

Wymagania edukacyjne niezbędne do uzyskania poszczególnych  
 śródrocznych i rocznych ocen klasyfikacyjnych z informatyki

Szkoła ponadpodstawowa

KLASA I – POZIOM PODSTAWOWY

Bezpieczna praca z komputerem				
2	3	4	5	6
Zna zasady korzystania z pracowni komputerowej	Charakteryzuje rodzaje danych osobowych i dotyczące ich przepisy RODO	Omawia i stosuje dobre praktyki w zakresie ochrony oprogramowania	Tworzy bezpieczne hasła, potrafi sprawdzić moc hasła	Zna i stosuje różne sposoby zabezpieczania kont
Wyszukiwanie informacji w Internecie				
2	3	4	5	6
Wyszukuje adresy stron WWW zawierające proste hasło – korzysta z wyszukiwarki internetowej. Zna zasady nawigacji po stronie WWW, poruszając się po wybranych stronach internetowych	Wie, czym są strona WWW oraz zna genezę powstania Internetu. Wymienia wybrane usługi Internetowe. Podaje opisy i zastosowania wyszukiwarki internetowej, katalogu stron WWW i portalu. Szuka informacji w Internecie, konstruując złożone hasło.	Omawia rozwój usług internetowych, wskazując najważniejsze fakty. Wyjaśnia, na czym polega przeglądanie strony internetowej. Potrafi właściwie zawęzić obszar poszukiwań, aby szybko odszukać informacje. Korzysta z encyklopedii i słowników w wersji elektronicznej.	Omawia organizację informacji w WWW. Wyjaśnia postać adresu URL. Potrafi zastosować różne narzędzia do wyszukiwania informacji, usprawniając szukanie informacji. Właściwie porządkuje informacje o stronach WWW. Potrafi odpowiednio ocenić przydatność i wiarygodność informacji.	Potrafi formułować własne wnioski i spostrzeżenia dotyczące rozwoju Internetu, jego znaczenia dla różnych dziedzin gospodarki i dla własnego rozwoju. Wyszukuje, gromadzi i właściwie selekcjonuje informacje, tworząc złożone projekty z różnych dziedzin
Stosowanie przepisów prawa				
2	3	4	5	6
Zna podstawowe przepisy prawa dotyczące korzystania z cudzych materiałów i stosuje je w praktyce. Zna podstawowe zasady	Wie, co jest przedmiotem prawa autorskiego i co jemu nie podlega. Zna pojęcie licencji. Wymienia przykładowe rodzaje	Wyjaśnia wybrane przepisy prawa autorskiego, m.in.: „dozwolony użytek utworów”, zasady korzystania z	Potrafi uzasadnić zastosowanie wybranego przepisu prawa w konkretnym przypadku. Podaje przykłady łamania wybranych	Potrafi samodzielnie interpretować ważniejsze przepisy prawa autorskiego dotyczące korzystania z różnych źródeł informacji i

korzystania z programów komputerowych. Rozumie konieczność posiadania licencji na programy komputerowe. Jest świadomy istnienia przestępstw komputerowych.	darmowych licencji. Wymienia przykładowe rodzaje przestępstw komputerowych.	cudzego utworu bez pytania o zgodę, ochrona wizerunku. Omawia przykładowe rodzaje licencji na programy komputerowe. Omawia wybrane przykłady przestępstw komputerowych.	przepisów prawa. Omawia różnice pomiędzy różnymi rodzajami licencji. Sprawdza, na podstawie jakiej licencji jest rozpowszechniany dany program. Wyjaśnia zasady tej licencji.	ochrony programów komputerowych . Wyszukuje dodatkowe informacje na temat przestępstw komputerowych .
--	---	---	---	---

### Edytor tekstu

2	3	4	5	6
Stosuje podstawowe zasady edycji tekstów	Formatuje znaki, akapity i strony	Wykorzystuje tabulatory, sprawdza poprawność pisowni, obramowuje akapit i stronę	Tworzy złożone dokumenty tekstowe, w tym stosuje tabulatory, obramowania i inne narzędzia formatowania	Umie wykorzystać korespondencję seryjną do tworzenia dokumentów

### Tworzenie prezentacji multimedialnych

2	3	4	5	6
Zna podstawowe zasady tworzenia prezentacji multimedialnej. Tworzy prezentację składającą się z kilku slajdów z zastosowaniem animacji niestandardowych. Korzysta z szablonów slajdów. Umieszcza na slajdach tekst i obrazy. Zapisuje prezentację we wskazanym folderze docelowym. Potrafi uruchomić pokaz slajdów.	Przygotowuje prezentację na zadany temat na podstawie konspektu. Zmienia kolejność slajdów. Ustawia przejścia poszczególnych slajdów. Wie, do czego służą poszczególne widoki slajdów. Potrafi ustawić inne tło dla każdego slajdu. Wstawia do slajdu wykresy, tabele, równania matematyczne, efekty dźwiękowe	Potrafi właściwie zaplanować prezentację na zadany temat. Pracuje z widokami slajdów. Wstawia dźwięki z plików spoza listy standardowej. Zmienia tło, wstawia obiekty i hiperłącza. Umieszcza przyciski akcji. Dopasowuje przejścia między slajdami. Dodaje animacje i efekty	Wstawia podkład muzyczny odtwarzany podczas całej prezentacji. Przygotowuje materiały informacyjne dla uczestników pokazu i przeprowadza pokaz. Konwertuje przygotowaną prezentację do formatu umożliwiającego publikację w Internecie. Otwiera ją lokalnie w przeglądarce internetowej	Potrafi samodzielnie zaprojektować i przygotować multimedialną prezentację na wybrany temat, cechującą się ciekawym ujęciem zagadnienia, interesującym układem slajdów.

