

INFORMATYKA klasa 2 LO (4-letniego) poziom rozszerzony

Rozkładu materiału nauczania informatyki w zakresie rozszerzonym dla liceum ogólnokształcącego – *Informatyka na czasie*, część 2

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej
Rozdział 1. Algorytmy na liczbach całkowitych i tekstach			
1	Od problemu do programu	3	I.1, I.5, II.2
2	Systemy liczbowe i reprezentacja danych w komputerze	3	I.2a, II.1, RI.5
3	Algorytmy zamiany reprezentacji liczb między systemami liczbowymi	3	I.2a, I.5, II.1, RI.3, RII.3, RIIII.1i
4	Czy to jest palindrom?	2	I.1, I.2b, I.5, II.1, II.2
5	Czy ta liczba jest pierwsza?	3	I.2a, I.5, II.1, RI.3, RII.3, RIIII.2a
6	Działania na liczbach w systemach innych niż dziesiętny	3	RI.1, RI.3, RII.3, RIIII.2b
7	Algorytm Euklidesa i działania na ułamkach	3	I.2a, I.5, II.1, RIIII.1a
8	Szyfr Cezara i inne szyfry podstawieniowe	3	I.1, I.2b, I.5, II.1, II.2
A	Uzupełnienie wiadomości i ćwiczeń		I.1, I.2a, I.2b, I.5, II.1, II.2, RI.1, RI.3, RI.5, RII.3, RIIII.1i, RIIII.2a, RIIII.2b, RIIII.3b
Rozdział 2. Rozwiązywanie problemów z wykorzystaniem struktur danych			
9	Łamiemy szyfr Cezara	3	I.2b, RII.1
10	Poszukujemy liczby	2	RI.3, RI.4, RII.3, RIIII.1b, RIIII.3a, RIIII.3c
11	Jak ocenić złożoność obliczeniową algorytmu?	4	RI.3, RI.4, RII.3, RIIII.1d
12	Metody sortowania prostego	3	I.1, I.2c, I.5, II.1, II.2
13	Szyfry przestawieniowe, anagramy	2	I.1, I.2b, I.5, II.1, II.2
14	Sito Eratostenesa	2	RI.2, RI.3, RII.3, RIIII.1c
15	Szukamy różnych podciągów	4	RI.3, RII.3, RIIII.2c
16	W poszukiwaniu lidera i idola	2	RI.3, RII.3, RIIII.1b
A	Uzupełnienie wiadomości i ćwiczeń	4	I.1, I.2b, I.2c, I.5, II.1, II.2, RI.2, RI.3, RI.4, RII.1, RII.3, RIIII.1b, RIIII.1c, RIIII.3a, RIIII.2c, RIIII.3c
Rozdział 3. Metody algorytmiczne			
17	Iteracja a rekurencja	4	I.2e, I.3, I.4, II.1, II.2, RIIII.1a, RIIII.3b
18	Metoda zachłanna	3	I.1, I.2d, I.3, I.5, II.1, II.2
19	Programowanie dynamiczne	5	RI.2, RI.3, RII.3, RIIII.3d, RIIII.3e
20	Dziel i zwyciężaj, czyli sortujemy sprawniej	4	RI.1, RI.2, RI.3, RI.4, RII.2, RII.3, RIIII.1e, RIIII.3b, RIIII.3c
A	Uzupełnienie wiadomości i ćwiczeń	4	I.1, I.2d, I.2e, I.3, I.4, I.5, II.1, II.2, RI.1, RI.2, RI.3, RI.4, RII.2, RII.3, RIIII.1a, RIIII.1d, RIIII.1e, RIIII.3b, RIIII.3c, RIIII.3d, RIIII.3e
P	Programowanie zespołowe	3	RI.1, RI.2, RI.3, RII.1, RII.2, RII.3, RII.4b, RIV.1, RIV.3

