Wymagania edukacyjne Informatyka – semestr IV (Liceum wieczorowe / stacjonarne)

Temat	Wymagania konieczne (ocena dopuszczająca)	Wymagania podstawowe (ocena dostateczna)	Wymagania rozszerzające (ocena dobra)	Wymagania dopełniające (ocena bardzo dobra)	Wymagania wykraczające (ocena celująca)
	Słuchacz:	**Słuchacz:**	**Słuchacz:**	**Słuchacz:**	**Słuchacz:**
1. Lekcja organizacyjna BHP pracy z komputerem, zasady promowania i oceniania	Zapoznanie z Przedmiotowym Systemem Oceniania (PSO) w semestrze IV (bazy danych, technologie webowe, reprezentacja danych, zaawansowane algorytmy i robotyka), zasadami BHP, kryteriami sprawdzianu dopuszczeniowego, egzaminu semestralnego oraz warunkami ukończenia szkoły.				
Moduł D Wokół Internetu i projektów					
2. Temat D1. Wybrane przepisy prawa dotyczące technologii informacyjno-komunikacyjnych	– podaje pojęcie prawa autorskiego – wie, że nie wolno bezprawnie kopiować cudzych programów i utworów z sieci – podaje definicję danych osobowych	– rozróżnia autorskie prawa osobiste i majątkowe – wyjaśnia pojęcie dozwolonego użytku prywatnego i edukacyjnego – wymienia podstawowe licencje oprogramowania (komercyjne, freeware, shareware, Open Source) – charakteryzuje cel i zasady ochrony danych osobowych (RODO)	– charakteryzuje licencje Creative Commons (uznanie autorstwa, użycie niekomercyjne, bez utworów zależnych) – omawia prawne konsekwencje naruszenia praw autorskich i kradzieży tożsamości cyfrowej – charakteryzuje procedury zgłaszania cyberprzestępstw i naruszeń ochrony danych – omawia rolę instytucji ochrony prawnej (UODO, policja, CERT)	– przeprowadza analizę prawną i etyczną wykorzystania zasobów sieciowych w działalności gospodarczej – bezbłędnie dobiera i weryfikuje licencje zasobów (grafik, czcionek, kodu) stosowanych w projektach komercyjnych – projektuje podstawowe wytyczne polityki bezpieczeństwa danych osobowych dla małego przedsiębiorstwa	– krytycznie analizuje wyzwania prawne związane z rozwojem sztucznej inteligencji, deepfake oraz prawem własności do dzieł generatywnych – formułuje precyzyjne klauzule licencyjne i zgody na przetwarzanie danych osobowych w aplikacjach webowych – wykazuje biegłość w prawie nowych technologii i doradza w zakresie zgodności z prawem cyfrowym
3. Temat D2. Komunikacja i wymiana informacji w Internecie. Temat D3. Tworzenie stron WWW z wykorzystaniem stylów	– podaje podstawowe znaczniki języka HTML (np. `<html>`, `<body>`, `<p>`, `<h1>`) – otwiera plik HTML w przeglądarce internetowej – odróżnia treść strony od jej wyglądu graficznego	– tworzy poprawny szkielet dokumentu HTML5 z nagłówkiem i sekcją treści – stosuje podstawowe znaczniki formatowania tekstu, list punktowanych i hiperłączy (`<a>`) – dołącza zewnętrzny arkusz stylów CSS za pomocą znacznika `<link>` – definiuje podstawowe reguły CSS (kolor tekstu `color`, tło `background-color`, krój pisma `font-family`)	– poprawnie stosuje selektory znaczników, klas (`.klasa`) i identyfikatorów (`#id`) w arkuszu CSS – formatuje jednostki miar w CSS (piksele `px`, jednostki względne `em`, `rem`, `%`) – wstawia