Przedmiotowy system oceniania i wymagania edukacyjne

Klasa 2 liceum ogólnokształcącego i technikum • zakres podstawowy i rozszerzony

Na informatyce ocenia się głównie zadania wykonywane w czasie pracy indywidualnej na lekcji (należy brać pod uwagę wykazanie się określonymi umiejętnościami, wkładem pracy i pomysłowością), uczestnictwo w pracy zespołowej i umiejętność współpracy, ogólną aktywność, a także systematyczność. Można promować podejmowanie zadań dodatkowych oraz udział i osiągnięcia w konkursach związanych z informatyką. Rzadziej przeprowadzamy sprawdziany, choć można to zrobić po zakończeniu rozdziałów poświęconych arkuszowi kalkulacyjnemu, algorytmice i programowaniu. Kartkówki warto wykorzystać do krótkich tematów, bardziej teoretycznych. Nie powinno też zabraknąć samooceny ucznia, zwłaszcza podczas realizacji projektów. Ewaluacji należy dokonywać na bieżąco. Ocena końcowa powinna uwzględniać osiągnięte efekty w porównaniu do zakładanych celów.

Wymagania ogólne na poszczególne oceny

Ocena celująca (6) – uczeń w wysokim stopniu opanował wiedzę i umiejętności opisane w podstawie programowej, samodzielnie i bezbłędnie wykonuje ćwiczenia z podręcznika; na lekcjach jest aktywny. Bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca

Ocena bardzo dobra (5) – uczeń samodzielnie i bezbłędnie wykonuje ćwiczenia z podręcznika; na lekcjach jest aktywny; posiada wiadomości i umiejętności opisane w planie wynikowym.

Ocena dobra (4) – uczeń samodzielnie wykonuje wszystkie ćwiczenia z podręcznika; na lekcjach jest aktywny; posiada wiadomości i umiejętności opisane w planie wynikowym.

Ocena dostateczna (3) – uczeń samodzielnie wykonuje łatwiejsze ćwiczenia z podręcznika, czasami z pomocą; stara się pracować systematycznie, robi postępy; posiada wiadomości i umiejętności opisane w planie wynikowym.

Ocena dopuszczająca (2) – uczeń wykonuje łatwe ćwiczenia z podręcznika z pomocą; posiada wiadomości i umiejętności opisane w planie wynikowym; ma problemy z systematycznością, niemniej jednak nie przekreśla to możliwości postępów w ciągu dalszej nauki.

Ocena niedostateczna (1) Niespełnienie wymagań na ocenę mierną. Nieopanowanie podstawowych wiadomości i umiejętności uniemożliwiające dalszą naukę; brak umiejętności scharakteryzowania pojęć algorytmu, warunku, iteracji i rekurencji; nieznajomość prostych algorytmów; nierozwiązywanie najprostszych zadań; brak czynnego udziału w lekcjach, wykonywania zadań, pisania programów i odrabiania prac domowych.

Nr tematu	Temat lekcji	Omawiane zagadnienia	Ocena	Zgodnie z wymaganiami uczeń
Zgodnie z prawem				
1	**Prawo w sieci**	Zasady współżycia społecznego, wolność słowa. Prawo autorskie i pojęcia z nim związane. Wykorzystywanie utworów zgodnie z prawem.	**2**	• **[P]** definiuje utwór w świetle ustawy o prawie autorskim i prawach pokrewnych
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wyjaśnia, jak zgłosić naruszenie praw autorskich
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** omawia zasady dotyczące dozwolonego użytku osobistego • **[P]** omawia zasady dotyczące prawa do cytatu
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wyjaśnia, czym jest wolne oprogramowanie, i podaje jego przykłady • **[P]** wyjaśnia zasady korzystania z licencji CC-BY-SA 3.0 • **[P]** wyjaśnia zasady korzystania z zasobów domeny publicznej
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** omawia zasady korzystania z poszczególnych licencji CC
Algorytmika i programowanie w Pythonie lub C++				
2	**Algorytm Euklidesa w praktyce (także rozszerzony**	Pętla warunkowa while. Stosowanie algorytmu Euklidesa do rozwiązywania zadań. Działania na	**2**	• **[P]** z pomocą omawia algorytm Euklidesa z odejmowaniem • **[R: 7]** omawia geometryczną interpretację algorytmu Euklidesa • **[R: 15]** opisuje rozszerzony algorytm Euklidesa • **[R: 7]*** opisuje geometryczną interpretację algorytmu Euklidesa

	algorytm Euklidesa) Wariant źródłowy 2|2A	ułamkach z wykorzystaniem NWD i NWW. Iteracja, rekurencja, struktury i działania na ułamkach. Rozszerzony algorytm Euklidesa.	**3**	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje pętlę while do rozwiązywania prostych problemów • **[R: 7]** wyjaśnia różnicę między parametrami formalnym i aktualnym, a także między zmiennymi lokalną i globalną • **[R: 7]** implementuje w języku C++ algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem • **[R: 15]** omawia rozszerzony algorytm Euklidesa • **[R: 7]*** pisze program realizujący algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem, stosując funkcję typu void • **[R: 7]*** stosuje zmienne lokalne i globalne, a także przekazywanie parametrów przez wartość
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** stosuje algorytm Euklidesa z odejmowaniem do obliczania NWD i NWW • **[P]** stosuje algorytm Euklidesa z dzieleniem do obliczania NWD i NWW • **[R: 7]** stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW • **[R: 7]*** wykorzystuje algorytm Euklidesa do działań na ułamkach • **[R: 15]** zapisuje w postaci programu rozszerzony algorytm Euklidesa, wyjaśnia jego działanie i zastosowanie • **[R: 7]*** stosuje strukturę do reprezentacji liczb wymiernych • **[R: 15]*** wyjaśnia, na czym polega rozszerzony algorytm Euklidesa, oraz implementuje go w języku C++
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje NWD i NWW do działań na ułamkach • **[R: 7]** stosuje różne sposoby przekazywania parametrów do funkcji, uzasadnia ich użycie • **[R: 7]** wykorzystuje algorytm Euklidesa do działań na ułamkach, stosując struktury lub pary (typ pair) • **[R: 15]** do implementacji rozszerzonego algorytmu Euklidesa stosuje zarówno iterację, jak i rekurencję • **[R: 7]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 7]*** stosuje funkcje i dobiera sposób przekazywania parametrów, jednocześnie go uzasadniając