		dźwiękowe do obiektów. Prezentuje swoje prace przed klasą.		
Grafika komputerowa				
2	3	4	5	6
Z pomocą nauczyciela korzysta z wybranego programu do tworzenia grafiki rastrowej. Wyszukuje potrzebne funkcje w menu programu. Z pomocą nauczyciela korzysta z wybranego programu do tworzenia grafiki rastrowej. Wyszukuje potrzebne funkcje w menu programu.	Zna formaty plików graficznych. Opracowuje grafikę rastrową: stosuje warstwy i selekcje, zmianę kontrastu i nasycenia kolorów, kadrowanie i skalowanie. Wykonuje proste projekty w grafice wektorowej, korzystając z możliwości wstawiania Autokształtów (Kształtów) w edytorze tekstu.	Sprawne korzysta z Pomocy wbudowanej do programów w celu znalezienia szczegółowych sposobów rozwiązania danego problemu. Dostrzega różnice między grafiką rastrową i wektorową. Opracowuje grafikę rastrową: uzyskuje efekty specjalne dzięki zastosowaniu tzw. filtrów. Tworzy proste kompozycje, korzystając z wybranego programu do tworzenia grafiki wektorowej.	Rozumie znaczenie zapisu pliku graficznego w danym formacie –zależnie od przeznaczenia. Omawia zalety, wady i zastosowanie wybranych formatów plików grafiki rastrowej. Potrafi zastosować odpowiedni format pliku graficznego. Zapisuje pliki w różnych formatach. Opracowuje grafikę wektorową: przekształca obraz (pochyla, obraca), grupuje obiekty.	Samodzielnie zapoznaje się z możliwościami wybranego programu graficznego, przygotowując złożone projekty z różnych dziedzin

## KLASA II – POZIOM PODSTAWOWY

Obliczenia w arkuszu kalkulacyjnym				
2	3	4	5	6
<p>Zna podstawowe zastosowania arkusza kalkulacyjnego. Potrafi zaznaczyć zadany blok komórek. Ustawia liczbowy format danych. Samodzielnie pisze formułę wykonującą jedno z czterech podstawowych działań arytmetycznych (dodawanie, odejmowanie, mnożenie, dzielenie). Potrafi zastosować kopiowanie i wklejanie formuł. Tworzy prosty wykres. Zapisuje utworzony skroszyt we wskazanym folderze docelowym. Zna i stosuje podstawowe funkcje arkusza kalkulacyjnego: SUMA, ŚREDNIA.</p>	<p>Rozróżnia zasady adresowania w arkuszu kalkulacyjnym. Stosuje adresowanie bezwzględne wtedy, gdy jest to uzasadnione. Potrafi tworzyć formuły wykonujące bardziej zaawansowane obliczenia (potęgowanie, pierwiastkowanie, z zastosowaniem nawiasów). Tworzy wykres składający się z wielu serii danych, dodając do niego odpowiednie opisy. Ustawia inne formaty danych poza liczbowym. Formatuje tabelę. Korzysta z możliwości wstawiania funkcji. Potrafi zastosować funkcję JEŻELI.</p>	<p>Poprawnie planuje tabelę w arkuszu kalkulacyjnym, umieszczając w niej dane liczbowe i opisy. Stosuje adresowanie mieszane wtedy, gdy jest to uzasadnione. Stosuje formatowanie warunkowe tabeli arkusza kalkulacyjnego. Zna zastosowania różnych typów wykresów. Dostosowuje typ wykresu do danych, jakie ma przedstawiać. Potrafi narysować wykres wybranej funkcji matematycznej. Drukuje tabelę utworzoną w arkuszu kalkulacyjnym. Potrafi stosować wybrane funkcje arkusza kalkulacyjnego do rozwiązywania zadań z różnych przedmiotów</p>	<p>Potrafi układać rozbudowane formuły z zastosowaniem funkcji arkusza kalkulacyjnego. Potrafi stosować filtry i selekcjonować dane na podstawie zaawansowanych kryteriów. Korzysta z filtrów. Tworzy wykres funkcji trygonometrycznej. Dopasowuje wygląd arkusza kalkulacyjnego po wydruku, dobiera ustawienia strony, ustawia podział stron i obszar wydruku. Stosuje wybrane funkcje arkusza kalkulacyjnego: statystyczne, logiczne, matematyczne, tekstowe, daty i czasu. Stosuje zagnieżdżoną funkcję JEŻELI.</p>	<p>Potrafi przeprowadzić analizę przykładowego o problemu i opracować właściwy algorytm obliczeń. Potrafi rejestrować makra, stosować je w celu ułatwienia wykonywania często powtarzanych czynności. Zna działanie i zastosowanie większości funkcji dostępnych w arkuszu kalkulacyjnym.</p>

Grafika komputerowa 3D				
2	3	4	5	6
Tworzy bryły obrotowe i kompozycje obiektów na scenie	Tworzy wektorowe modele sfotografowanego przez siebie wybranego obiektu	Projektuje bardziej skomplikowane trójwymiarowe modele	Potrafi przeprowadzić tekstuowanie obiektów	Potrafi tworzyć animacje w grafice 3D
Interaktywne strony WWW				
2	3	4	5	6
Z pomocą nauczyciela tworzy szablony strony WWW	Wstawia elementy do dokumentu HTML	Projektuje wygląd typowych elementów strony	Dostosowuje wygląd strony do różnych urządzeń	Umieszcza skrypty języka JavaScript na stronie WWW