Plan wynikowy

Lp.	Temat	Liczba godzin	Osiągnięcia uczniów	
			Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
1	Od problemu do programu	3	<ul style="list-style-type: none"> <input type="checkbox"/> definiuje pojęcie specyfikacja algorytmu, określa dane i wyniki <input type="checkbox"/> planuje kolejne kroki rozwiązania problemu <input type="checkbox"/> omawia różne sposoby przedstawiania algorytmów (opis słowny, lista kroków, schemat blokowy, pseudokod) <input type="checkbox"/> programuje i testuje rozwiązanie problemu <input type="checkbox"/> sprawdza działanie algorytmów dla różnych danych <input type="checkbox"/> tworzy algorytmy działania na liczbach całkowitych <input type="checkbox"/> stosuje w języku C++ podstawowe konstrukcje programistyczne (operacje wejścia i wyjścia, instrukcja warunkowa, operatory matematyczne i logiczne) <input type="checkbox"/> tworzy w języku C++ programy wykonujące działania na liczbach całkowitych 	<ul style="list-style-type: none"> <input type="checkbox"/> dobiera struktury i typy danych do rodzaju problemu <input type="checkbox"/> wyszukuje optymalne rozwiązania problemów <input type="checkbox"/> ocenia efektywność algorytmu <input type="checkbox"/> objaśnia dobrany do danego problemu algorytm, uzasadnia jego poprawność i wybór
2	Systemy liczbowe i reprezentacja danych w komputerze	3	<ul style="list-style-type: none"> <input type="checkbox"/> definiuje pojęcie pozycyjnego systemu liczbowego <input type="checkbox"/> wymienia systemy liczbowe stosowane w informatyce <input type="checkbox"/> definiuje pojęcia bit i bajt <input type="checkbox"/> dokonuje konwersji między pozycyjnymi systemami liczbowymi, wykorzystując przy tym zależności między systemami binarnym i ósemkowym oraz binarnym i heksadecymalnym <input type="checkbox"/> omawia sposób reprezentowania liczb całkowitych w komputerze <input type="checkbox"/> wymienia typy danych służące do zapisu liczb całkowitych (short int, int, long int, long long int, unsigned), stosuje je w pisanych programach <input type="checkbox"/> opisuje, jak w komputerze reprezentowane są znaki i napisy (char, string), odwołuje się do znaku w napisie za pomocą indeksu <input type="checkbox"/> wyjaśnia, czym jest tablica kodów ASCII <input type="checkbox"/> omawia działanie operacji logicznych i reprezentację ich wyników w komputerze (wynik może przyjmować wartość prawda – 1 lub fałsz – 0, co zajmuje 1 bajt pamięci) <input type="checkbox"/> opisuje istotę cyfrowej reprezentacji w komputerze obrazów, dźwięków i animacji 	<ul style="list-style-type: none"> <input type="checkbox"/> 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	<ul style="list-style-type: none"> <input type="checkbox"/> tworzy programy do konwersji między liczbami w systemach binarnym i decymalnym <input type="checkbox"/> pisze programy konwertujące liczbę dziesiętną na liczbę w podanym systemie pozycyjnym <input type="checkbox"/> posługuje się środowiskiem programistycznym, strukturami danych oraz językiem programowania w stopniu umożliwiającym implementację omawianych algorytmów <input type="checkbox"/> stosuje binarną reprezentację liczby w algorytmie szybkiego podnoszenia do potęgi 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy zamieniające liczby z systemu decymalnego na system heksadecymalny <input type="checkbox"/> 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 <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym do rozwiązania problemu dobiera optymalny algorytm i struktury danych <input type="checkbox"/> uzasadnia poprawność zaproponowanego rozwiązania <input type="checkbox"/> korzysta z dostępnych bibliotek w tworzonych przez siebie programach <input type="checkbox"/> tworzy własne funkcje rozwiązujące problemy
4	Czy to jest palindrom?	2	<ul style="list-style-type: none"> <input type="checkbox"/> definiuje pojęcie palindromu <input type="checkbox"/> określa, czy dany napis lub liczba są palindromami <input type="checkbox"/> 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) <input type="checkbox"/> definiuje własne funkcje w języku C++, wyjaśnia celowość ich stosowania, rozróżnia parametry formalne i aktualne <input type="checkbox"/> realizuje w języku C++ algorytmy sprawdzające, czy dany napis jest palindromem, oraz wyszukujące palindromy w zdaniach <input type="checkbox"/> opisuje popularne funkcje oraz metody stosowane dla zmiennych typu string (toupper, tolower, size, substr, erase) 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy dotyczące palindromów o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> optymalizuje algorytmy i ocenia ich efektywność
5	Czy ta liczba jest pierwsza?	3	<ul style="list-style-type: none"> <input type="checkbox"/> wymienia podstawowe własności liczb pierwszych <input type="checkbox"/> sprawdza, czy dana liczba jest pierwsza, stosując algorytm naiwny <input type="checkbox"/> rozkłada liczbę złożoną na czynniki pierwsze <input type="checkbox"/> wyznacza liczby bliźniacze 	<ul style="list-style-type: none"> <input type="checkbox"/> implementuje algorytmy dotyczące liczb pierwszych o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> stosuje optymalny algorytm sprawdzający, czy liczba jest pierwsza, wykorzystując funkcję logiczną; uzasadnia jego efektywność <input type="checkbox"/> pisze program rozkładający liczbę złożoną na sumę dwóch liczb pierwszych (hipoteza Goldbacha)