i formatuje obrazy (`<img>`) z zachowaniem atrybutów dostępności (`alt`) – tworzy czytelną, wielopodstronicową witrynę WWW połączoną menu nawigacyjnym	– projektuje logiczną strukturę witryny z wykorzystaniem semantycznych znaczników HTML5 (`<header>`, `<nav>`, `<main>`, `<section>`, `<footer>`) – stosuje zaawansowane reguły typograficzne i kaskadowość stylów (dziedziczenie, specyficzność selektorów) – dba o pełną walidację kodu HTML i CSS zgodnie ze standardami W3C	– samodzielnie projektuje nowoczesną, estetyczną i funkcjonalną stronę WWW o nienagannej strukturze semantycznej – stosuje zaawansowane selektory pseudoklas i pseudoelementów (`:hover`, `:nth-child`, `::after`) – optymalizuje zasoby witryny pod kątem dostępności cyfrowej (standard WCAG) i szybkości ładowania

4. Temat D3. Tworzenie stron WWW z elementami programowania	– wstawia tabelę na stronę WWW za pomocą znaczników `<table>`, `<tr>`, `<td>` – uruchamia stronę z prostym skryptem wyświetlającym okno dialogowe (`alert`) – wstawia przycisk `<button>` na stronę	– wyjaśnia i stosuje model pudełkowy (Box Model) w CSS: marginesy zewnętrzne (`margin`), obramowanie (`border`), dopełnienie (`padding`) – formatuje tabele w CSS (wyrównanie, kolory obramowania, tło naprzemienne) – wstawia prosty kod JavaScript obsługujący zdarzenie kliknięcia (`onclick`) – modyfikuje treść elementu HTML za pomocą prostego skryptu JS	– precyzyjnie kontroluje wymiary i położenie elementów na stronie z użyciem właściwości `box-sizing: border-box` – tworzy formularz kontaktowy w HTML (`<input>`, `<select>`, `<textarea>`) i formatuje go w CSS – pobiera wartości z pól formularza i przetwarza je w prostym skrypcie JavaScript – dodaje interaktywne efekty wizualne (np. zmiana koloru tła lub widoczności bloku po kliknięciu)	– projektuje responsywny formularz z walidacją poprawności wprowadzonych danych w JavaScript (np. sprawdzanie pustych pól, formatu liczbowego) – posługuje się manipulacją modelem DOM (`document.getElementById`, `querySelector`) – tworzy animacje i przejścia płynne w stylach CSS (`transition`, `@keyframes`)	– projektuje kompletną, interaktywną mini-aplikację webową (np. kalkulator internetowy, quiz wiedzy) – integruje skrypty JS z dynamicznym generowaniem elementów struktury HTML w czasie rzeczywistym – wykazuje biegłość w oddzielaniu warstwy struktury (HTML), prezentacji (CSS) i logiki biznesowej (JS)
Moduł A Wokół komputera					
5. Temat A1. Reprezentacja danych w komputerze oraz Temat A2. Wybrane urządzenia cyfrowe	– podaje nazwy systemów liczbowych: dwójkowy (binarny) i dziesiętny – wie, że komputer operuje na bitach (0 i 1) – wymienia współczesne urządzenia cyfrowe (smartfon, tablet, czytnik e-booków)	– przelicza małe liczby naturalne pomiędzy systemem dziesiętnym a dwójkowym (metoda wagowa lub dzielenia przez 2) – wyjaśnia jednostki bit i bajt oraz ich przedrostki wielokrotne – omawia zasadę kodowania tekstu za pomocą tablicy ASCII – charakteryzuje wybrane urządzenia cyfrowe i czujniki (sensory) w otoczeniu człowieka	– sprawnie przelicza liczby pomiędzy systemami: