

Wymagania edukacyjne dla przedmiotu informatyka dla klasy II liceum ogólnokształcącego i technikum w zakresie rozszerzonym, uwzględniający kształcone umiejętności i treści podstawy programowej.

Uwaga!

W planie pominięto podstawowe umiejętności, które uczeń powinien już posiadać, np. zachowywanie plików projektów, wczytywanie dokumentów do edycji i posługiwanie się systemem operacyjnym.

Kryteria danej oceny opracowano z założeniem, że zostały spełnione kryteria ocen niższych.

Temat	Ocena dopuszczająca Uczeń:	Ocena dostateczna Uczeń:	Ocena dobra Uczeń:	Ocena bardzo dobra Uczeń:	Ocena celująca Uczeń:
I. Opracowania rozwiązań problemów wybranymi aplikacjami					
1. Host i serwer, czyli o działaniu sieci	<ul style="list-style-type: none"> – na podstawie podręcznika i z pomocą nauczyciela omawia podstawowe pojęcia związane z budową sieci komputerowych (np. <i>host, serwer, klient, LAN, WAN, WiFi</i>); – wie, czym są media transmisyjne; – wymienia nazwy najczęściej spotykanych topologii sieci lokalnej i na podstawie ilustracji (np. z podręcznika) wymienia ich podstawowe cechy. 	<ul style="list-style-type: none"> – samodzielnie omawia znaczenie podstawowych pojęć związanych z budową i działaniem sieci komputerowych; – na podstawie podręcznika omawia cechy poszczególnych mediów transmisyjnych; – samodzielnie wymienia nazwy najczęściej spotykanych topologii sieci lokalnej i wymienia ich podstawowe cechy; – wymienia jednostki przesyłu informacji i omawia zależności pomiędzy nimi; – wie, że do różnych zastosowań można używać różnych rodzajów kabla sieciowego. 	<ul style="list-style-type: none"> – samodzielnie omawia znaczenie wszystkich pojęć związanych z budową i działaniem sieci komputerowych; – oblicza maksymalną ilość danych przesyłanych w sieci na podstawie danych o czasie i przepustowości łącza; – z niewielką pomocą nauczyciela wymienia nazwy i cechy podstawowych kabli dla sieci lokalnych CAT 3, 5 i 5e, 6 i 7; – wymienia podstawowe dla transmisji danych cechy światłowodu. 	<ul style="list-style-type: none"> – wybiera odpowiednią topologię i medium dla różnych sieci lokalnych; – identyfikuje topologię istniejącej sieci (np. w pracowni); – identyfikuje rodzaje mediów transmisyjnych; – prawidłowo stosuje poznane pojęcia dotyczące sieci lokalnych. 	<ul style="list-style-type: none"> – omawia niewymienione w podstawie programowej zagadnienia związane z tematem.