			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** omawia algorytm dodawania, odejmowania, mnożenia i dzielenia dwóch ułamków zwykłych z wykorzystaniem algorytmów NWD i NWW • **[R: 7]** implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję • **[R: 7]*** pisze programy o podwyższonym stopniu trudności prezentujące zastosowanie algorytmu Euklidesa
3	**Badanie własności liczb całkowitych Wariant źródłowy 3|3A**	Sprawdzanie, czy liczba jest pierwsza czy złożona. Badanie szczególnych własności liczb całkowitych. Liczby bliźniacze, rozkład na czynniki, hipoteza Goldbacha. Sito Eratostenesa i jego optymalizacja.	**2**	• **[P]** omawia algorytm znajdowania liczb pierwszych metodą sita Eratostenesa • **[R: 5]** podaje definicje liczby pierwszej i liczby złożonej • **[R: 5]** implementuje w języku C++ algorytm zliczający dzielniki danej liczby • **[R: 13]** wypisuje liczby pierwsze z zadanego przedziału, stosując metodę sita Eratostenesa • **[R: 5]*** wymienia podstawowe własności liczb pierwszych
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** z pomocą wykorzystuje algorytm sprawdzania pierwszości liczby do rozwiązywania prostych zadań na temat liczb • **[R: 5]** implementuje w języku C++ algorytm naiwny sprawdzający, czy liczba jest pierwsza • **[R: 13]** omawia algorytm sita Eratostenesa • **[R: 5]*** sprawdza, czy dana liczba jest pierwsza, stosując algorytm naiwny • **[R: 13]*** opisuje algorytmy sprawdzające, czy liczba jest pierwsza
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje algorytm sprawdzania pierwszości liczby do rozwiązywania prostych zadań na temat liczb • **[P]** analizuje i testuje rozwiązania zadań • **[R: 5]** implementuje w języku C++ i optymalizuje algorytm sprawdzający, czy liczba jest pierwsza • **[R: 5]** pisze program rozkładający liczby na czynniki pierwsze • **[R: 13]** pisze program realizujący algorytm sita Eratostenesa • **[R: 5]*** rozkłada liczbę złożoną na czynniki pierwsze • **[R: 13]*** omawia i stosuje algorytm sita Eratostenesa do wyszukiwania liczb pierwszych w określonym przedziale liczbowym

			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje poznane algorytmy do rozwiązywania zadań na temat liczb • **[R: 5]** przy testowaniu liczby na pierwszość stosuje funkcję typu logicznego • **[R: 5]** wyszukuje liczby bliźniacze • **[R: 13]** optymalizuje program realizujący algorytm sita Eratostenesa i szacuje jego złożoność czasową • **[R: 5]*** wyznacza liczby bliźniacze • **[R: 5]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 5]*** stosuje optymalny algorytm sprawdzający, czy liczba jest pierwsza, wykorzystując funkcję logiczną; uzasadnia jego efektywność • **[R: 13]*** określa złożoność obliczeniową algorytmu • **[R: 13]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 13]*** optymalizuje algorytm, dążąc do minimalnej złożoności obliczeniowej
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje poznane algorytmy do rozwiązywania trudniejszych zadań na temat liczb • **[R: 5]** pisze program rozkładający liczbę złożoną na dwie liczby pierwsze (hipoteza Goldbacha) • **[R: 5]*** implementuje algorytmy dotyczące liczb pierwszych o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych • **[R: 5]*** pisze program rozkładający liczbę złożoną na sumę dwóch liczb pierwszych (hipoteza Goldbacha) • **[R: 13]*** pisze programy o podwyższonym stopniu trudności wykorzystujące sito Eratostenesa

4	**Sortowanie bąbelkowe i przez wstawianie (także wybieranie, quicksort i scalanie) Wariant źródłowy 4|4A**	Sortowanie danych. Sortowanie metodą bąbelkową. Sortowanie przez wstawianie. Sortowanie przez wybieranie, szybkie i przez scalanie. Stabilność, sortowanie in situ, efektywność oraz pliki i struktury.	**2**	• **[P]** wymienia zastosowania sortowania w praktyce • **[R: 18]** wyjaśnia, na czym polega metoda „dziel i zwyciężaj” • **[R: 12]** omawia metody sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie na przykładowych danych • **[R: 12]*** definiuje pojęcie sortowania, prawidłowo określając klucz i porządek sortowania • **[R: 12]*** podaje przykłady sortowania prostego w życiu codziennym • **[R: 18]*** omawia metodę „dziel i zwyciężaj” oraz rekurencję
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** omawia sortowanie metodą bąbelkową • **[P]** omawia sortowanie metodą przez wstawianie • **[R: 12]** przedstawia w postaci listy kroków algorytmy sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie, określa operacje dominujące • **[R: 12]*** definiuje pojęcia sortowania in situ i stabilnego • **[R: 12]*** definiuje operacje kluczowe (dominujące) w algorytmach sortowania
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** realizuje sortowanie metodą bąbelkową • **[R: 12]** pisze programy sortujące metodami prostymi z zastosowaniem funkcji typu void • **[R: 18]** stosuje algorytmy sortowania szybkiego i przez scalanie • **[R: 12]*** stosuje metody sortowania prostego do sortowania liczb w zbiorze – bąbelkowe i przez wybieranie • **[R: 12]*** pisze programy realizujące poznane algorytmy sortowania • **[R: 12]*** dobiera właściwe struktury danych • **[R: 12]*** definiuje własne funkcje do rozwiązywania problemów z wykorzystaniem algorytmów sortowania • **[R: 18]*** wyjaśnia, na czym polega algorytm sortowania szybkiego oraz przez scalanie i implementuje je