dziesiętnym, dwójkowym i szesnastkowym (heksadecymalnym) – wyjaśnia ideę kodowania znaków narodowych w standardzie Unicode / UTF-8 – omawia cyfrową reprezentację obrazu (piksel, głębia kolorów, model barw RGB i zapis szesnastkowy `#RRGGBB`) – analizuje parametry techniczne i możliwości nowoczesnych urządzeń cyfrowych (smartwatche, kamery cyfrowe, systemy IoT)	– wykonuje dodawanie i odejmowanie w systemie dwójkowym – szczegółowo analizuje sposób zapisu liczb całkowitych ze znakiem (kod U2) oraz liczb rzeczywistych (zmiennoprzecinkowych) – charakteryzuje formaty kompresji danych stratnej i bezstratnej (audio, wideo, grafika)	– wykazuje biegłość w arytmetyce komputerowej dowolnych systemów pozycyjnych – analizuje zjawiska przepełnienia bufora i błędów zaokrągleń w reprezentacji zmiennoprzecinkowej – przedstawia architekturę i zasady działania zaawansowanych systemów wbudowanych i sensorów sensorycznych w technologii Smart Home
Moduł B Wokół dokumentów komputerowych					
6. Temat B1. Tworzenie baz danych	– podaje pojęcie bazy danych i podaje przykład jej zastosowania (np. dziennik elektroniczny, katalog biblioteczny) – odróżnia rekord (wiersz) od pola (kolumny) w tabeli bazy danych – otwiera gotową bazę danych w	– wyjaśnia cechy relacyjnej bazy danych – projektuje prostą tabelę bazy danych w widoku projektu – dobiera podstawowe typy danych pól (krótki tekst, liczba, data/godzina, waluta, logiczny) – definiuje klucz podstawowy (Primary Key) i wyjaśnia rolę	– precyzyjnie konfiguruje właściwości pól tabeli (rozmiar pola, format wyświetlania, wymagane, reguła poprawności, komunikat o błędzie) – stosuje maski wprowadzania do ujednolicenia formatu danych (np. kod pocztowy, numer PESEL, telefon) – definiuje wartości domyślne dla pól tabeli	– projektuje spójną strukturę bazy danych składającej się z kilku tabel z zachowaniem zasad normalizacji (1NF, 2NF) – definiuje pola odnośnika (listy wyboru) pobierające dane z innej tabeli – optymalizuje strukturę tabel pod	– samodzielnie projektuje profesjonalny schemat bazy danych dla złożonego przedsiębiorstwa – przeprowadza pełny proces normalizacji bazy danych do trzeciej postaci normalnej (3NF) – implementuje zaawansowane reguły integralności semantycznej i

	programie MS Access lub LibreOffice Base	unikalnego identyfikatora rekordu – wprowadza i edytuje dane w widoku arkusza danych tabeli	– importuje dane do tabeli bazy danych z zewnętrznych arkuszy kalkulacyjnych lub plików tekstowych CSV	kątem integralności i eliminacji redundancji danych	audytu danych na poziomie definicji tabel
7. Temat B2. Wykonywanie podstawowych operacji na relacyjnej bazie danych	– sortuje rekordy w tabeli według wybranego pola rosnąco lub malejąco – stosuje prosty filtr do wyświetlenia rekordów spełniających jedno kryterium – tworzy prosty formularz wprowadzania danych za pomocą kreatora	– definiuje relację jeden-do-wielu (1:N) pomiędzy dwiema tabelami i wymusza więzy integralności referencyjnej – tworzy kwerendę (zapytanie) wybierające z użyciem kreatora lub siatki QBE – ustala kryteria wyboru w kwerendzie z operatorami