<p>2. Włączamy do ruchu, czyli konfiguracja lokalnej sieci komputerowej</p>	<ul style="list-style-type: none"> – omawia funkcję routera w sieci lokalnej; – wie, że router należy odpowiednio skonfigurować; – podaje, jak znaleźć opis konfiguracji routera (np. instrukcję, samouczek w sieci); – identyfikuje nazwę i producenta routera. 	<ul style="list-style-type: none"> – omawia funkcję DHCP w kontekście konfiguracji routera; – na podstawie podręcznika lub innych wiarygodnych źródeł wyjaśnia pojęcia: <i>DHCP</i>, <i>serwer</i>, <i>szyfrowanie</i> (np. <i>WPA2-PSK</i>) i ich znaczenie dla konfiguracji routera; – na podstawie instrukcji i z pomocą nauczyciela omawia lub przeprowadza konfigurację przykładowego routera. 	<ul style="list-style-type: none"> – na podstawie instrukcji omawia lub przeprowadza konfigurację przykładowego routera; – na podstawie instrukcji lub opcji Pomocy omawia lub przeprowadza konfigurację przykładowego routera bez wykorzystania DHCP. 	<ul style="list-style-type: none"> – samodzielnie omawia lub przeprowadza konfigurację przykładowego routera z wykorzystaniem DHCP i bez tego mechanizmu. 	<ul style="list-style-type: none"> – dokonuje konfiguracji zaawansowanych funkcji routera (np. uprawnień poszczególnych użytkowników).
<p>3. Nie tylko WiFi, czyli bezprzewodowy dostęp do internetu</p>	<ul style="list-style-type: none"> – wie czym się różni WLAN od WiFi; – omawia zastosowanie złączy Bluetooth do połączenia urządzeń komputerowych; – z niewielką pomocą nauczyciela przeprowadza przesłanie pliku (np. z telefonu do laptopa) za pośrednictwem Bluetooth. 	<ul style="list-style-type: none"> – wymienia nazwy i omawia przeznaczenie podstawowych mediów i urządzeń transmisyjnych w tym: WLAN, WiFi, Bluetooth; – odróżnia i prawidłowo przypisuje do danych mediów ich symbole graficzne. 	<ul style="list-style-type: none"> – wyjaśnia, na czym polega łączenie urządzeń za pomocą fal radiowych i podaje, jakie warunki muszą być przy tym spełnione; – na podstawie opisów omawia różnice między mediami transmisyjnymi; – wyjaśnia znaczenie norm określających ramy działania poszczególnych mediów transmisyjnych. 	<ul style="list-style-type: none"> – wymienia pasma radiowe przeznaczone dla mediów transmisyjnych stosowanych w sieciach bezprzewodowych; – opisuje możliwości wykorzystania mediów bezprzewodowych i samodzielnie przeprowadza konfigurację podłączanych do nich urządzeń (np. słuchawek, telefonu, laptopa). 	<ul style="list-style-type: none"> – omawia zastosowania i działanie WiMAX; – samodzielnie łączy klika urządzeń jednocześnie za pomocą Bluetooth.
<p>4. Zespół, czyli realizujemy projekty w chmurze</p>	<ul style="list-style-type: none"> – omawia budowę przykładowej chmury, wymieniając jej podstawowe składniki. 	<ul style="list-style-type: none"> – przedstawia i omawia przykładowy schemat struktury chmury informatycznej; – wymienia najczęściej 	<ul style="list-style-type: none"> – omawia przeznaczenie poszczególnych elementów chmury użytej na zajęciach; – porównuje działanie 	<ul style="list-style-type: none"> – biegle posługuje się programami i dyskami sieciowymi; – samodzielnie przygotowuje chmurę do 	<ul style="list-style-type: none"> – samodzielnie planuje i organizuje pracę zespołu w chmurze; – posługuje się kilkoma chmurami (np. Google