## KLASA III – POZIOM PODSTAWOWY

Algorytmika i programowanie				
2	3	4	5	6
<p>Wie, co to jest algorytm. Określa dane do zadania oraz wyniki. Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym. Potrafi narysować (odręcznie) schemat blokowy algorytmu liniowego. Potrafi napisać prosty program.</p>	<p>Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy. Zna pojęcie specyfikacji zadania. Zna wybrane sposoby prezentacji algorytmów. Przedstawia algorytm liniowy w postaci listy kroków. Podczas rysowania schematów blokowych potrafi wykorzystać Autokształty z edytora tekstu. Określa pojęcia program komputerowy, język programowania. Zapisuje prosty algorytm liniowy w wybranym języku programowania. Potrafi go skompilować i uruchomić.</p>	<p>Określa zależności między problemem, algorytmem a programem komputerowym. Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów, i podać przykłady. Przedstawia dokładną specyfikację dowolnego zadania. Analizuje poprawność budowy schematu blokowego. Wyjaśnia pojęcia: program wynikowy, kompilacja, translacja, interpretacja. Realizuje przykładowy algorytm liniowy w wybranym języku programowania. Wykonuje program i testuje go, podstawiając różne dane</p>	<p>Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji). Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych. Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego. Analizuje działanie algorytmu dla przykładowych danych. Potrafi posłużyć się kompilatorem danego języka. Potrafi wskazać i poprawić błędy w programie.</p>	<p>Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji). Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia. Potrafi samodzielnie zapoznać się z kompilatorem wybranego języka programowania. Samodzielnie pisze program realizujący algorytm liniowy.</p>
Bazy danych				
2	3	4	5	6
<p>Podaje obszary zastosowań baz danych –na przykładach z najbliższego otoczenia – szkoły,</p>	<p>Rozumie metody przetwarzania danych na przykładzie gotowej bazy danych.</p>	<p>Projektuje relacyjną bazę danych (na zadany temat) składającą się z trzech tabel połączonych</p>	<p>Potrafi wytłumaczyć pojęcie relacji. Projektuje relacyjną bazę danych składającą się z</p>	<p>Zna dokładnie wybrany program do projektowania baz danych. Potrafi samodzielnie</p>

instytucji naukowych, społecznych i gospodarczych. Podaje przykłady programów do tworzenia baz danych. Potrafi wykonać podstawowe operacje na bazie danych przygotowanej w jednej tabeli (wprowadzanie, redagowanie, sortowanie, wyszukiwanie, prezentacja). Potrafi utworzyć prostą kwerendę, jeden formularz i raport.	Określa podstawowe pojęcia (rekord, pole, typ pola, relacja, klucz podstawowy). Tworzy bazę danych składającą się z dwóch tabel, w każdej po kilka pól różnych typów. Projektuje formularze i raporty. Tworzy proste kwerendy wybierające. Potrafi wykonywać operacje przetwarzania danych w bazie składającej się z kilku rekordów. Zna zasady przygotowania korespondencji seryjnej.	relacją. Omawia typy relacji w bazie danych. Zna zasady definiowania kluczy podstawowych. Projektuje formularze i raporty według wskazówek nauczyciela. Potrafi utworzyć formularz z podformularzem. Umieszcza przyciski nawigacyjne. Tworzy kwerendy wybierające. Importuje dane z tabel arkusza kalkulacyjnego i do dokumentu tekstowego do tabel bazy danych. Eksportuje dane z tabel bazy danych do tabel arkusza kalkulacyjnego i do dokumentu tekstowego.	trzech lub większej liczby tabel. Samodzielnie ustala zawartość bazy (rodzaj informacji). Zna kilka rodzajów formularzy i raportów. Umie zaprojektować samodzielnie wygląd formularza i raportu. Na formularzach umieszcza pola kombi, ogranicza wartości, wstawia (gdy jest taka potrzeba) bieżącą datę, umieszcza przyciski poleceń. Stosuje funkcje standardowe w kwerendach i standardowe operatory w kryteriach wyszukiwania. Korzysta z parametrów w kwerendzie.	zaprojektować bazę danych, korzystając z wybranego narzędzia (programu). Projekt bazy opiera na rzeczywistych informacjach, aby można było wykorzystać ją w praktyce, np. w szkole czy w domu. Korzysta z dodatkowej, fachowej literatury.
--	--	---	--	--

### Sieci komputerowe

2	3	4	5	6
Testuje prędkość połączenia z siecią Internet na wybranym urządzeniu i interpretuje otrzymany wynik	Zna polecenia tekstowe służące do diagnostyki sieci i korzysta z nich	Oblicza liczbę możliwych do zaadresowania hostów na podstawie adresów IP i masek podsieci	Rozumie, czym jest model warstwowy TCP/IP	Wyjaśnia sposoby działania usługi NAT

## KLASA I – POZIOM ROZSZERZONY

Reprezentacja danych w komputerze				
2	3	4	5	6
Zna pojęcie systemu pozycyjnego. Wie, co to jest system binarny. Korzystając z przykładów, potrafi obliczyć wartość dziesiętną liczby zapisanej w systemie dwójkowym.	Wyjaśnia, co to jest system binarny, i potrafi dokonać zamiany liczby z systemu dziesiętnego na binarny i odwrotnie	Zna system szesnastkowy i potrafi wykonać konwersję liczb dziesiętnych na liczby w systemie szesnastkowym i odwrotnie. Zna zależność między systemem binarnym i szesnastkowym	Potrafi wykonać dowolną konwersję pomiędzy systemem dziesiętnym, dwójkowym i szesnastkowym. Potrafi napisać program obliczający wartość dziesiętną liczby dwójkowej.	Potrafi napisać program (w wersji iteracyjnej i rekurencyjnej) realizujący algorytm zamiany liczby dziesiętnej na postać binarną. Potrafi napisać program realizujący algorytm umożliwiający zamianę liczb z systemu szesnastkowego na dziesiętny i odwrotnie.
Podstawowe zasady programowania				
2	3	4	5	6
Zna klasyfikację języków programowania a. Zna ogólną budowę programu i najważniejsze elementy języka programowania – słowa kluczowe, instrukcje, wyrażenia, zasady składni. Potrafi zrealizować prosty algorytm liniowy i z warunkami w języku wysokiego poziomu; potrafi skompilować i uruchomić program.	Zapisuje program w czytelnej postaci – stosuje wcięcia, komentarze. Rozumie znaczenie i działanie podstawowych instrukcji (m.in. iteracyjnych, warunkowych) wybranego języka programowania a wysokiego poziomu. Wie, na czym polega programowanie strukturalne. Rozróżnia i poprawia błędy kompilacji i błędy wykonania. Potrafi	Wymienia i omawia modele programowania. Potrafi prezentować złożone algorytmy (z podprogramami) w wybranym języku programowania. Zna i stosuje instrukcje wyboru. Prezentuje wybrane algorytmy iteracyjne w postaci programu komputerowego. Zna rekurencyjne realizacje prostych algorytmów. Rozumie i stosuje zasady programowania strukturalnego. Deklaruje	Wie, jaka jest różnica między językiem wysokiego poziomu a językiem wewnętrznym; potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów. Deklaruje procedury i funkcje z parametrami. Wie, jakie znaczenie ma zasięg zmiennej. Definiuje funkcje rekurencyjne. Potrafi prezentować algorytmy rekurencyjne w postaci programu. Zapisuje w	Ocenia efektywność działania programu. Wskazuje podobieństwa i różnice dotyczące tworzenia programów zapisanych w różnych językach programowania; wyjaśnia działanie poszczególnych instrukcji, sposób deklaracji zmiennych. Sprawnie definiuje i stosuje procedury i funkcje w programach. Sprawnie korzysta z