relacyjnymi (`>`, `<`, `=`) oraz logicznymi (`AND`, `OR`) – generuje czytelny raport z bazy danych za pomocą kreatora raportów	– tworzy kwerendy wielotabelowe pobierające powiązane dane z kilku relacyjnych tabel – stosuje kwerendy z polami obliczeniowymi oraz funkcjami agregującymi (Suma, Średnia, Policz, Min, Max) – tworzy kwerendy parametryczne wymagające podania wartości przez użytkownika przy uruchomieniu – zapisuje podstawowe zapytania w języku SQL z klauzulami `SELECT`, `FROM`, `WHERE`, `ORDER BY`	– buduje zaawansowane zapytania SQL z grupowaniem (`GROUP BY`, `HAVING`) oraz złączeniami tabel (`INNER JOIN`, `LEFT JOIN`) – tworzy kwerendy funkcjonalne (aktualizujące, dołączające, usuwające rekordy) – projektuje intuicyjny, powiązany formularz główny z podformularzem do obsługi zamówień/relacji 1:N	– projektuje kompletny, zintegrowany system bazy danych z panelem nawigacyjnym, zabezpieczeniami i zaawansowanymi kwerendami SQL – optymalizuje wydajność zapytań bazodanowych na dużych wolumenach rekordów z wykorzystaniem indeksów – demonstruje pełną biegłość w pisaniu skomplikowanych skryptów SQL integrujących relacyjne bazy danych z aplikacjami zewnętrznymi
8. Temat B3. Projektowanie modeli dwuwymiarowych i trójwymiarowych	– uruchamia program do modelowania przestrzennego (np. SketchUp, Tinkercad) – porusza się w przestrzeni trójwymiarowej za pomocą narzędzi kamery (obrót, zoom, przesuwanie) – wstawia podstawowe bryły elementarne (sześcian, walec)	– rysuje płaskie kształty dwuwymiarowe (2D) i przekształca je w bryły 3D za pomocą narzędzia wyciągania (`Push/Pull`) – stosuje wymiarowanie do tworzenia obiektów o określonych wymiarach rzeczywistych – wykonuje operacje modyfikacji brył: przesuwanie, obrót, skalowanie – nadaje materiały i kolory powierzchniom projektowanego modelu	– tworzy złożony model przestrzenny (np. mebel, element architektoniczny, bryła techniczna) z zachowaniem proporcji i symetrii – posługuje się grupami i komponentami wielokrotnego użytku w projekcie 3D – wykonuje operacje wycinania otworów i modelowania wklęsłego/wypukłego w bryłach – eksportuje widok modelu do pliku graficznego 2D oraz modelu 3D	– projektuje kompletny model wnętrza lub obiektu użytkowego z precyzyjnym uwzględnieniem warstw, przekrojów i oświetlenia – optymalizuje geometrię siatki modelu przestrzennego pod kątem technologii druku 3D (format STL) – przygotowuje rendering modelu z realistycznymi cieniami i teksturami	– samodzielnie tworzy profesjonalny, zaawansowany projekt architektoniczny lub wzornictwa przemysłowego w technologii CAD/3D – przygotowuje model do bezbłędnego druku 3D w oprogramowaniu slicera z optymalizacją podpór i parametrów wypełnienia – łączy techniki modelowania parametrycznego z wizualizacją fotorealistyczną
9. Sprawdzian dopuszczeniowy	Praktyczny sprawdzian przy stanowisku komputerowym weryfikujący umiejętności z zakresu relacyjnych baz danych (tabele, relacje, kwerendy SQL), tworzenia stron WWW w HTML/CSS oraz systemów liczbowych – warunek konieczny dopuszczenia do egzaminu semestralnego kończącego naukę w liceum.				
