>stosuje dodawanie liczby przeciwnej zapisanej w kodzie U2 przy odejmowaniu liczby binarnej</p>	<p>pisze programy wykonujące operacje arytmetyczne na liczbach w różnych systemach pozycyjnych</p> <p>optymalizuje programy, szacuje ich efektywność</p>
7	Algorytm Euklidesa i działania na ułamkach	3	I.1, I.2, I.2a, I.3, RI.2, RI.3, RI.5, RI.6, RI.10, II.1, RII.2, RI.11.1, RI.11.1a	<p>opisuje geometryczną interpretację algorytmu Euklidesa</p> <p>pisze program realizujący algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem, stosując funkcję typu void</p> <p>stosuje strukturę do reprezentacji liczb wymiernych</p> <p>wykorzystuje algorytm Euklidesa do działań na ułamkach</p> <p>stosuje zmienne lokalne i globalne, a także przekazywanie parametrów przez wartość</p>	<p>pisze programy o podwyższonym stopniu trudności prezentujące zastosowanie algorytmu Euklidesa</p> <p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p> <p>stosuje funkcje i dobiera sposób przekazywania parametrów, jednocześnie go uzasadniając</p>
8	Szyfr Cezara i inne szyfry podstawieniowe	3	I.1, I.2b, I.3, RI.3, II.1, RII.2	<p>definiuje szyfry: podstawieniowy, monoalfabetyczny i permutacyjny, wymienia przykłady takich szyfrów</p> <p>pisze program szyfrujący informację szyfrem Cezara z wykorzystaniem liter z polskimi znakami diakrytycznymi</p>	<p>pisze programy szyfrujące o podwyższonym stopniu trudności</p> <p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p>

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
				omawia szyfr Vigenere'a stosuje w swoich programach operacje plikowe – wczytywanie danych z pliku dyskowego, zapis wyniku do pliku	wykorzystuje odpowiednio dobrane struktury danych korzysta z funkcji bibliotecznych tworzy własne funkcje, dobierając sposób przekazywania parametrów
W	Wiesz, umiesz, zdasz	4	I.1, I.2, I.2a, I.2b, I.3, RI.2, RI.3, RI.5, RI.7, II.1, RI.11.1		
Rozdział 2. Rozwiązywanie problemów z wykorzystaniem struktur danych					
9	Łamiemy szyfr Cezara	3	I.1, I.2b, I.3, RI.3, RI.10, II.1, RI.2	wyjaśnia, na czym polega łamanie szyfru (kryptoanaliza) łamie szyfr Cezara, stosując analizę częstości stosuje algorytmy zliczające liczbę wystąpień znaków w tekście z zastosowaniem strukturalnego typu danych – tablic pisze program znajdujący maksimum w tablicy i wypisujący jego pozycję (algorytm „dziel i zwyciężaj”)	opisuje różne sposoby łamania szyfrów i implementuje je w języku C++ pisze programy deszyfrujące o podwyższonym poziomie trudności
10	Poszukujemy liczby	2	I.1, I.2, I.3, RI.1, RI.2, RI.3, RI.4, RI.5, RI.6, RI.10, II.1, RI.2, RI.11.1, RI.11.1b, RI.11.1d, RI.11.3a, RI.11.3c	znajduje wartość w zbiorach uporządkowanym i nieuporządkowanym, stosując odpowiednio algorytmy wyszukiwania liniowego, liniowego z wartownikiem i binarnego pisze programy wykorzystujące przekazywanie parametru do funkcji przez wskaźnik i referencję	pisze programy o podwyższonym stopniu trudności szacuje złożoność czasową zastosowanych algorytmów wyszukiwania wyjaśnia na przykładach różnice między różnymi sposobami przekazywania parametrów do funkcji

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
11				stosuje algorytm „dziel i zwyciężaj” do jednoczesnego znajdowania maksimum i minimum w zbiorze	podaje wzór na liczbę wykonywanych operacji w algorytmie „dziel i zwyciężaj”
	Jak ocenić złożoność obliczeniową algorytmu?	2	RI.2, RI.5, RI.6, RI.10,	definiuje złożoność obliczeniową algorytmu szacuje złożoność czasową i pamięciową wyjaśnia, czym jest złożoność oczekiwana (średnia), optymistyczna i pesymistyczna	określa złożoność czasową i pamięciową algorytmów z zastosowaniem odpowiednich wzorów rozdzieli pojęcia algorytmu naiwnego i optymalnego ocenia efektywność algorytmów
12	Metody sortowania prostego	3	I.1, I.2, I.2c, I.3, RI.2, RI.3, RI.5, RI.6, RI.10, II.1, RII.2, RI.11.1, RI.11.3a	definiuje pojęcie sortowania, prawidłowo określając klucz i porządek sortowania definiuje pojęcia sortowania <i>in situ</i> i stabilnego stosuje metody sortowania prostego do sortowania liczb w zbiorze – bąbelkowe i przez wybieranie szacuje złożoność obliczeniową stosowanych algorytmów definiuje operacje kluczowe (dominujące) w algorytmach sortowania pisze programy realizujące poznane algorytmy sortowania	pisze programy sortujące o podwyższonym stopniu trudności: sortowanie danych w plikach tekstowych, sortowanie struktur podaje przykłady sortowania prostego w życiu codziennym dobiera właściwe struktury danych definiuje własne funkcje do rozwiązywania problemów z wykorzystaniem algorytmów sortowania ocenia wpływ pierwotnego ułożenia danych w zbiorze na liczbę wykonywanych operacji
13	Sito Eratostenesa	2	I.1, I.3, RI.3, RI.5, RI.6; RI.10, II.1, RII.2, RI.11.1c	opisuje algorytmy sprawdzające, czy liczba jest pierwsza	pisze programy o podwyższonym stopniu trudności wykorzystujące sito Eratostenesa