	zrealizować algorytmy iteracyjne w języku wysokiego poziomu.	procedury i funkcje bez parametrów. Wie, na czym polega różnica pomiędzy przekazywaniem parametrów przez zmienną i przez wartość w procedurach i funkcjach. Rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą.	postaci programu wybrane algorytmy sortowania, algorytmy na tekstach, definiując odpowiednie procedury lub funkcje.	dodatkowej, fachowej literatury. Rozwiązuje przykładowe zadania z matury i olimpiady informatycznej.
--	--	--	---	--

### Multimedia i grafika komputerowa

2	3	4	5	6
Zna urządzenia multimedialne, wymienia przykładowe nazwy, określa ogólnie przeznaczenie urządzeń multimedialnych. Posługuje się drukarką i skanerem. Po zeskanowaniu zapisuje obraz w pliku w formacie domyślnym. Wymienia programy do tworzenia i obróbki grafiki. Posługuje się jednym z nich w celu tworzenia własnych rysunków. Tworzy i edytuje obrazy w wybranym programie graficznym,	Korzysta z różnych urządzeń multimedialnych, zna ich działanie, podaje ich przeznaczenie. Definiuje pojęcie komputer multimedialny. Zna sposoby reprezentacji obrazu i dźwięku w komputerze. Zna modele barw. Zapisuje plik graficzny w różnych formatach. Zna zastosowanie poszczególnych formatów, ich zalety i wady. Wymienia formaty zapisu dźwięku. Zna podstawowe możliwości wybranych programów do	Posługuje się sprawnie wybranymi urządzeniami multimedialnymi. Zna różne możliwości komputera w zakresie edycji obrazu, dźwięku, animacji i wideo. Omawia model barw. Zna pojęcia: RGB i CMYK. Omawia formaty plików dźwiękowych. Potrafi stosować różne narzędzia malarskie i korekcyjne oraz wybrać odpowiedni tryb ich pracy. Wie, na czym polega praca z warstwami. Wykonuje rysunki, korzystając z warstw. Potrafi tworzyć przykładowe fotomontaże.	Zna i stosuje w praktyce zaawansowaną obróbkę grafiki rastrowej i wektorowej. Przy użyciu odpowiednich narzędzi potrafi zaznaczyć fragmenty obrazu nawet o skomplikowanym kształcie. Potrafi zapisywać pliki multimedialne w różnych formatach, ze szczególnym uwzględnieniem formatów internetowych. Zna pojęcia: filtr, histogram, krzywa barw. Potrafi zdefiniować barwy i wykonać na nich operacje. Stosuje filtry. Wykonuje	Samodzielnie zapoznaje się z programami do obróbki grafiki rastrowej i wektorowej. Tworzy obrazy, wykorzystując różne możliwości programów. Korzystając z Pomocy i innych źródeł, poznaje możliwości programów graficznych. Przygotowuje grafikę do własnej strony internetowej lub prezentacji multimedialnej. Uczestniczy w konkursach dotyczących gra

<p>korzystając z podstawowych narzędzi do edycji obrazu. Wie, czym różni się grafika rastrowa od wektorowej.</p>	<p>edycji obrazu rastrowego i wektorowego. Potrafi wybrać fragmenty obrazu i wykonać na nich różne operacje, np. selekcje, przekształcenia (obroty, odbicia). Potrafi, korzystając z gotowego pliku video, wykonać proste operacje, tj. podzielić film na fragmenty, przyciąć film, dodać efekty.</p>	<p>Tworzy obraz w grafice wektorowej, rysuje figury, ścieżki. Wykonuje przekształcenia obrazu (obroty, odbicia), tworząc obrazy</p>	<p>ćwiczenia z maskami (wybiera fragmenty obrazu). Potrafi retuszować obraz. Wie, czym są krzywe Béziera i rysuje je. Opracowuje samodzielnie krótki film</p>	
--	---	---	---	--

### Algorytmy i zależności funkcyjne w arkuszu kalkulacyjnym

2	3	4	5	6
<p>Zna zasady tworzenia formuł i stosowania funkcji arkusza kalkulacyjnego . Zna i stosuje zasady adresowania względnego i bezwzględnego w arkuszu kalkulacyjnym. Zapisuje w arkuszu kalkulacyjnym algorytm liniowy i z warunkami. Stosuje wybrane funkcje arkusza kalkulacyjnego . Potrafi utworzyć wykres w</p>	<p>Zapisuje w arkuszu kalkulacyjnym algorytm z warunkami zagnieżdżonymi. Zna i stosuje zasady adresowania mieszanego w arkuszu kalkulacyjnym. Potrafi zrealizować iterację w arkuszu kalkulacyjnym. Rysuje wykres</p>	<p>Potrafi zrealizować pętlę zagnieżdżoną w arkuszu kalkulacyjnym. Wie, co to jest fraktal i w jaki sposób się go tworzy. Podaje przykłady fraktali. Rysuje wykres wybranej funkcji trygonometrycznej. Dobiera odpowiedni typ wykresu do prezentowanych danych</p>	<p>Rysuje wykres funkcji liniowej, wielomianu, wybranej funkcji trygonometrycznej i funkcji logarytmicznej. Zna możliwości zastosowania algorytmów iteracyjnych w rysowaniu fraktali, m.in.: śnieżynki Kocha, dywanu i trójkąta Sierpińskiego. Stosuje wybrane możliwości arkusza kalkulacyjnego do rozwiązywania zadań z różnych dziedzin.</p>	<p>Potrafi samodzielnie zobrazować wybraną zależność funkcyjną w arkuszu kalkulacyjnym, np. algorytm rozwiązywania układu równań liniowych metodą wyznaczników. Rozumie, w jaki sposób narysować paprotkę Barnsleya w arkuszu kalkulacyjnym. Korzystając z Internetu i innych źródeł, wyszukuje dodatkowe informacje na temat geometrii</p>

arkusza kalkulacyjnym				fraktalnej, m.in. dotyczące jej zastosowań.
-----------------------	--	--	--	---