			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** realizuje sortowanie metodą przez wstawianie • **[R: 12]** szacuje złożoność obliczeniową programów sortujących, modyfikuje funkcje sortujące, zmieniając porządek sortowania • **[R: 12]*** szacuje złożoność obliczeniową stosowanych algorytmów • **[R: 12]*** ocenia wpływ pierwotnego ułożenia danych w zbiorze na liczbę wykonywanych operacji • **[R: 18]*** ocenia i porównuje złożoność czasową i obliczeniową algorytmów • **[R: 18]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 18]*** do rozwiązania problemu dobiera optymalny algorytm i struktury danych, a także korzysta z funkcji bibliotecznych
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** analizuje i testuje różne metody sortowania • **[P]** realizuje sortowanie uproszczoną metodą bąbelkową • **[R: 12]** pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary) • **[R: 18]** stosuje w programach algorytmy sortowania inne niż omawiane na lekcjach (np. heapsort) • **[R: 12]*** pisze programy sortujące o podwyższonym stopniu trudności: sortowanie danych w plikach tekstowych, sortowanie struktur • **[R: 18]*** pisze programy o podwyższonym stopniu trudności wykorzystujące metodę „dziel i zwyciężaj” oraz algorytmy sortowania
Arkusz kalkulacyjny				
5	**Wykresy funkcji w arkuszu**	Przygotowywanie danych do wykresów funkcji. Tworzenie wykresów funkcji. Kolejność wykonywania działań w arkuszu.	**2**	• **[P]** z pomocą tworzy wykres funkcji liniowej
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy wykres funkcji liniowej
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy wykres funkcji kwadratowej • **[P]** zmienia wartości za pomocą pokrętła lub suwaka

			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy złożone wykresy funkcji
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** przygotowuje trudniejsze wykresy
6	**Komputerowe wspomaganie pomiarów**	Pozyskiwanie danych pomiarowych z czujników. Przygotowywanie surowych danych do przetwarzania. Uzyskiwanie danych liczbowych z materiału wideo.	**2**	• **[P]** z pomocą pobiera surowe dane z czujników
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** przygotowuje dane do analizy
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykonuje eksperymenty w aplikacji Phyphox, eksportuje dane • **[P]** opracowuje pobrane dane, dobiera odpowiednie narzędzia
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** analizuje eksperymenty w programie Tracker, opracowuje wyniki • **[P]** wykorzystuje linie trendu w wykresach funkcji liniowej
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** samodzielnie wykonuje doświadczenia i eksperymenty • **[P]** analizuje wyniki dodatkowych doświadczeń i eksperymentów
7	**Symulacje komputerowe w arkuszu**	Budowanie modelu. Opracowywanie arkusza. Prezentacja wyników.	**2**	• **[P]** z pomocą planuje kolejne kroki symulacji w arkuszu
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** korzysta z funkcji zaokrąglania wyników
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** przeprowadza symulację • **[P]** samodzielnie korzysta z Pomocy arkusza
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wprowadza dynamiczne tytuły osi wykresów
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** samodzielnie planuje i realizuje symulacje
8	**Tabele przestawne**		**2**	• **[P]** porządkuje dane, aby móc utworzyć tabelę przestawną

		Otwarte dane publiczne. Stosowanie tabel przestawnych. Wykresy przestawne i przebiegu w czasie.	**3**	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy wykresy przebiegu w czasie
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy tabele przestawne
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** prawidłowo dobiera pola do wyświetlania w tabeli przestawnej • **[P]** dokonuje wizualizacji danych z wykorzystaniem wykresów przebiegu w czasie
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** samodzielnie wykorzystuje tabele przestawne do analizy różnych danych
9	**Gromadzenie i przetwarzanie danych**	Samodzielne zbieranie danych za pomocą ankiet. Analizowanie danych z wykorzystaniem tabel przestawnych. Generowanie raportów.	**2**	• **[P]** z pomocą tworzy ankietę w chmurze
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** pobiera i importuje do arkusza wyniki ankiety
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** stosuje zaawansowane kryteria filtrowania
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** korzysta z fragmentatorów • **[P]** tworzy raporty z wykorzystaniem tabeli przestawnych i wykresów przebiegu w czasie
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** samodzielnie planuje i realizuje badanie na wybrany temat – przeprowadza ankietę, porządkuje dane i tworzy raport
Projekt multimedialny przewodnik				
10	**Planowanie projektu**	Opracowanie koncepcji projektu. Podział prac i harmonogram. Pozyskiwanie informacji. Role zespołowe,	**2**	• **[P]** wspólnie z innymi uczniami planuje zadania do wykonania
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wyszukuje potrzebne informacje • **[R: P]*** przyjmuje różne role w zespole realizującym projekt

		przydział zadań i kierowanie zespołem.	**4**	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** rozplanowuje podział zadań
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** analizuje i ocenia wyszukane informacje
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** stosuje zaawansowane wyszukiwanie • **[P]** tworzy wykres harmonogramu prac nad projektem • **[R: P]** w projektach zespołowych przyjmuje rolę lidera. • **[R: P]*** przyjmuje rolę lidera odpowiedzialnego za zespół i projekt • **[R: P]*** przydziela zadania, nadzoruje pracę innych
11	**Wykres mapy**	Prezentacja danych statystycznych na mapie. Tworzenie wykresów map. Wykresy interaktywne i dynamiczne.	**2**	• **[P]** pobiera dane statystyczne z ogólnodostępnych portali
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** importuje dane do arkusza
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** dokonuje analizy danych
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy kartogramy • **[P]** przedstawia wykres mapy w sposób czytelny
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje mapy 3D do prezentacji danych
12	**Spersonalizowana e-mapa**	Systemy GIS i GPS. Pozyskiwanie danych GPS. Tworzenie spersonalizowanej mapy.	**2**	• **[P]** z pomocą tworzy spersonalizowaną mapę
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** odczytuje i zapisuje geotagi we właściwościach zdjęcia
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wymienia formaty plików przechowujących dane GPS
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** samodzielnie tworzy spersonalizowaną mapę