6	Działania na liczbach w systemach innych niż dziesiętne	3	<ul style="list-style-type: none"> <input type="checkbox"/> wykonuje działania arytmetyczne na liczbach w różnych systemach pozycyjnych <input type="checkbox"/> wykonuje obliczenia na dowolnie dużych liczbach, wykorzystując napisy <input type="checkbox"/> wyjaśnia różnicę między operacjami na liczbach o podstawie od 1 do 9 i większej od 10 <input type="checkbox"/> stosuje odejmowanie w dzieleniu pisemnym liczb binarnych <input type="checkbox"/> stosuje dodawanie liczby przeciwnej zapisanej w kodzie U2 przy odejmowaniu liczby binarnej 	<ul style="list-style-type: none"> <input type="checkbox"/> wykonuje działania o podwyższonym stopniu trudności <input type="checkbox"/> pisze programy wykonujące operacje arytmetyczne na liczbach w różnych systemach pozycyjnych <input type="checkbox"/> optymalizuje programy, szacuje ich efektywność
7	Algorytm Euklidesa i działania na ułamkach	3	<ul style="list-style-type: none"> <input type="checkbox"/> opisuje geometryczną interpretację algorytmu Euklidesa <input type="checkbox"/> pisze program realizujący algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem, stosując funkcję typu void <input type="checkbox"/> stosuje strukturę do reprezentacji liczb wymiernych <input type="checkbox"/> wykorzystuje algorytm Euklidesa do działań na ułamkach <input type="checkbox"/> stosuje zmienne lokalne i globalne, a także przekazywanie parametrów przez wartość 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności prezentujące zastosowanie algorytmu Euklidesa <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> stosuje funkcje i dobiera sposób przekazywania parametrów, jednocześnie go uzasadniając
8	Szyfr Cezara i inne szyfry podstawieniowe	3	<ul style="list-style-type: none"> <input type="checkbox"/> definiuje szyfry: podstawieniowy, monoalfabetyczny i permutacyjny, wymienia przykłady takich szyfrów <input type="checkbox"/> pisze program szyfrujący informację szyfrem Cezara z wykorzystaniem liter z polskimi znakami diakrytycznymi <input type="checkbox"/> omawia szyfr Vigenere'a <input type="checkbox"/> stosuje w swoich programach operacje plikowe – wczytywanie danych z pliku dyskowego, zapis wyniku do pliku 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy szyfrujące o podwyższonym stopniu trudności <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> wykorzystuje odpowiednio dobrane struktury danych <input type="checkbox"/> korzysta z funkcji bibliotecznych <input type="checkbox"/> tworzy własne funkcje, dobierając sposób przekazywania parametrów
9	Łamiemy szyfr Cezara	3	<ul style="list-style-type: none"> <input type="checkbox"/> wyjaśnia, na czym polega łamanie szyfru (kryptoanaliza) <input type="checkbox"/> łamie szyfr Cezara, stosując analizę częstości <input type="checkbox"/> stosuje algorytmy zliczające liczbę wystąpień znaków w tekście z zastosowaniem strukturalnego typu danych – tablic <input type="checkbox"/> pisze program znajdujący maksimum w tablicy i wypisujący jego pozycję (algorytm „dziel i zwyciężaj”) 	<ul style="list-style-type: none"> <input type="checkbox"/> opisuje różne sposoby łamania szyfrów i implementuje je w języku C++ <input type="checkbox"/> pisze programy deszyfrujące o podwyższonym poziomie trudności
10	Poszukujemy liczby	2	<ul style="list-style-type: none"> <input type="checkbox"/> znajduje wartość w zbiorach uporządkowanym i nieuporządkowanym, stosując odpowiednio algorytmy wyszukiwania liniowego, liniowego z wartownikiem i binarnego <input type="checkbox"/> pisze programy wykorzystujące przekazywanie parametru do funkcji przez wskaźnik i referencję 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności <input type="checkbox"/> szacuje złożoność czasową zastosowanych algorytmów wyszukiwania <input type="checkbox"/> wyjaśnia na przykładach różnice między różnymi sposobami przekazywania parametrów do funkcji <input type="checkbox"/> podaje wzór na liczbę wykonywanych operacji w algorytmie „dziel i zwyciężaj”