## KLASA II – POZIOM ROZSZERZONY

Prezentacja algorytmu liniowego w wybranej notacji				
2	3	4	5	6
<p>Wie, co to jest algorytm. Określa dane do zadania oraz wyniki. Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym. Potrafi narysować (odręcznie) schemat blokowy algorytmu liniowego. Potrafi napisać prosty program, wyświetlaj</p>	<p>Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy. Zna pojęcie specyfikacji zadania. Zna wybrane sposoby prezentacji algorytmów. Przedstawia algorytm liniowy w postaci listy kroków. Podczas rysowania schematów blokowych potrafi wykorzystać Autokształty z edytora tekstu. Określa pojęcia program komputerowy, język programowania. Zapisuje prosty algorytm liniowy w wybranym języku programowania. Potrafi go skompilować i uruchomić.</p>	<p>Określa zależności między problemem, algorytmem a programem komputerowym. Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów, i podać przykłady. Przedstawia dokładną specyfikację dowolnego zadania. Analizuje poprawność budowy schematu blokowego. Wyjaśnia pojęcia: program wynikowy, kompilacja, translacja, interpretacja. Realizuje przykładowy algorytm liniowy w wybranym języku programowania. Wykonuje program i testuje go, podstawiając różne dane.</p>	<p>Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji). Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych. Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego. Analizuje działanie algorytmu dla przykładowych danych. Potrafi posłużyć się kompilatorem danego języka. Potrafi wskazać i poprawić błędy w programie.</p>	<p>Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji). Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia. Potrafi samodzielnie zapoznać się z kompilatorem wybranego języka programowania. Samodzielnie pisze program realizujący algorytm liniowy.</p>
Podstawowe zasady programowania				
2	3	4	5	6
Wymienia przykłady	Wie, czym jest zmienna w	Potrafi zastosować łańcuchowy i	Rozumie, na czym polega	Dobiera najlepszy

prostych struktur danych. Potrafi zadeklarować zmienne typu liczbowego (całkowite, rzeczywiste) i stosować je w zadaniach.	programie i co oznacza przypisanie jej konkretnej wartości. Rozróżnia struktury danych: proste i złożone. Podaje przykłady.	tablicowy typ danych w zadaniach. Deklaruje typ tablicowy i łańcuchowy.	dobór struktur danych do algorytmu. Wczytuje i wyprowadza elementy tablicy. Wprowadza dane tekstowe. Tworzy programy, dobierając odpowiednie	algorytm i odpowiednie struktury danych do rozwiązania postawionego problemu.
--	---	---	--	---

### Techniki algorytmiczne i wybrane algorytmy

2	3	4	5	6
Określa sytuacje warunkowe. Podaje przykłady zadań, w których występują sytuacje warunkowe. Wie, na czym polega powtarzanie tych samych operacji. Potrafi omówić na konkretnym przykładzie algorytm znajdowania najmniejszego o z trzech elementów. Wymienia własności algorytmów. Potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy kroków lub w postaci schematu blokowego	Potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzieniami) . Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady. Wie, od czego zależy liczba powtórzeń. Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Testuje rozwiązanie dla wybranych danych. Określa problemy, w których występuje rekurencja i podaje przykłady „zjawisk rekurencyjnych” – wziętych z życia i zadań szkolnych. Potrafi omówić algorytm	Analizuje algorytmy, w których występują powtórzenia (iteracje). Ocenia zgodność algorytmu ze specyfikacją. Zna sposoby zakończenia iteracji. Określa kroki iteracji. Potrafi zapisać w wybranej notacji np. algorytm sumowania n liczb, algorytm obliczania silni, znajdowania minimum w ciągu n liczb, algorytm rozwiązywania równania liniowego. Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci. Zna iteracyjną postać algorytmu Euklidesa. Zna rekurencyjną realizację wybranego algorytmu, np. silni. Zna przynajmniej dwie techniki sortowania (np. bąbelkowe, przez wybór) i zapisuje wybrany algorytm w postaci programu	Zapisuje algorytmy z pętlą zagnieżdżoną. Zna metodę „dziel i zwyciężaj” , algorytm generowania liczb Fibonacciego, schemat Hornera. Omawia ich iteracyjną realizację i potrafi przedstawić jeden z nich w wybranej notacji. Zna inne algorytmy sortowania, np. pozycyjne, przez wstawianie. Wskazuje różnicę między rekurencją a iteracją. Zna rekurencyjną realizację wybranych algorytmów, np. obliczania silni i algorytm	Rozumie dokładnie technikę rekurencji (znaczenie stosu). Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję. Zna trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, szukanie wzorca w tekście. Potrafi zapisać je w różnych notacjach (również w języku programowania a wysokiego poziomu). Korzysta samodzielnie z dodatkowej literatury fachowej. Określa złożoność