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
				omawia i stosuje algorytm sita Eratostenesa do wyszukiwania liczb pierwszych w określonym przedziale liczbowym określa złożoność obliczeniową algorytmu	posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym optymalizuje algorytm, dążąc do minimalnej złożoności obliczeniowej
14	Szukamy różnych podciągów	4	I.1, RI.2, I.3, RI.3, RI.5, RI.6, RI.10, II.1, RII.2, RI.11.2c	definiuje pojęcia podciągu oraz podciągu spójnego znajduje w zbiorze podciągi o różnych własnościach oblicza długość najdłuższego niemalejącego spójnego podciągu oraz liczbę jego elementów wymienia i stosuje różne algorytmy znajdowania maksymalnej sumy elementów spójnych podciągów, oceniając ich złożoność obliczeniową znajduje w zbiorze spójny podciąg o maksymalnej sumie i wypisuje jego elementy	pisze programy o podwyższonym stopniu trudności wyszukujące spójne podciągi posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym do rozwiązania problemu dobiera optymalny algorytm i struktury danych
W	Wiesz, umiesz, zdasz	4	I.1, I.2, I.2a, I.2b, I.2c, I.3, RI.3, RI.4, II.1, RII.2, RI.11.2c, RI.11.3a		
Rozdział 3. Metody algorytmiczne					
15	Iteracja a rekurencja	4	I.1, I.3, RI.1, RI.2, RI.3, RI.4, RI.5, RI.10, II.1, RII.2, RI.11.1a, RI.11.1i, RI.11.3b	opisuje zasadę działania rekurencji implementuje w języku C++ algorytmy rekurencyjne, określa warunki brzegowe	pisze programy o podwyższonym stopniu trudności, np. sprawdzanie hipotezy Collata

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
				<p>porównuje iteracyjne i rekurencyjne wersje algorytmów</p> <p>opisuje zasadę złotego podziału</p> <p>oblicza n-ty wyraz ciągu Fibonacciego metodami iteracyjną i rekurencyjną</p> <p>wyjaśnia, na czym polega rozszerzony algorytm Euklidesa, oraz implementuje go w języku C++</p>	<p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p> <p>do rozwiązania problemu dobiera optymalny algorytm i struktury danych</p> <p>uzasadnia wybór iteracji lub rekurencji do rozwiązania problemu</p> <p>szacuje złożoność czasową stosowanych algorytmów</p> <p>oblicza liczbę wykonywanych operacji w algorytmach rekurencyjnych</p>
16	Metoda zachłanna	4	I.1, I.3, RI.1, RI.2, RI.3, RI.4, RI.10, II.1, RII.2, RI.3d	<p>wyjaśnia, na czym polega metoda zachłanna, i wymienia przykłady jej stosowania</p> <p>implementuje następujące algorytmy zachłanne: problem kasjera (wydawania reszty minimalną liczbą nominałów), problem telewizza/kinomana (optymalny harmonogram wykorzystania sali), wyszukiwanie optymalnej drogi</p> <p>ocenia przydatność zastosowanych algorytmów</p> <p>stosuje własne kryterium porównania w funkcji sort z biblioteki STL</p>	<p>pisze programy o podwyższonym stopniu trudności z wykorzystaniem algorytmów zachłannych, stosując rekurencję i algorytmy z nawrotami</p> <p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p> <p>do rozwiązania problemu dobiera optymalny algorytm i struktury danych</p> <p>objaśnia algorytm wybrany do rozwiązania problemu oraz ocenia jego efektywność i niedoskonałość</p>
17	Programowanie dynamiczne	5	I.1, I.3, RI.2, RI.3, RI.4, RI.5, RI.10, II.1, RII.2, RI.2c, RI.3e	<p>wyjaśnia, na czym polega metoda dynamiczna</p>	<p>pisze programy o podwyższonym stopniu trudności wykorzystujące algorytmy dynamiczne</p>