			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** przedstawia dane w różnych serwisach – Google Maps, OpenStreetMap, Google Earth lub Traseo.pl
13	**Edycja i montaż wywiadu**	Planowanie nagrania. Przeprowadzenie wywiadu. Edycja i montaż materiału audio.	**2**	• **[P]** z pomocą opracowuje scenariusz nagrania
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** planuje i przygotowuje wywiad
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** nagrywa wywiad
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** dokonuje korekty i montażu nagrania
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** ocenia nagranie, dokonuje poprawek
14	**Pokaz i sprawozdanie z projektu**	Wzorzec slajdów. Dodawanie grafiki i audio. Zespołowe prezentowanie. Prezentacja rezultatów projektu programistycznego.	**2**	• **[P]** wymienia podstawowe zasady tworzenia prezentacji
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** przygotowuje prezentację
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje na slajdach diagramy, listy graficzne, schematy organizacyjne • **[P]** przygotowuje prezentację na podstawie własnego wzorca • **[P]** zapisuje prezentację w odpowiednim formacie • **[P]** przygotowuje się do prezentacji projektu i prezentuje projekt na forum klasy
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** dodaje nagranie do prezentacji • **[P]** dokonuje samooceny i ocenia projekty innych zespołów • **[R: P]** aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy. • **[R: P]*** prezentuje efekty wspólnej pracy
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** prezentuje bezbłędnie przygotowane wystąpienie

Grafika wektorowa				
15	**Podstawy edycji grafiki wektorowej**	Cechy charakterystyczne grafiki wektorowej. Tworzenie i przekształcanie rysunków w programie Inkscape. Operacje na obiektach.	**2**	• **[P]** z pomocą wykonuje proste rysunki z wykorzystaniem operacji na obiektach
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** omawia pojęcie grafiki wektorowej, jej wady i zalety
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykonuje podstawowe operacje na obiektach
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** ustawia kontur i wypełnienie obiektu
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje różne obiekty do wykonania skomplikowanych rysunków
16	**Praca z krzywymi**	Krzywe Béziera. Modyfikowanie ścieżek, edycja węzłów. Rozmieszczanie kopii wybranego obiektu.	**2**	• **[P]** z pomocą rysuje krzywe z wykorzystaniem narzędzia Pióro
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wyjaśnia, czym są krzywe Béziera i kiedy się je stosuje
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** rozróżnia rodzaje węzłów, wygładza węzły • **[P]** zamienia obiekt w ścieżkę
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** rysuje proste wzory z wykorzystaniem krzywych Béziera • **[P]** wstawia deseń wzdłuż ścieżki • **[P]** nakłada na ścieżkę tryb Spiro
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** rysuje skomplikowane wzory z wykorzystaniem krzywych Béziera • **[P]** wykorzystuje tutoriale w sieci do przygotowania obrazków
17	**Przekształcanie obiektów**	Kopiowanie i klonowanie obiektów. Edytowanie obiektów o nieregularnych	**2**	• **[P]** z pomocą tworzy kopię obiektu
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** klonuje obiekty

		kształtach. Tworzenie układu klonów.	**4**	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy układy klonów
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy motywy wykorzystujące interpolację
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje mechanizmy klonowania do projektowania grafiki
18	**Projektowanie logo**	Opracowanie projektu graficznego. Edycja tekstu wzdłuż ścieżki. Umieszczanie liter w kształcie.	**2**	• **[P]** z pomocą wykorzystuje narzędzie Tekst, tworzy obiekt tekstowy
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wstawia tekst na ścieżkę
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** omawia budowę logo i charakteryzuje logotyp • **[P]** tworzy prosty logotyp
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** wykorzystuje deformację obwiedni • **[P]** projektuje logo tekstowo-graficzne, tworzy wizytówkę
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy różne wersje logo do użycia w różnych okolicznościach
19	**Projektowanie infografiki**	Funkcje infografiki. Elementy składowe infografiki. Narzędzia do tworzenia infografiki.	**2**	• **[P]** omawia funkcje infografiki
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** przedstawia historię rozwoju infografiki oraz jej najnowsze trendy
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy prostą infografikę
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy infografikę, stosując zasadę czterech kroków • **[P]** ocenia infografikę własną i innych uczniów
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[P]** tworzy własne zaawansowane projekty

Uzupełnienie zakresu rozszerzonego • Algorytmy na liczbach całkowitych i tekstach				
20	**Od problemu do programu**	Specyfikacja problemu. Reprezentacje algorytmów. Konstrukcje języka C++. Funkcje, testowanie i dobór struktur danych.	**2**	• **[R: 1]** definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja • **[R: 1]** wymienia sposoby reprezentacji algorytmów • **[R: 1]** korzysta ze środowiska programistycznego: pisze w nim kod, kompiluje i uruchamia program, odczytuje i zapisuje pliki • **[R: 1]** pisze programy o niewielkim stopniu trudności • **[R: 1]** korzysta z podstawowych funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych • **[R: 1]*** omawia różne sposoby przedstawiania algorytmów (opis słowny, lista kroków, pseudokod) • **[R: 1]*** stosuje w języku C++ podstawowe konstrukcje programistyczne (operacje wejścia i wyjścia, instrukcja warunkowa, operatory matematyczne i logiczne)
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 1]** przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu • **[R: 1]** pisze programy wykonujące działania na liczbach całkowitych • **[R: 1]** przy pisaniu programów stosuje własne funkcje różnych typów, w tym funkcję typu void • **[R: 1]*** planuje kolejne kroki rozwiązania problemu • **[R: 1]*** tworzy algorytmy działania na liczbach całkowitych • **[R: 1]*** tworzy w języku C++ programy wykonujące działania na liczbach całkowitych
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 1]** określa specyfikację algorytmu (dane, wynik) • **[R: 1]** pisze programy o różnym stopniu trudności, szacuje ich efektywność • **[R: 1]** przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, pseudokodu • **[R: 1]** dobiera typy danych do realizacji problemu • **[R: 1]*** definiuje pojęcie specyfikacja algorytmu, określa dane i wyniki • **[R: 1]*** programuje i testuje rozwiązanie problemu • **[R: 1]*** sprawdza działanie algorytmów dla różnych danych • **[R: 1]*** ocenia efektywność algorytmu