		<p>spotykane w chmurach narzędzia (m.in. dysk, edytor, arkusz, kalendarz);</p> <p>– wie, że chmurę można wykorzystać do pracy zespołowej.</p>	<p>programów z chmury z ich odpowiednikami z aplikacji komputerowych;</p> <p>– posługuje się chmurą w stopniu wystarczającym do wykonywania podstawowych czynności, takich jak edycja dokumentu, kopiowanie plików itp.;</p> <p>– wie, które aplikacje można wykorzystać do organizacji pracy zespołu.</p>	<p>pracy zespołu, m.in. konfiguruje kalendarz, udostępnia foldery i pliki.</p>	<p>i OneDrive).</p>
<p>5. Tutorial, czyli jak stworzyć pomoce i instrukcje obsługi</p>	<p>– wie czym są style i szablony;</p> <p>– zmienia styl w trakcie edycji dokumentu.</p>	<p>– prawidłowo dobiera style do treści zawartej w dokumencie;</p> <p>– wie, że można samodzielnie definiować style i szablony;</p> <p>– uruchamia edycję, wykorzystując szablon oferowany przez edytor.</p>	<p>– dobiera styl i szablon zgodnie z planowaną zawartością i tematem dokumentu;</p> <p>– na podstawie podręcznika definiuje szablon i styl dokumentu;</p> <p>– odnajduje w sieci (np. w chmurze) szablony dla różnych dokumentów.</p>	<p>– samodzielnie definiuje szablon i styl dokumentu;</p> <p>– projektuje szablon zgodnie z treścią przyszłego dokumentu;</p> <p>– modyfikuje i projektuje nowe style.</p>	<p>– projektuje szablony dla różnych edytorów (np. LibreOffice Draw) i ich używa.</p>
<p>6. Rozbudowana struktura, czyli korzystamy z konspektu w edytorze tekstu</p>	<p>– rozumie i omawia pojęcie konspektu;</p> <p>– podaje, czym jest akapit i jaką pełni funkcję w edycji i formatowaniu tekstu.</p>	<p>– na podstawie podręcznika tworzy konspekt dokumentu;</p> <p>– na podstawie podręcznika dzieli dokument na sekcje i kolumny.</p>	<p>– samodzielnie tworzy konspekt dokumentu;</p> <p>– samodzielnie dzieli dokument na sekcje i kolumny;</p> <p>– tworzy spis treści na podstawie konspektu.</p>	<p>– uzasadnia stosowanie podziału dokumentu tekstowego na sekcje lub kolumny;</p> <p>– stosuje różny podział na niektórych stronach (sekcjach) dokumentu.</p>	<p>– stosuje podziały i sekcje w różnych edytorach tekstu.</p>
<p>7. Broszura, czyli jak projektować duże dokumenty</p>	<p>– wstawia do dokumentów SmartArt i Kształty.</p>	<p>– uzasadnia wybór danego kształtu lub ilustracji SmartArt;</p> <p>– na podstawie podręcznika tworzy spisy ilustracji</p>	<p>– na podstawie podręcznika lub tutoriali zmienia domyślne opcje edytora;</p> <p>– na podstawie podręcznika lub tutoriali wyko-</p>	<p>– samodzielnie zmienia domyślne opcje edytora;</p> <p>– samodzielnie wykonuje automatyczne spisy treści, tabel i ilustracji;</p>	<p>– spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice).</p>

		i tabel.	nuje automatyczne spisy treści, tabel i ilustracji; – na podstawie podręcznika lub tutoriali aktualizuje spisy po zmianach.	– samodzielnie aktualizuje spisy po zmianach; – decyduje, które z opcji domyślnych zmienić i uzasadnia swój wybór.	
8. Recenzja, czyli proponujemy poprawki w tekście	– odczytuje i interpretuje dokumenty, w których zastosowano tryb recenzji; – odczytuje notatki i zaznaczenia w Adobe Acrobat Reader DC.	– na podstawie podręcznika używa opcji Recenzja w Word i wstawiania komentarzy w dokumencie PDF.	– na podstawie podręcznika lub tutoriali używa opcji Recenzja do wpisywania komentarzy w Word oraz opcji wstawiania komentarzy i zaznaczania fragmentów w dokumencie PDF; – na podstawie podręcznika lub tutoriali uruchamia opcję śledzenia zmian.	– samodzielnie używa opcji Recenzja do wpisywania komentarzy w Word oraz opcji wstawiania komentarzy i zaznaczania fragmentów w dokumencie PDF; – samodzielnie uruchamia opcję śledzenia zmian; – świadomie używa opcji Recenzja do proponowania zmian w dokumencie; – analizuje wynik działania opcji porównywania dokumentów.	– spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice); – przedstawia dokument z innego przedmiotu lub projektu, w którym użył opcji Recenzja dla dokumentu Word lub Notatka dla PDF.
9. OLE, czyli łączymy dokumenty w dokumentacji pracy zespołu	– na podstawie podręcznika wyjaśnia, czym jest i do czego służy technika OLE.	– wyjaśnia, jakie obiekty mogą być wstawiane za pomocą techniki OLE; – na podstawie podręcznika lub opcji Pomoc umieszcza w dokumencie obiekty OLE; – na podstawie podręcznika wyjaśnia, czym są OLE2 i ActiveX; – wyjaśnia, dlaczego podczas używania ActiveX należy zachować szczególną ostrożność.	– na podstawie podręcznika i opcji Pomocy przetwarza obiekty OLE w edytorach tekstu itp.	– samodzielnie przetwarza obiekty OLE w edytorach tekstu itp.	– spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice).