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej		
				Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
				<p>implementuje optymalne algorytmy dotyczące problemu kasjera, telewidza, znajdowania drogi</p> <p>stosuje metodę dynamiczną do znajdowania najdłuższego wspólnego podciągu</p> <p>porównuje metody zachłanną i dynamiczną</p>	<p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p> <p>do rozwiązania problemu dobiera optymalny algorytm i struktury danych</p>
18	Dziel i zwyciężaj, czyli sortujemy sprawniej	4	I.1, I.3, RI.1, RI.2, RI.3, RI.4, RI.5, RI.6, RI.10, II.1, RII.2, RI.11.1e, RI.11.3b, RI.11.3c	<p>omawia metodę „dziel i zwyciężaj” oraz rekurencję</p> <p>wyjaśnia, na czym polega algorytm sortowania szybkiego oraz przez scalanie i implementuje je</p> <p>ocenia i porównuje złożoność czasową i obliczeniową algorytmów</p>	<p>pisze programy o podwyższonym stopniu trudności wykorzystujące metodę „dziel i zwyciężaj” oraz algorytmy sortowania</p> <p>posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</p> <p>do rozwiązania problemu dobiera optymalny algorytm i struktury danych, a także korzysta z funkcji bibliotecznych</p>
W	Wiesz, umiesz, zdasz	4	I.1, I.2b, I.3, RI.3, RI.4, II.1, RII.2		
P	Programowanie zespołowe – projekt zespołowy	3	I.1, I.3, RI.3, RI.4, RI.5, RI.10, II.1, II.2, RII.1, RII.2, RII.3b, RIV.1	<p>wyjaśnia, czym jest dokumentacja projektu (projektowa, użytkownika, techniczna), bierze czynny udział w jej tworzeniu</p> <p>aktywnie uczestniczy w realizacji projektu</p> <p>przyjmuje różne role w zespole realizującym projekt</p> <p>prezentuje efekty wspólnej pracy</p>	<p>przyjmuje rolę lidera odpowiedzialnego za zespół i projekt</p> <p>przydziela zadania, nadzoruje pracę innych opracowując złożone problemy, posługuje się aplikacjami w stopniu zaawansowanym</p>
Suma godzin: 71					

Wymagania edukacyjne z Informatyki – KLASA 2-ZR:

Ocenę celującą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- 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,
- pisze program rozkładający liczbę złożoną na dwie liczby pierwsze (hipoteza Goldbacha),
- implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję,
- pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere'a) i różnych kluczy,
- implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej,
- pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary),
- stosuje zaawansowane algorytmy i struktury danych do wyszukiwania spójnych podciągów,
- stosuje zaawansowane algorytmy wyszukiwania, np. najlepszego wyboru (trwałych par), stosując rekurencję,
- pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,
- stosuje w programach algorytmy sortowania inne niż omawiane na lekcjach (np. heapsort),
- bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- w projektach zespołowych przyjmuje rolę lidera.

Ocenę bardzo dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- 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 pozycyjnych wykorzystuje bibliotekę string i strukturalne typy danych,
- 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,
- 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 sortujące, zmieniając porządek sortowania,
- 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 obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,
- 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,
- aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy.

Ocenę dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- określa specyfikację algorytmu (dane, wynik),
- pisze programy o różnym stopniu trudności, szacuje ich efektywność,
- przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, 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 liczbach w różnych systemach,
- w algorytmach zamiany wykorzystuje zależności między systemami binarnym, ósemkowym i heksadecymalnym,
- 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, wykorzystując funkcję sort z biblioteki STL,
- 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 zastosowaniem funkcji typu void,
- stosuje algorytmy sortowania szybkiego i przez scalanie,
- pisze program realizujący algorytm sita Eratostenesa,
- implementuje w języku C++ algorytmy wyszukujące spójne podciągi o różnych cechach,

- porównuje algorytmy iteracyjne i rekurencyjne (liczbę wykonywanych operacji), szacuje ich złożoność czasową,
- zapisuje w postaci programu rozszerzony algorytm Euklidesa, wyjaśnia jego działanie i zastosowanie,
- stosuje metodę zachłanną w programach – problem kasjera, harmonogram wykorzystania sali, wyszukiwanie drogi,
- uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej.

Ocenę dostateczną otrzymuje uczeń, który spełnia kryteria oceny dopuszczającej, a ponadto:

- przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu,
- dodaje 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 C++: short int, int, long int, long long int,
- 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 między zmiennymi lokalną i globalną,
- 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 algorytmy sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie, określa operacje dominujące,
- omawia algorytm sita Eratostenesa,
- przedstawia algorytmy znajdowania spójnych podciągów, wyznaczania najdłuższego z nich oraz podciągu o największej sumie elementów,
- implementuje w języku C++ algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi,
- omawia rozszerzony algorytm Euklidesa,
- formułuje algorytm wydawania reszty minimalną liczbą monet, harmonogramu wykorzystania sali, znajdowania drogi metodami zachłanną i dynamiczną,
- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania.

Ocenę dopuszczającą otrzymuje uczeń, który:

- 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 funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych,
- 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,
- 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”,
- 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,
- wyszukuje w ciągu liczb spójne podciągi (nierosnący, niemalejący, stały), wskazuje najdłuższe, oblicza ich sumę,
- definiuje pojęcia iteracji i rekurencji,
- 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,

- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania o niewielkim stopniu trudności.

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.