			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 1]** charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania • **[R: 1]** pisze programy o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku • **[R: 1]** optymalizuje rozwiązania • **[R: 1]** stosuje zaawansowane funkcje środowiska i języka programowania (np. z biblioteki STL) • **[R: 1]** dobiera struktury danych i metody do rodzaju problemu • **[R: 1]** wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach • **[R: 1]*** dobiera struktury i typy danych do rodzaju problemu • **[R: 1]*** wyszukuje optymalne rozwiązania problemów • **[R: 1]*** objaśnia dobrany do danego problemu algorytm, uzasadnia jego poprawność i wybór
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 1]** charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++ • **[R: 1]** pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych lub oznaczone trzema gwiazdkami w podręczniku
21	**Systemy liczbowe i reprezentacja danych w komputerze**	Systemy pozycyjne, bit i bajt. Typy liczbowe, znaki i napisy. Kod ASCII, operacje logiczne i kod U2.	**2**	• **[R: 2]** wymienia podstawowe typy danych, operacje arytmetyczne i logiczne • **[R: 2]** w pisanych programach korzysta ze strukturalnych typów danych: napisów, struktur, tablic • **[R: 2]** definiuje pojęcie systemów liczbowych • **[R: 2]*** wyjaśnia, czym jest tablica kodów ASCII • **[R: 2]** wymienia systemy liczbowe używane w informatyce • **[R: 2]*** definiuje pojęcie pozycyjnego systemu liczbowego • **[R: 2]*** wymienia systemy liczbowe stosowane w informatyce • **[R: 2]*** definiuje pojęcia bit i bajt

			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 2]** konwertuje liczby między pozycyjnymi systemami liczbowymi • **[R: 2]** przedstawia liczby w kodzie U2 • **[R: 2]** definiuje pojęcie zdania logicznego, charakteryzuje podstawowe operacje logiczne (koniunkcja, alternatywa, negacja) oraz operatory logiczne • **[R: 2]** charakteryzuje wybrane typy zmiennych służących do zapisu liczb całkowitych w języku C++: short int, int, long int, long long int • **[R: 2]*** omawia sposób reprezentowania liczb całkowitych w komputerze • **[R: 2]*** opisuje, jak w komputerze reprezentowane są znaki i napisy (char, string), odwołuje się do znaku w napisie za pomocą indeksu • **[R: 2]*** omawia działanie operacji logicznych i reprezentację ich wyników w komputerze (wynik może przyjmować wartość prawda – 1 lub fałsz – 0, co zajmuje 1 bajt pamięci) *Do weryfikacji: podana w źródle wielkość pamięci wyniku logicznego wymaga odniesienia do typu i środowiska programistycznego.*
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 2]** stosuje zmienne typu unsigned w tworzonych programach • **[R: 2]*** dokonuje konwersji między pozycyjnymi systemami liczbowymi, wykorzystując przy tym zależności między systemami binarnym i ósemkowym oraz binarnym i heksadecymalnym • **[R: 2]*** wymienia typy danych służące do zapisu liczb całkowitych (short int, int, long int, long long int, unsigned), stosuje je w pisanych programach
			5	*Spełnia kryteria ocen niższych w danym zakresie.*
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 2]*** wykonuje zadania o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku, z arkuszy maturalnych z lat poprzednich lub konkursów i olimpiad informatycznych
22	**Algorytmy zamiany reprezentacji liczb między systemami liczbowymi**	Konwersja reprezentacji liczb. Systemy binarny, dziesiętny, ósemkowy i szesnastkowy. Szybkie potęgowanie.	**2**	• **[R: 3]** konwertuje liczby między systemami binarnym i decymalnym
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 3]*** korzysta z dostępnych bibliotek w tworzonych przez siebie programach • **[R: 3]*** tworzy własne funkcje rozwiązujące problemy

			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 3]** pisze programy konwertujące liczby między systemem dziesiętnym i binarnym • **[R: 3]** w algorytmach zamiany wykorzystuje zależności między systemami binarnym, ósemkowym i heksadecymalnym • **[R: 3]*** tworzy programy do konwersji między liczbami w systemach binarnym i decymalnym • **[R: 3]*** posługuje się środowiskiem programistycznym, strukturami danych oraz językiem programowania w stopniu umożliwiającym implementację omawianych algorytmów
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 3]** pisze programy konwertujące liczby między różnymi systemami pozycyjnymi • **[R: 3]** wykorzystuje rozwinięcie binarne liczby dziesiętnej w algorytmie szybkiego podnoszenia do potęgi • **[R: 3]*** pisze programy zamieniające liczby z systemu decymalnego na system heksadecymalny • **[R: 3]*** pisze programy konwertujące liczbę dziesiętną na liczbę w podanym systemie pozycyjnym • **[R: 3]*** stosuje binarną reprezentację liczby w algorytmie szybkiego podnoszenia do potęgi • **[R: 3]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 3]*** do rozwiązania problemu dobiera optymalny algorytm i struktury danych • **[R: 3]*** uzasadnia poprawność zaproponowanego rozwiązania
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 3]*** 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
23	**Czy to jest palindrom? (także anagramy)**	Operacje na napisach. Palindromy i anagramy. Funkcje logiczne.	**2**	• **[R: 4]** wyjaśnia, czym są palindrom i anagram, podaje przykłady • **[R: 4]*** definiuje pojęcie palindromu • **[R: 4]*** określa, czy dany napis lub liczba są palindromami