			<ul style="list-style-type: none"> <input type="checkbox"/> stosuje algorytm „dziel i zwyciężaj” do jednoczesnego znajdowania maksimum i minimum w zbiorze 	
11	Jak ocenić złożoność obliczeniową algorytmu?	4	<ul style="list-style-type: none"> <input type="checkbox"/> definiuje złożoność obliczeniową algorytmu <input type="checkbox"/> szacuje złożoność czasową i pamięciową <input type="checkbox"/> wyjaśnia, czym jest złożoność oczekiwana (średnia), optymistyczna i pesymistyczna 	<ul style="list-style-type: none"> <input type="checkbox"/> określa złożoność czasową i pamięciową algorytmów z zastosowaniem odpowiednich wzorów <input type="checkbox"/> rozróżnia pojęcia algorytmu naiwnego i optymalnego <input type="checkbox"/> ocenia efektywność algorytmów
12	Metody sortowania prostego	3	<ul style="list-style-type: none"> <input type="checkbox"/> definiuje pojęcie sortowania, prawidłowo określając klucz i porządek sortowania <input type="checkbox"/> definiuje pojęcia sortowania <i>in situ</i> i stabilnego <input type="checkbox"/> stosuje metody sortowania prostego do sortowania liczb w zbiorze – bąbelkowe i przez wybieranie <input type="checkbox"/> szacuje złożoność obliczeniową stosowanych algorytmów <input type="checkbox"/> definiuje operacje kluczowe (dominujące) w algorytmach sortowania <input type="checkbox"/> pisze programy realizujące poznane algorytmy sortowania 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy sortujące o podwyższonym stopniu trudności: sortowanie danych w plikach tekstowych, sortowanie struktur <input type="checkbox"/> podaje przykłady sortowania prostego w życiu codziennym <input type="checkbox"/> dobiera właściwe struktury danych <input type="checkbox"/> definiuje własne funkcje do rozwiązywania problemów z wykorzystaniem algorytmów sortowania <input type="checkbox"/> ocenia wpływ pierwotnego ułożenia danych w zbiorze na liczbę wykonywanych operacji
13	Szyfry przestawieniowe, anagramy	2	<ul style="list-style-type: none"> <input type="checkbox"/> omawia zasadę działania szyfrów przestawieniowych, wymienia przykłady takich szyfrów <input type="checkbox"/> sprawdza, czy słowa (napisy) są anagramami <input type="checkbox"/> pisze funkcje sprawdzające <input type="checkbox"/> wykorzystuje poznane wcześniej algorytmy sortowania i zliczania w rozwiązywaniu problemów 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy szyfrujące o podwyższonym stopniu trudności <input type="checkbox"/> wyszukuje anagramy w plikach tekstowych <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> do rozwiązania problemu dobiera optymalny algorytm i struktury danych
14	Sito Eratostenesa	2	<ul style="list-style-type: none"> <input type="checkbox"/> opisuje algorytmy sprawdzające, czy liczba jest pierwsza <input type="checkbox"/> omawia i stosuje algorytm sita Eratostenesa do wyszukiwania liczb pierwszych w określonym przedziale liczbowym <input type="checkbox"/> określa złożoność obliczeniową algorytmu 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności wykorzystujące sito Eratostenesa <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> optymalizuje algorytm, dążąc do minimalnej złożoności obliczeniowej
15	Szukamy różnych podciągów	4	<ul style="list-style-type: none"> <input type="checkbox"/> definiuje pojęcia podciągu oraz podciągu spójnego <input type="checkbox"/> znajduje w zbiorze podciągi o różnych własnościach <input type="checkbox"/> oblicza długość najdłuższego niemalejącego spójnego podciągu oraz liczbę jego elementów <input type="checkbox"/> wymienia i stosuje różne algorytmy znajdowania maksymalnej sumy elementów spójnych podciągów, oceniając ich złożoność obliczeniową <input type="checkbox"/> znajduje w zbiorze spójny podciąg o maksymalnej sumie i wypisuje jego elementy 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności wyszukujące spójne podciągi <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> do rozwiązania problemu dobiera optymalny algorytm i struktury danych