<p>dla przykładowych danych i ocenić w ten sposób jego poprawność. Zna i potrafi omówić wybrane algorytmy sortowania. Analizuje gotowe listy kroków wybranych algorytmów sortowania, wykonuje algorytmy dla wybranych danych. Potrafi omówić algorytm rozkładu liczby na czynniki pierwsze. Zna łamigłówkę Wież Hanoi. Wykonuje praktyczne ćwiczenie, odpowiednio przekłada trzy krążki.</p>	<p>porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy. Omawia wybrane algorytmy sortowania. Omawia wybrany algorytm na tekstach (np. tworzenie anagramów). Zna i omawia własności algorytmów. Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją. Określa liczbę prostych działań zawartych w algorytmie. Zna jeden z wybranych algorytmów sortowania: np. przez wstawianie. Sprawdza liczbę porównań elementów w tym algorytmie. Zna algorytmy zamiany liczb między dowolnymi systemami pozycyjnymi: obliczania wartości dziesiętnej liczby, zapisywania liczby dziesiętnej w</p>	<p>komputerowego. Omawia wybrane algorytmy na tekstach. Potrafi wyjaśnić, na czym polega wydawanie reszty metodą zachłanną i napisać listę kroków tego algorytmu. Rozumie, co to jest złożoność czasowa algorytmu i potrafi określić liczbę operacji wykonywanych na elementach zbioru w wybranym algorytmie sortowania. Rozróżnia złożoność czasową i pamięciową. Zna algorytmy sortowania: np. przez wstawianie, przez scalanie, metodą szybką. Sprawdza liczbę porównań elementów w przypadku każdego z algorytmów. Zna algorytm przeszukiwania binarnego. Potrafi utworzyć listę kroków tego algorytmu. Zna przykładowe algorytmy na liczbach naturalnych: generowanie liczb pierwszych (podejście naiwne, sito Eratostenesa). Omawia wybrany algorytm numeryczny, np. obliczanie wartości pierwiastka kwadratowego. Zna sposób zapisu liczby całkowitej i rzeczywistej</p>	<p>Euklidesa. Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną. Zapisuje wybrany algorytm na tekstach (np. tworzenie anagramów, zliczanie znaków w tekście, sprawdzanie, czy dany ciąg jest palindromem) w postaci programu komputerowego. Zapisuje algorytm wydawania reszty metodą zachłanną w postaci programu komputerowego. Wie, jak ocenić złożoność pamięciową algorytmu. Potrafi porównać złożoność obliczeniową różnych algorytmów tego samego zadania dla tych samych danych. Wie, kiedy algorytm jest efektywny. Omawia i stosuje w zadaniach algorytmy na liczbach</p>	<p>czasową i pamięciową wybranych algorytmów. Zna odpowiednie wzory. Określa efektywność algorytmów. Zapisuje wybrane algorytmy sortowania (np. przez wstawianie, przez scalanie, metodą szybką) w postaci programu komputerowego. Potrafi zapisać w języku programowania wysokiego poziomu algorytm konwersji liczb z dowolnego systemu pozycyjnego na inny. Tworzy programy komputerowe, stosując wybrane algorytmy. Rozwiązuje zadania z matury i olimpiady informatycznej.</p>
--	---	---	---	--

	<p>systemie liczbowym o określonej podstawie. Wie, jak sprawdzić, czy dana liczba jest liczbą doskonałą. Zna reprezentację danych numerycznych w komputerze: reprezentację binarną liczb ujemnych, reprezentację stałopozycyjną liczb, reprezentację zmiennopozycyjną liczb.</p>	<p>(zmiennoprzecinkowej).</p>	<p>naturalnych: generowanie liczb pierwszych (podejście naiwne, sito Eratostenesa). Omawia algorytm szybkiego podnoszenia do potęgi i algorytmy badające własności geometryczne (np. przynależność punktu do odcinka, badanie położenia punktu względem prostej). Omawia wybrany algorytm numeryczny, np. wyznaczanie miejsca zerowego funkcji, obliczanie wartości pierwiastka kwadratowego, obliczanie pola obszaru ograniczonego. Zna właściwości arytmetyki komputerowej. Na konkretnych przykładach potrafi sprawdzić, jak zmienia się wartość błędu względnego. Tworzy program</p>	
--	--	-------------------------------	--	--

			komputerowy, stosując wybrany algorytm	
Kompresja i szyfrowanie danych				
2	3	4	5	6
Wie, co to jest kompresja danych. Zna przynajmniej jeden algorytm kompresji danych. Wie, czym jest szyfrowanie danych. Zna przynajmniej jeden algorytm szyfrowania danych.	Rozumie, na czym polega kompresja danych i w jakim celu się ją wykonuje. Wymienia rodzaje kompresji. Omawia jeden przykładowy algorytm kompresji. Koduje tekst, używając alfabetu Morse'a. Wymienia przykładowe algorytmy szyfrowania. Potrafi zaszyfrować i odszyfrować prosty tekst.	Wie, co to jest współczynnik kompresji. Omawia rodzaje kompresji: kompresję stratną i bezstratną. Podaje przykłady algorytmów kompresji stratnej i bezstratnej. Omawia algorytm statyczny i słownikowy. Omawia przynajmniej dwa algorytmy szyfrowania: szyfr podstawieniowy i przestawieniowy.	Potrafi policzyć współczynnik kompresji. Wyjaśnia różnicę pomiędzy algorytmem statycznym a słownikowym. Stosuje algorytm słownikowy do kompresji ciągu znaków. Omawia przykładowy szyfr z kluczem. Stosuje szyfr Vigenère'a do zaszyfrowania ciągu znaków. Omawia wykorzystanie algorytmów szyfrowania w podpisie elektronicznym.	Wyszukuje dodatkowe informacje na temat kompresji i szyfrowania danych. Omawia inne algorytmy kompresji i szyfrowania. Potrafi zapisać wybrany algorytm kompresji lub szyfrowania w postaci programu. Zapoznaje się samodzielnie z kodem Huffmana i pokazuje na przykładzie jego zastosowanie. Omawia, czym się zajmuje stenografia, samodzielnie wyszukując informacje na ten temat. Wyjaśnia, w jaki sposób tworzy się podpis elektroniczny.