		Przetwarzanie plików tekstowych.	**3**	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 4]** korzysta z biblioteki string do operacji na łańcuchach znaków • **[R: 4]** wykonuje operacje na napisach, wykorzystując słowa kluczowe: size, find, substr, erase, toupper, tolower • **[R: 4]** wczytuje napisy ze spacjami, wykorzystując słowo kluczowe getline • **[R: 4]** tworzy algorytmy sprawdzające, czy napis jest palindromem • **[R: 4]** przedstawia w postaci algorytmu problem wyszukiwania anagramów • **[R: 4]*** 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) • **[R: 4]*** definiuje własne funkcje w języku C++, wyjaśnia celowość ich stosowania, rozróżnia parametry formalne i aktualne • **[R: 4]*** opisuje popularne funkcje oraz metody stosowane dla zmiennych typu string (toupper, tolower, size, substr, erase)
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 4]** implementuje w języku C++ algorytmy sprawdzające, czy napis jest palindromem • **[R: 4]** pisze programy sprawdzające, czy dwa napisy są anagramami, wykorzystując funkcję sort z biblioteki STL • **[R: 4]*** realizuje w języku C++ algorytmy sprawdzające, czy dany napis jest palindromem, oraz wyszukujące palindromy w zdaniach
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 4]** pisze funkcje typu logicznego, np. sprawdzającą, czy napis jest palindromem • **[R: 4]** sprawdza, czy napisy są anagramami, stosując sortowanie lub zliczanie znaków • **[R: 4]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 4]*** optymalizuje algorytmy i ocenia ich efektywność
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 4]** wyszukuje palindromy lub anagramy w plikach tekstowych • **[R: 4]** tworzy palindromy z napisów, dopisując minimalną liczbę znaków • **[R: 4]*** pisze programy dotyczące palindromów o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych

24	**Działania na liczbach w systemach innych niż dziesiętny**	Arytmetyka w różnych systemach pozycyjnych. Duże liczby zapisane jako napisy. Kod U2.	**2**	• **[R: 6]** dodaje pisemnie liczby binarne
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 6]** dodaje liczby binarne • **[R: 6]** wykonuje działania arytmetyczne na liczbach w systemach liczbowych o różnych podstawach • **[R: 6]*** wykonuje działania arytmetyczne na liczbach w różnych systemach pozycyjnych • **[R: 6]*** wyjaśnia różnicę między operacjami na liczbach o podstawie od 1 do 9 i większej od 10 *Do weryfikacji: źródłowe sformułowanie „podstawie od 1 do 9 i większej od 10” pomija podstawę 10 i obejmuje 1. Zachowano tekst źródła.* • **[R: 6]*** stosuje odejmowanie w dzieleniu pisemnym liczb binarnych • **[R: 6]*** stosuje dodawanie liczby przeciwnej zapisanej w kodzie U2 przy odejmowaniu liczby binarnej
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 6]** implementuje w języku C++ algorytmy wykonujące działania arytmetyczne na liczbach w różnych systemach • **[R: 6]*** pisze programy wykonujące operacje arytmetyczne na liczbach w różnych systemach pozycyjnych
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 6]** w programach wykonujących działania na liczbach w różnych systemach pozycyjnych wykorzystuje bibliotekę string i strukturalne typy danych • **[R: 6]** wykonuje operacje arytmetyczne na liczbach w różnych systemach, implementuje je w języku C++ • **[R: 6]*** wykonuje działania o podwyższonym stopniu trudności • **[R: 6]*** wykonuje obliczenia na dowolnie dużych liczbach, wykorzystując napisy • **[R: 6]*** optymalizuje programy, szacuje ich efektywność
			6	*Spełnia kryteria ocen niższych w danym zakresie.*

25	**Szyfr Cezara i inne szyfry podstawieniowe**	Kryptologia i kryptografia. Szyfr Cezara, szyfr kolumnowy, permutacyjny i Vigenere'a. Klucze, znaki diakrytyczne i pliki.	**2**	• **[R: 8]** definiuje pojęcia: kryptologia, kryptografia, kryptoanaliza, tekst jawny, klucz, szyfrogram • **[R: 8]** rozróżnia szyfry podstawieniowe i przestawieniowe • **[R: 8]** omawia szyfr Cezara jako przykład szyfru podstawieniowego i szyfr kolumnowy jako przykład szyfru przestawieniowego • **[R: 8]** wczytuje dane z pliku tekstowego, zapisuje wyniki w pliku • **[R: 8]*** definiuje szyfry: podstawieniowy, monoalfabetyczny i permutacyjny, wymienia przykłady takich szyfrów • **[R: 8]*** stosuje w swoich programach operacje plikowe – wczytywanie danych z pliku dyskowego, zapis wyniku do pliku
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 8]** pisze program szyfrujący napis szyfrem Cezara • **[R: 8]*** omawia szyfr Vigenere'a • **[R: 8]*** korzysta z funkcji bibliotecznych
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 8]** szyfruje dane wczytane z pliku tekstowego • **[R: 8]*** wykorzystuje odpowiednio dobrane struktury danych
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 8]** szyfruje dane wczytane z pliku z uwzględnieniem polskich znaków diakrytycznych • **[R: 8]*** pisze program szyfrujący informację szyfrem Cezara z wykorzystaniem liter z polskimi znakami diakrytycznymi • **[R: 8]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 8]*** tworzy własne funkcje, dobierając sposób przekazywania parametrów
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 8]** pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere'a) i różnych kluczy • **[R: 8]*** pisze programy szyfrujące o podwyższonym stopniu trudności