Moduł C Wokół algorytmiki i programowania					
10. Temat C1. Algorytmy na tekstach w językach C++ i Python	– tworzy zmienną przechowującą napis (tekst) w wybranym języku – łączy dwa napisy (konkatenacja	– odwołuje się do pojedynczych znaków napisu za pomocą indeksów (np. `napis[0]`)	– sprawnie posługuje się wycinaniem fragmentów napisów (slicing: `napis[start:stop]`) w Pythonie	– implementuje algorytm badania anagramów (czy dwa słowa składają się z tych samych liter)	– implementuje zaawansowane algorytmy dopasowywania wzorca w tekście (np. algorytm KMP lub

	operatorem `+`) – odczytuje długość napisu za pomocą funkcji `len()` (Python) lub metody `.length()` (C++)	– przegląda znaki napisu z wykorzystaniem pętli `for` – sprawdza równość dwóch tekstów za pomocą operatora `==` – zlicza wystąpienia określonego znaku w podanym tekście	– implementuje algorytm sprawdzania, czy podane słowo jest palindromem (czyta się tak samo od początku i od końca) – programuje wyszukiwanie znaku i podciągu w tekście bez użycia gotowych funkcji wbudowanych – stosuje wbudowane metody manipulacji tekstem (zamiana liter na wielkie/małe, zamiana znaków `replace`)	– programuje algorytm wyszukiwania wzorca w tekście metodą naiwną i analizuje jego złożoność – tworzy zaawansowane procedury walidacji danych tekstowych (np. weryfikacja poprawności formatu kodu pocztowego, numeru PESEL)	Karpa-Rabina) – programuje leksykograficzne porządkowanie ciągów tekstowych oraz obliczanie odległości Levenshteina pomiędzy słowami – wykazuje biegłość w przetwarzaniu języka naturalnego i tokenizacji tekstu w aplikacjach użytkowych
11. Temat C2. Algorytmy szyfrowania w językach C++ i Python	– podaje pojęcia: kryptografia, tekst jawny, szyfrogram, klucz szyfrowania – wyjaśnia cel stosowania szyfrowania w informatyce – odczytuje kod ASCII pojedynczego znaku za pomocą funkcji `ord()`	– wyjaśnia zasadę działania klasycznego szyfru podstawieniowego na przykładzie szyfru Cezara (przesunięcie w alfabecie o stały klucz k) – przelicza kody znaków z użyciem funkcji `ord()` oraz `chr()` (Python) – implementuje prosty algorytm szyfrujący pojedyncze litery szyfrem Cezara o zadanym stałym kluczu – rozróżnia szyfrowanie symetryczne od asymetrycznego	– implementuje pełny program szyfrujący i deszyfrujący dowolny tekst szyfrem Cezara z uwzględnieniem zawijania alfabetu (operacja modulo `%`) – programuje obsługę spacji i znaków interpunkcyjnych podczas szyfrowania tekstu – implementuje szyfr przestawieniowy (np. szyfrowanie w kolumnach macierzy) – testuje program szyfrujący dla różnych wartości klucza	– implementuje algorytm kryptoanalizy (łamania) szyfru Cezara metodą ataku brutalnego (brute-force – testowanie wszystkich 26 przesunięć) – programuje algorytm łamania szyfru z wykorzystaniem analizy częstościowej występowania liter w języku polskim – charakteryzuje współczesne standardy kryptograficzne (AES, RSA) i protokół HTTPS/SSL	– samodzielnie implementuje zaawansowany algorytm kryptograficzny (np. szyfr Vigenère'a z kluczem słownym) – projektuje i programuje bezpieczny system szyfrowania i uwierzytelniania wiadomości w sieci z generowaniem skrótów kryptograficznych (funkcje hashujące SHA) – demonstruje głęboką wiedzę z zakresu kryptografii kwantowej i nowoczesnego bezpieczeństwa danych