### KLASA III – POZIOM ROZSZERZONY

Algorytmika i programowanie –problemy zaawansowane				
2	3	4	5	6
Zna i stosuje podstawowe sposoby	Potrafi poprawnie zadeklarować dane potrzebne	Tworzy własne typy danych (typ okrojony, typ	Poprawnie formatuje wyprowadzane	Dobiera sposób wprowadzania danych do

<p>wprowadzania danych i wyprowadzania wyników w wybranym języku programowania. Zna i omawia strukturę typów danych w wybranym języku programowania. Wie, jakiego typu dane zapisuje się w rekordzie. Wie, czym charakteryzują się dane typu tablicowego. Wymienia modele programowania. Omawia model programowania strukturalnego. Wie, jakie są korzyści z definiowania procedur i funkcji. Potrafi zdefiniować procedury i funkcje w wybranym języku programowania. Wie, co to jest moduł.</p>	<p>do rozwiązania zadania. Wie, co określa typ danych i format danych. Potrafi odpowiednio sformatować wyprowadzane wyniki. Tworzy prosty program, w którym deklaruje dane typu rekordowego (strukturalnego). Wczytuje dane do rekordu. Potrafi omówić ogólnie zasady przetwarzania plików w wybranym języku programowania. Omawia szczegółowo modele programowania. Zna zasady programowania modularnego. Analizując przykładowe programy, tworzy własne moduły.</p>	<p>wyliczeniowy) i stałe. Zna procedury i funkcje umożliwiające generowanie danych losowych. Stosuje w programach generowanie losowych danych. Potrafi zastosować rekordowy typ danych do przetwarzania danych różnego rodzaju. Zna zasady przetwarzania plików w wybranym języku programowania (w językach programowania). Stosuje odpowiednie procedury i funkcje. Deklaruje zmienne typu plikowego. Korzystając z przykładów, odczytuje dane z pliku i zapisuje dane do pliku. Wie, na czym polega programowanie obiektowe. Zna podstawowe pojęcia programowania obiektowego: klasa, obiekt, pola, metody.</p>	<p>wyniki. Omawia stos jako przykład struktury danych. Wykonuje operacje na stosie. Implementuje stos z wykorzystaniem tablicy. Zna i omawia zastosowanie stosu (odwrotną notację polską – ONP). Wykonuje operacje na plikach w wybranym języku programowania (w językach programowania). Przetwarza pliki tekstowe. Tworzy własne programy, w których wykorzystuje przetwarzanie plików. Zna dynamiczne struktury danych i typ wskaźnikowy danych. Deklaruje zmienne typu wskaźnikowego. Stosuje w programach zmienne wskaźnikowe. Tworzy zmienne dynamiczne. Zna wybrane struktury dynamiczne. Analizuje gotowe programy, w których</p>	<p>rozwiązywanego zadania. Potrafi sprawdzić poprawność danych wprowadzanych do programu. Definiuje własne typy danych, potrzebne do rozwiązania danego zadania. Korzystając z możliwości generowania danych losowych, tworzy własne gry komputerowe. Stosuje w programach wybrane struktury dynamiczne. Tworzy listę jednokierunkową. Tworzy i przegląda binarne drzewo poszukiwań. Pisze program wyszukujący wartości w binarnym drzewie uporządkowanym. Korzystając z dodatkowej literatury, zapoznaje się z innymi strukturami dynamicznymi, np. z listą dwukierunkową. Tworzy programy z zastosowaniem struktur dynamicznych. Rozwiązuje zadania z matury i olimpiady informatycznej i bierze w niej udział.</p>
---	---	---	---	---



			<p>zastosowano listę jednokierunkową i binarne drzewo poszukiwań; uruchamia je i testuje dla wybranych danych.</p> <p>Rozumie i potrafi zastosować typ obiektowy.</p> <p>Definiuje klasy.</p> <p>Deklaruje pola prywatne.</p> <p>Omawia cechy programowania obiektowego: dziedziczenie i polimorfizm.</p> <p>Wie, czym są metody wirtualne.</p> <p>Poznaje zastosowanie tych cech, analizując gotowe programy.</p> <p>Modyfikuje programy według wskazówek nauczyciela.</p>	
--	--	--	---	--

### Komputer i system operacyjny

2	3	4	5	6
<p>Wymienia części składowe zestawu komputerowego, podaje ich parametry i przeznaczenie.</p> <p>Rozróżnia rodzaje pamięci komputera, określa ich własności i przeznaczenie.</p> <p>Wie, co to jest bit i bajt. Wie, co</p>	<p>Potrafi sklasyfikować środki (urządzenia) i narzędzia (oprogramowanie) technologii informacyjnej.</p> <p>Wie, jak działa komputer.</p> <p>Wyjaśnia rolę procesora.</p> <p>Rozumie sposób organizacji pamięci</p>	<p>Analizuje model komputera zgodny z ideą von Neumanna.</p> <p>Potrafi wymienić i omówić rodzaje aktualnie używanych komputerów.</p> <p>Omawia, jak działa procesor.</p> <p>Wymienia i omawia popularne systemy</p>	<p>Omawia szczegółowo model komputera zgodny z ideą von Neumanna.</p> <p>Wyjaśnia, w jaki sposób procesor wykonuje dodawanie liczb.</p> <p>Porównuje cechy różnych systemów operacyjnych, np. Microsoft</p>	<p>Omawia szczegółowo system Linux, porównując go do systemu Microsoft Windows.</p> <p>Korzystając z dodatkowych źródeł, omawia kierunek rozwoju systemów operacyjnych.</p> <p>Korzystając z dodatkowych</p>

to jest system operacyjny, wymienia i omawia jego podstawowe funkcje oraz z nich korzysta. Podaje przykłady systemów operacyjnych.	komputerowej. Zna jednostki pamięci, pojemności nośników i programów. Omawia dwa przykładowe systemy operacyjne.	operacyjne: Microsoft Windows, Unix, Linux, Mac Os.	Windows, Unix, Linux, Mac Os. Omawia przykładowe systemy operacyjne dla urządzeń mobilnych. Samodzielnie zapoznaje się z możliwościami nowych urządzeń związanych z TIK.	źródeł, omawia najnowsze osiągnięcia dotyczące systemów operacyjnych stosowanych w urządzeniach mobilnych.
--	--	---	--	--