Uzupełnienie zakresu rozszerzonego • Struktury danych i złożoność				
26	**Łamiemy szyfr Cezara**	Kryptoanaliza. Analiza częstości znaków. Zliczanie znaków i wyszukiwanie maksimum w tablicy.	**2**	• **[R: 9]** wyjaśnia, na czym polega łamanie szyfru • **[R: 9]** omawia algorytm zliczania znaków w tekście • **[R: 9]*** wyjaśnia, na czym polega łamanie szyfru (kryptoanaliza)
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 9]** omawia algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 9]** implementuje w języku C++ algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic • **[R: 9]*** stosuje algorytmy zliczające liczbę wystąpień znaków w tekście z zastosowaniem strukturalnego typu danych – tablic • **[R: 9]*** pisze program znajdujący maksimum w tablicy i wypisujący jego pozycję (algorytm „dziel i zwyciężaj”)
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 9]** pisze program odczytujący informację ukrytą za pomocą szyfru Cezara z wykorzystaniem analizy częstości znaków w tekście • **[R: 9]*** łamie szyfr Cezara, stosując analizę częstości
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 9]*** opisuje różne sposoby łamania szyfrów i implementuje je w języku C++ • **[R: 9]*** pisze programy deszyfrujące o podwyższonym poziomie trudności
27	**Poszukujemy liczby**	Wyszukiwanie liniowe, z wartownikiem i binarne. Minimum i maksimum. Wskaźniki i referencje.	**2**	• **[R: 10]** omawia algorytmy wyszukiwania liczby w zbiorach uporządkowanym i nieuporządkowanym • **[R: 10]** stosuje funkcję losującą w tworzonych programach
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 10]** implementuje w języku C++ algorytmy wyszukiwania liniowego i liniowego z wartownikiem, porównuje ich efektywność

			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 10]** stosuje różne sposoby przekazywania parametrów do funkcji: przez wartość, referencję lub wskaźnik • **[R: 10]** stosuje algorytm wyszukiwania binarnego i oszacowuje jego złożoność czasową • **[R: 10]*** znajduje wartość w zbiorach uporządkowanym i nieuporządkowanym, stosując odpowiednio algorytmy wyszukiwania liniowego, liniowego z wartownikiem i binarnego • **[R: 10]*** pisze programy wykorzystujące przekazywanie parametru do funkcji przez wskaźnik i referencję • **[R: 10]*** szacuje złożoność czasową zastosowanych algorytmów wyszukiwania • **[R: 10]*** wyjaśnia na przykładach różnice między różnymi sposobami przekazywania parametrów do funkcji
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 10]** pisze program wyszukujący jednocześnie minimum i maksimum w zbiorze z wykorzystaniem metody „dziel i zwyciężaj” oraz podaje wzór na liczbę wykonywanych operacji • **[R: 10]*** stosuje algorytm „dziel i zwyciężaj” do jednoczesnego znajdowania maksimum i minimum w zbiorze • **[R: 10]*** podaje wzór na liczbę wykonywanych operacji w algorytmie „dziel i zwyciężaj”
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 10]** implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej • **[R: 10]** stosuje zaawansowane algorytmy wyszukiwania, np. najlepszego wyboru (trwałych par), stosując rekurencję • **[R: 10]*** pisze programy o podwyższonym stopniu trudności
28	**Jak ocenić złożoność obliczeniową algorytmu?**	Operacja dominująca. Złożoność czasowa i pamięciowa. Przypadek optymistyczny, średni i pesymistyczny.	**2**	• **[R: 11]** omawia pojęcia: złożoność obliczeniowa algorytmu, algorytm naiwny, algorytm optymalny, złożoność pesymistyczna, złożoność oczekiwana (średnia) • **[R: 11]*** definiuje złożoność obliczeniową algorytmu • **[R: 11]*** wyjaśnia, czym jest złożoność oczekiwana (średnia), optymistyczna i pesymistyczna • **[R: 11]*** rozróżnia pojęcia algorytmu naiwnego i optymalnego
			3	*Spełnia kryteria ocen niższych w danym zakresie.*

			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 11]*** szacuje złożoność czasową i pamięciową
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 11]*** określa złożoność czasową i pamięciową algorytmów z zastosowaniem odpowiednich wzorów • **[R: 11]*** ocenia efektywność algorytmów
			6	*Spełnia kryteria ocen niższych w danym zakresie.*
29	**Szukamy różnych podciągów**	Spójne podciągi o zadanych własnościach. Najdłuższy podciąg, maksymalna suma i analiza efektywności.	**2**	• **[R: 14]** wyszukuje w ciągu liczb spójne podciągi (nierosnący, niemalejący, stały), wskazuje najdłuższe, oblicza ich sumę • **[R: 14]*** definiuje pojęcia podciągu oraz podciągu spójnego • **[R: 14]*** znajduje w zbiorze podciągi o różnych własnościach
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 14]** przedstawia algorytmy znajdowania spójnych podciągów, wyznaczania najdłuższego z nich oraz podciągu o największej sumie elementów
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 14]** implementuje w języku C++ algorytmy wyszukujące spójne podciągi o różnych cechach • **[R: 14]*** oblicza długość najdłuższego niemalejącego spójnego podciągu oraz liczbę jego elementów • **[R: 14]*** znajduje w zbiorze spójny podciąg o maksymalnej sumie i wypisuje jego elementy
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 14]** wyszukuje spójne podciągi w plikach tekstowych, stosując optymalne algorytmy (w tym programowanie dynamiczne), wyjaśnia ich działanie • **[R: 14]*** wymienia i stosuje różne algorytmy znajdowania maksymalnej sumy elementów spójnych podciągów, oceniając ich złożoność obliczeniową • **[R: 14]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 14]*** do rozwiązania problemu dobiera optymalny algorytm i struktury danych