16	W poszukiwaniu lidera i idola	2	<ul style="list-style-type: none"> <input type="checkbox"/> definiuje pojęcia idola w grupie i lidera w zbiorze <input type="checkbox"/> znajduje idola w grupie lub stwierdza jego brak <input type="checkbox"/> określa, czy w zbiorze jest lider <input type="checkbox"/> omawia i implementuje w języku C++ algorytmy szukania idola oraz lidera <input type="checkbox"/> ocenia złożoność obliczeniową stosowanych algorytmów i ich efektywność <input type="checkbox"/> stosuje tablice dwuwymiarowe w pisanych programach <input type="checkbox"/> stosuje funkcję sort z biblioteki STL do wyszukiwania lidera 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> do rozwiązania problemu dobiera optymalny algorytm i struktury danych
17	Iteracja a rekurencja	4	<ul style="list-style-type: none"> <input type="checkbox"/> opisuje zasadę działania rekurencji <input type="checkbox"/> implementuje w języku C++ algorytmy rekurencyjne, określa warunki brzegowe <input type="checkbox"/> porównuje iteracyjne i rekurencyjne wersje algorytmów <input type="checkbox"/> opisuje zasadę złotego podziału <input type="checkbox"/> oblicza n-ty wyraz ciągu Fibonacciego metodami iteracyjną i rekurencyjną <input type="checkbox"/> wyjaśnia, na czym polega rozszerzony algorytm Euklidesa, oraz implementuje go w języku C++ 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności, np. sprawdzanie hipotezy Collatza <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> do rozwiązania problemu dobiera optymalny algorytm i struktury danych <input type="checkbox"/> uzasadnia wybór iteracji lub rekurencji do rozwiązania problemu <input type="checkbox"/> szacuje złożoność czasową stosowanych algorytmów <input type="checkbox"/> oblicza liczbę wykonywanych operacji w algorytmach rekurencyjnych
18	Metoda zachłanna	3	<ul style="list-style-type: none"> <input type="checkbox"/> wyjaśnia, na czym polega metoda zachłanna, i wymienia przykłady jej stosowania <input type="checkbox"/> implementuje następujące algorytmy zachłanne: problem kasjera (wydawania reszty minimalną liczbą nominałów), problem telewizja/kinomana (optymalny harmonogram wykorzystania sali), pakowanie plecaka, wyszukiwanie optymalnej drogi <input type="checkbox"/> ocenia przydatność zastosowanych algorytmów <input type="checkbox"/> stosuje własne kryterium porównania w funkcji sort z biblioteki STL 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności z wykorzystaniem algorytmów zachłannych, stosując rekurencję i algorytmy z nawrotami <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> do rozwiązania problemu dobiera optymalny algorytm i struktury danych <input type="checkbox"/> objaśnia algorytm wybrany do rozwiązania problemu oraz ocenia jego efektywność i niedoskonałość
19	Programowanie dynamiczne	5	<ul style="list-style-type: none"> <input type="checkbox"/> wyjaśnia, na czym polega metoda dynamiczna <input type="checkbox"/> implementuje optymalne algorytmy dotyczące problemu kasjera, telewizja, znajdowania drogi oraz pakowania plecaka <input type="checkbox"/> stosuje metodę dynamiczną do znajdowania najdłuższego wspólnego podciągu <input type="checkbox"/> porównuje metody zachłanną i dynamiczną 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności wykorzystujące algorytmy dynamiczne <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> do rozwiązania problemu dobiera optymalny algorytm i struktury danych

20	Dziel i zwyciężaj, czyli sortujemy sprawniej	4	<ul style="list-style-type: none"> <input type="checkbox"/> omawia metodę „dziel i zwyciężaj” oraz rekurencję <input type="checkbox"/> wyjaśnia, na czym polega algorytm sortowania szybkiego oraz przez scalanie i implementuje je <input type="checkbox"/> ocenia i porównuje złożoność czasową i obliczeniową algorytmów 	<ul style="list-style-type: none"> <input type="checkbox"/> pisze programy o podwyższonym stopniu trudności wykorzystujące metodę „dziel i zwyciężaj” oraz algorytmy sortowania <input type="checkbox"/> posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym <input type="checkbox"/> do rozwiązania problemu dobiera optymalny algorytm i struktury danych, a także korzysta z funkcji bibliotecznych
P	Programowanie zespołowe	3	<ul style="list-style-type: none"> <input type="checkbox"/> wyjaśnia, czym jest dokumentacja projektu (projektowa, użytkownika, techniczna), bierze czynny udział w jej tworzeniu <input type="checkbox"/> aktywnie uczestniczy w realizacji projektu <input type="checkbox"/> przyjmuje różne role w zespole realizującym projekt <input type="checkbox"/> prezentuje efekty wspólnej pracy 	<ul style="list-style-type: none"> <input type="checkbox"/> przyjmuje rolę lidera odpowiedzialnego za zespół i projekt <input type="checkbox"/> przydziela zadania, nadzoruje pracę innych <input type="checkbox"/> opracowując złożone problemy, posługuje się aplikacjami w stopniu zaawansowanym