## KLASA IV – POZIOM ROZSZERZONY

Sieci komputerowe				
2	3	4	5	6
Wyjaśnia pojęcia: sieć komputerowa, zasoby sieciowe, klient, serwer. Podaje podział sieci ze względu na wielkość, ze względu na model funkcjonowania i na topologię. Potrafi wymienić kilka cech pracy w sieci, odróżniając ją od pracy na autonomicznym komputerze.	Wymienia korzyści płynące z korzystania z sieci. Wyjaśnia, na czym polega wymiana informacji w sieci. Omawia podstawowe klasy i topologie sieciowe. Potrafi wymienić urządzenia i elementy sieciowe oraz omówić ich ogólne przeznaczenie. Zna cechy systemu działającego w szkolnej pracowni. Wie, co to jest protokół komunikacyjny. Zna zasady pracy w sieci, m.in. zasady udostępniania zasobów. Wie, z jakich warstw	Posługuje się terminologią sieciową. Potrafi wymienić zalety i wady różnych topologii sieci. Charakteryzuje topologie gwiazdy, magistrali i pierścienia. Zna znaczenie protokołu w sieciach (w tym TCP/IP). Definiuje funkcje i usługi poszczególnych warstw modelu warstwowego sieci. Wie, co to jest adres sieciowy. Wyjaśnia, co to jest adres domenowy i omawia jego strukturę. Potrafi omówić ogólne zasady administrowania siecią komputerową w	Swobodnie posługuje się terminologią sieciową. Zna schemat działania sieci komputerowych. Omawia ogólnie określanie ustawień sieciowych danego komputera i jego lokalizacji w sieci (podsieci IPv4, protokół DHCP, DNS, TCP). Wie, co określa maska podsieci. Potrafi z pomocą nauczyciela zrealizować małą sieć komputerową – skonfigurować jej składniki, udostępnić pliki, dyski, drukarki, dodać użytkowników.	Omawia szczegółowo model warstwowy sieci. Omawia różne systemy sieciowe. Dokonuje ich analizy porównawczej.

	składa się warstwowy model sieci. Ogólnie je omawia.	architekturze „klient-serwer”.		
Projektowanie relacyjnej bazy danych				
2	3	4	5	6
Podaje obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, instytucji naukowych, społecznych i gospodarczych. Podaje przykłady programów do tworzenia baz danych. Potrafi wykonać podstawowe operacje na bazie danych przygotowanej w jednej tabeli (wprowadzanie, redagowanie, sortowanie, wyszukiwanie, prezentacja). Potrafi utworzyć prostą kwerendę, jeden formularz i raport.	Rozumie metody przetwarzania danych na przykładzie gotowej bazy danych. Określa podstawowe pojęcia (rekord, pole, typ pola, relacja, klucz podstawowy). Tworzy bazę danych składającą się z dwóch tabel, w każdej po kilka pól różnych typów. Projektuje formularze i raporty. Tworzy proste kwerendy wybierające. Potrafi wykonywać operacje przetwarzania danych w bazie składającej się z kilku rekordów. Zna zasady przygotowania korespondencji seryjnej.	Projektuje relacyjną bazę danych (na zadany temat) składającą się z trzech tabel połączonych relacją. Omawia typy relacji w bazie danych. Zna zasady definiowania kluczy podstawowych. Projektuje formularze i raporty według wskazówek nauczyciela. Potrafi utworzyć formularz z podformularzem. Umieszcza przyciski nawigacyjne. Tworzy kwerendy wybierające. Importuje dane z tabel arkusza kalkulacyjnego i dokumentu tekstowego do tabel bazy danych. Eksportuje dane z tabel bazy danych do tabel arkusza kalkulacyjnego i do dokumentu tekstowego.	Potrafi wytłumaczyć pojęcie relacji. Projektuje relacyjną bazę danych składającą się z trzech lub większej liczby tabel. Samodzielnie ustala zawartość bazy (rodzaj informacji). Zna kilka rodzajów formularzy i raportów. Umie zaprojektować samodzielnie wygląd formularza i raportu. Na formularzach umieszcza pola kombi, ogranicza wartości, wstawia (gdy jest taka potrzeba) bieżącą datę, umieszcza przyciski poleceń. Stosuje funkcje standardowe w kwerendach i standardowe operatory w kryteriach wyszukiwania. Korzysta z parametrów w kwerendzie.	Zna dokładnie wybrany program do projektowania baz danych. Potrafi samodzielnie zaprojektować bazę danych, korzystając z wybranego narzędzia (programu). Projekt bazy opiera na rzeczywistych informacjach, aby można było wykorzystać ją w praktyce, np. w szkole czy w domu. Korzysta z dodatkowej, fachowej literatury.
Tworzenie kwerend z wykorzystaniem języka SQL				
2	3	4	5	6

<p>Wyszukuje informacje w bazie, korzystając wyłącznie z gotowych kwerend i narzędzi wbudowanych do programu.</p>	<p>Tworzy samodzielnie kwerendy (proste i złożone), korzystając z wbudowanych do programu narzędzi. Wie, co to jest język SQL. Potrafi przeanalizować przykład zapytania utworzonego w języku SQL.</p>	<p>Zna zasady wyszukiwania informacji w bazie z wykorzystaniem języka zapytań. Zna składnię i działanie podstawowych instrukcji. Potrafi zapisać prostą kwerendę, korzystając z języka zapytań.</p>	<p>Potrafi zapisać złożone kwerendy, korzystając z wybranej instrukcji. Stosuje instrukcję SELECT i jej główne klauzule, by wybrać kolumny z tabel bazy danych. Wykorzystuje klauzulę JOIN do łączenia informacji z wielu tabel i kwerend oraz przedstawiania wyników jako jednego logicznego połączenia rekordów. Stosuje instrukcje INSERT do dopisywania rekordów i UPDATE do modyfikowania rekordów w bazie. Usuwa rekordy, korzystając z instrukcji DELETE</p>	<p>Opierając się na profesjonalnej literaturze, potrafi samodzielnie zapisywać złożone kwerendy z wykorzystaniem języka zapytań SQL.</p>
---	--	---	---	--

Uzyskanie oceny wyższej wymaga spełnienia kryteriów na ocenę niższą