			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 14]** stosuje zaawansowane algorytmy i struktury danych do wyszukiwania spójnych podciągów • **[R: 14]*** pisze programy o podwyższonym stopniu trudności wyszukujące spójne podciągi
Uzupełnienie zakresu rozszerzonego • Metody algorytmiczne				
30	**Iteracja a rekurencja**	Iteracja i rekurencja. Warunki brzegowe. Ciąg Fibonacciego, silnia i potęga. Złoty podział. Wieże Hanoi i hipoteza Collatza.	**2**	• **[R: 15]** definiuje pojęcia iteracji i rekurencji • **[R: 15]** omawia zasadę złotego podziału • **[R: 15]*** opisuje zasadę działania rekurencji • **[R: 15]*** opisuje zasadę złotego podziału
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 15]** implementuje w języku C++ algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi • **[R: 15]*** implementuje w języku C++ algorytmy rekurencyjne, określa warunki brzegowe • **[R: 15]*** oblicza n-ty wyraz ciągu Fibonacciego metodami iteracyjną i rekurencyjną
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 15]** porównuje algorytmy iteracyjne i rekurencyjne (liczbę wykonywanych operacji), szacuje ich złożoność czasową • **[R: 15]*** porównuje iteracyjne i rekurencyjne wersje algorytmów • **[R: 15]*** szacuje złożoność czasową stosowanych algorytmów • **[R: 15]*** oblicza liczbę wykonywanych operacji w algorytmach rekurencyjnych
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 15]** pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję *W źródle identyczne kryterium wież Hanoi występuje przy ocenach 5 i 6. Zachowano oba przypisania; wymagają rozstrzygnięcia przy ustalaniu zasad oceniania.* • **[R: 15]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 15]*** do rozwiązania problemu dobiera optymalny algorytm i struktury danych • **[R: 15]*** uzasadnia wybór iteracji lub rekurencji do rozwiązania problemu

			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 15]** pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję *W źródle identyczne kryterium wież Hanoi występuje przy ocenach 5 i 6. Zachowano oba przypisania; wymagają rozstrzygnięcia przy ustalaniu zasad oceniania.* • **[R: 15]*** pisze programy o podwyższonym stopniu trudności, np. sprawdzanie hipotezy Collatza
31	**Metoda zachłanna**	Problem kasjera, harmonogram wykorzystania sali i wyszukiwanie drogi. Algorytmy zachłanne i algorytmy z nawrotami.	**2**	• **[R: 16]** omawia metody zachłanne na przykładzie problemu kasjera, harmonogramu sali i wyszukiwania drogi • **[R: 16]*** wyjaśnia, na czym polega metoda zachłanna, i wymienia przykłady jej stosowania
			3	*Spełnia kryteria ocen niższych w danym zakresie.*
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 16]** stosuje metodę zachłanną w programach – problem kasjera, harmonogram wykorzystania sali, wyszukiwanie drogi • **[R: 16]*** implementuje następujące algorytmy zachłanne: problem kasjera (wydawania reszty minimalną liczbą nominałów), problem telewidza/kinomana (optymalny harmonogram wykorzystania sali), wyszukiwanie optymalnej drogi
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 16]*** ocenia przydatność zastosowanych algorytmów • **[R: 16]*** stosuje własne kryterium porównania w funkcji sort z biblioteki STL • **[R: 16]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 16]*** do rozwiązania problemu dobiera optymalny algorytm i struktury danych • **[R: 16]*** objaśnia algorytm wybrany do rozwiązania problemu oraz ocenia jego efektywność i niedoskonałość
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 16]*** pisze programy o podwyższonym stopniu trudności z wykorzystaniem algorytmów zachłannych, stosując rekurencję i algorytmy z nawrotami
32	**Programowanie dynamiczne**	Programowanie dynamiczne. Problem kasjera,	**2**	• **[R: 17]*** porównuje metody zachłanną i dynamiczną • **[R: 17]*** wyjaśnia, na czym polega metoda dynamiczna

		harmonogram i droga. Najdłuższy wspólny podciąg. Porównanie z metodą zachłanną.	**3**	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 17]** formułuje algorytm wydawania reszty minimalną liczbą monet, harmonogramu wykorzystania sali, znajdowania drogi metodami zachłanną i dynamiczną
			4	*Spełnia kryteria ocen niższych w danym zakresie.*
			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 17]** stosuje metody zachłanną i dynamiczną w problemach kasjera, harmonogramu wykorzystania sali i wyszukiwania drogi, wskazuje wady i zalety obu metod, szacuje złożoność czasową • **[R: 17]*** implementuje optymalne algorytmy dotyczące problemu kasjera, telewidza, znajdowania drogi • **[R: 17]*** stosuje metodę dynamiczną do znajdowania najdłuższego wspólnego podciągu • **[R: 17]*** posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym • **[R: 17]*** do rozwiązania problemu dobiera optymalny algorytm i struktury danych
			6	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: 17]*** pisze programy o podwyższonym stopniu trudności wykorzystujące algorytmy dynamiczne
Projekt programistyczny				
33	**Programowanie zespołowe – projekt zespołowy**	Realizacja zespołowego projektu programistycznego. Dokumentacja projektowa, użytkownika i techniczna.	**2**	• **[R: P]** uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania o niewielkim stopniu trudności.
			3	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: P]** uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania.
			4	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: P]** uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej. • **[R: P]*** wyjaśnia, czym jest dokumentacja projektu (projektowa, użytkownika, techniczna), bierze czynny udział w jej tworzeniu • **[R: P]*** aktywnie uczestniczy w realizacji projektu

			5	*Spełnia kryteria ocen niższych w danym zakresie.* • **[R: P]*** opracowując złożone problemy, posługuje się aplikacjami w stopniu zaawansowanym
			6	*Spełnia kryteria ocen niższych w danym zakresie.*