12. Temat C3. Algorytmy na liczbach w językach C++ i Python	– stosuje operatory dzielenia całkowitego (`//`) oraz reszty z dzielenia (`%`) w prostych działaniach – uruchamia gotowy program obliczający sumę cyfr liczby – podaje ostatnią cyfrę liczby naturalnej	– wyjaśnia algorytm wyodrębniania kolejnych cyfr liczby naturalnej za pomocą pętli `while` oraz operacji `% 10` i `// 10` – implementuje algorytm obliczania sumy cyfr podanej liczby naturalnej – programuje zliczanie cyfr parzystych i nieparzystych w liczbie – dobiera dane testowe sprawdzające poprawność obliczeń dla liczb jednocyfrowych i wielocyfrowych	– implementuje algorytm odwracania kolejności cyfr liczby naturalnej (np. z 1234 tworzy 4321) – programuje badanie, czy liczba jest palindromem liczbowym – implementuje iteracyjny algorytm rozkładu liczby naturalnej na czynniki pierwsze – programuje sprawdzanie, czy liczba jest liczbą doskonałą (równą sumie swoich dzielników właściwych)	– implementuje algorytm szybkiego potęgowania modularnego $a^b \bmod m$ i analizuje jego złożoność – programuje algorytm generowania liczb Fibonacciego w postaci iteracyjnej z optymalizacją zużycia pamięci – implementuje obliczanie cyfr kontrolnych w numerach identyfikacyjnych (np. walidacja sumy kontrolnej numeru PESEL lub ISBN)	– projektuje i implementuje wydajne algorytmy arytmetyki wielkich liczb (operacje arytmetyczne na liczbach o setkach cyfr zapisanych w strukturach tablicowych) – formalnie analizuje i dowodzi optymalności algorytmów numerycznych – tworzy zaawansowane biblioteki funkcji teorii liczb znajdujące zastosowanie w systemach informatycznych

13. Temat C5. Metoda połowienia, podejście zachłanne i rekurencja	– podaje różnicę między przeszukiwaniem po kolei (liniowym) a szukaniem w uporządkowanym słowniku – wie, że funkcja może wywołać samą siebie – uruchamia program wydający resztę za pomocą monet	– omawia ideę wyszukiwania binarnego (metoda połowienia) w posortowanym zbiorze danych – wyjaśnia zasadę działania algorytmu zachłannego na przykładzie problemu wydawania reszty – definiuje pojęcie rekurencji i wskazuje warunek stopu wywołania rekurencyjnego – implementuje prosty algorytm wydawania reszty monetami o zadanych nominałach	– implementuje algorytm wyszukiwania binarnego w posortowanej liście/tablicy w wybranym języku – porównuje liczbę porównań w wyszukiwaniu binarnym i liniowym, wskazując przewagę złożoności $O(\log n)$ – implementuje klasyczną funkcję rekurencyjną: obliczanie silni ($n!$) lub ciągu Fibonacciego – analizuje zachowanie algorytmu zachłannego i wskazuje przypadki, w których nie daje on rozwiązania optymalnego	– implementuje algorytm wyszukiwania miejsc zerowych funkcji metodą połowienia (bisekcji) z zadaną dokładnością ε – przekształca algorytmy rekurencyjne w wydajne odpowiedniki iteracyjne i odwrotnie – rozwiązuje klasyczny problem wież Hanoi z wykorzystaniem elegancji zapisu rekurencyjnego	– projektuje i programuje zaawansowane algorytmy oparte na paradygmacie „dziel i zwyciężaj” (Divide and Conquer) – formalnie analizuje stos wywołań rekurencyjnych, zapobiega błędowi przepełnienia stosu (Stack Overflow) i stosuje memoizację – dowodzi poprawności podejścia zachłannego dla zadanych systemów monetarnych (systemy kanoniczne)
14. Temat C6. Elementy robotyki	– podaje definicję robota i wymienia przykłady robotów przemysłowych oraz domowych – wymienia podstawowe elementy robota (komputer sterujący, silnik, kółka, czujnik) – uruchamia wirtualne środowisko symulacji robota	– charakteryzuje rolę mikrokontrolera jako „mózgu” układu robotycznego – rozróżnia sensory (czujnik odległości, światła, koloru, dotyku) od aktuatorów (silniki, serwomechanizmy) – tworzy prosty algorytm sterowania ruchem robota po linii prostej lub po zadanym torze – omawia wpływ automatyzacji i robotyzacji na współczesny rynek pracy	– programuje robota do omijania przeszkód na podstawie odczytów z ultradźwiękowego czujnika odległości – implementuje algorytm podążania za linią (Line Follower) z wykorzystaniem czujników odbiciowych światła – posługuje się pętlą główną programu sterującego mikrokontrolerem i odczytem wejść analogowych/cyfrowych – testuje poprawność zachowania robota w symulatorze lub na zestawie edukacyjnym	– projektuje rozbudowany algorytm autonomicznego poruszania się robota w nieznanym labiryncie (metoda prawej dłoni) – integruje dane z wielu czujników jednocześnie, eliminując zakłócenia i szum pomiarowy – analizuje etyczne, prawne i społeczne konsekwencje autonomii robotów i sztucznej inteligencji	– samodzielnie buduje i programuje zaawansowany prototyp robota mobilnego lub ramienia manipulatora – implementuje regulatory sterowania ruchem (np. prosty regulator proporcjonalny P) dla płynnej jazdy robota – wykazuje biegłość w programowaniu systemów mechatronicznych i architekturze Przemysłu 4.0
Moduł D Wokół Internetu i projektów					
15. Temat D1. Więcej na temat tworzenia stron internetowych	– wie, czym jest responsywność strony internetowej (dostosowanie do smartfona) – otwiera stronę WWW w trybie responsywnym w narzędziach deweloperskich przeglądarki – wstawia proste elementy	– wyjaśnia zasady Responsive Web Design (RWD) i rolę znacznika `<meta name="viewport">` – stosuje reguły zapytań medialnych (`@media`) w stylach CSS do zmiany układu strony na urządzeniach mobilnych – tworzy elastyczne obrazy i kontenery skalujące się wraz z	– posługuje się nowoczesnym modelem układu Flexbox w CSS (`display: flex`, `justify-content`, `align-items`) – projektuje responsywne, rozwijane menu nawigacyjne dla wersji mobilnej i desktopowej – integruje stronę WWW z zewnętrznymi API (np. interaktywna	– projektuje zaawansowany, responsywny układ wielokolumnowy z wykorzystaniem technologii CSS Grid – optymalizuje wydajność witryny (kompresja zasobów, formaty WebP, czysty kod, minifikacja plików) – przeprowadza audyt witryny pod	– projektuje i wdraża w środowisku produkcyjnym kompletną, profesjonalną witrynę internetową o nienagannej architekturze informacji – integruje stronę z nowoczesnymi narzędziami analitycznymi i dynamicznymi frameworkami frontendowymi

	multimedialne (wideo, audio) za pomocą znaczników HTML5	szerokością okna przeglądarki – publikuje gotową stronę w sieci internetowej (np. darmowy serwer WWW, GitHub Pages)	mapa Google, widżet mediów społecznościowych) – zarządza plikami strony na zdalnym serwerze za pomocą protokołu FTP lub systemu kontroli wersji Git	kątem dostępności (WCAG 2.1), semantyki i pozycjonowania (SEO)	– demonstruje mistrzowski poziom znajomości ekosystemu webowego i standardów sieciowych
16. Egzamin semestralny	Praktyczny egzamin semestralny przy stanowisku komputerowym sprawdzający wiadomości i umiejętności z semestru IV oraz całego cyklu kształcenia: relacyjne bazy danych (projekt tabeli, relacja, kwerendy SQL), technologie tworzenia stron WWW (HTML5/CSS) oraz zaawansowana algorytmiczna analiza i implementacja kodu (wyszukiwanie, szyfrowanie, operacje na liczbach/tekstach).				
17. Omówienie wyników egzaminu	Przedstawienie i szczegółowe omówienie wyników końcowego egzaminu semestralnego, analiza prac praktycznych, podsumowanie osiągnięć edukacyjnych słuchacza z całego 4-semestralnego toku nauki informatyki, wystawienie ocen końcowych i zatwierdzenie ukończenia liceum ogólnokształcącego.				