II. Rozwijanie kompetencji społecznych					
10. Informatyka pokonuje schody, czyli nikt nie powinien być wykluczony	– wymienia cechy aplikacji ratujących życie lub ułatwiających funkcjonowanie osobom niepełnosprawnym.	– znajduje i instaluje wiarygodne aplikacje ratujące życie lub zdrowie; – wymienia urządzenia peryferyjne do monitorowania parametrów organizmu; – posługuje się tłumaczem języków.	– ocenia jakość aplikacji ratujące życie lub zdrowie; – ocenia jakość aplikacji ułatwiających funkcjonowanie osobom niepełnosprawnym i starszym; – wymienia cechy strony dostosowanej do potrzeb osób niepełnosprawnych; – używa urządzeń monitorujących stan organizmu.	– dobiera peryferia i aplikacje pomagające w ratowaniu życia lub zdrowia; – omawia cechy dobranych przez siebie aplikacji i porównuje ich jakość; – pokazuje przykłady stron przystosowanych do potrzeb osób niepełnosprawnych.	– projektuje strony przystosowane do potrzeb osób niepełnosprawnych; – korzysta z tłumacza w trybie offline.
11. Platformy uczą, czyli rola e-learningu w nauce i pracy	– wymienia podstawowe różnice między nauczaniem zdalnym a klasycznym.	– wymienia zalety i wady e-learningu; – umie skorzystać z gotowych lekcji e-learningowych.	– opisuje przykładową strukturę lekcji e-learningowej; – opisuje przykładowy cykl nauczania e-learningowego.	– wymienia właściwości wybranego systemu do tworzenia i prowadzenia kursów e-learningowych; – porównuje zalety i wady e-learningu z nauczaniem tradycyjnym; – znajduje w sieci kursy e-learningowe na dany temat i odczytuje, na jakich warunkach można w nich uczestniczyć.	– układa scenariusz lekcji e-learningowej i wskazuje różnice w porównaniu z tradycyjną lekcją.
12. Możesz być administratorem, czyli jak zarządzać platformą e-learningową	– omawia warunki, jakie musi spełniać platforma e-learningowa.	– uruchamia aplikację Classroom; – wymienia elementy, jakie powinny się znaleźć w strukturze kursu e-learningowego; – na podstawie podręcznika tworzy lekcje	– projektuje przykładową strukturę kursu e-learningowego; – sporadycznie korzystając z podręcznika, tworzy lekcję i dodaje temat zajęć w Classroom.	– projektuje strukturę kursu e-learningowego na zadany temat i tworzy ją w Classroom; – testuje poprawność działania kursu.	– układa kurs e-learningowy w innej platformie niż Classroom.

		w Classroom.			
13. Zasoby i testy, czyli wypełniamy kursy treścią	– wymienia rodzaje zasobów, które można umieszczać w kursach e-learningowych.	– objaśnia rolę poszczególnych rodzajów zasobów kursów e-learningowych; – wymienia rodzaje pytań testowych, jakie mogą się znaleźć w testach online.	– dzieli zasoby kursów według różnych kryteriów; – sporadycznie korzystając z podręcznika, umieszcza w Classroom gotowe zasoby edukacyjne i testy; – zaprasza użytkowników do korzystania z kursu.	– samodzielnie tworzy część zasobów, które umieści w Classroom; – tworzy krótkie testy w Classroom.	– umieszcza zasoby w innej platformie niż Classroom.
14. Dzielimy się wiedzą, czyli współtworzymy zasoby udostępniane na platformach	– uzasadnia potrzebę współdzielenia zasobów chmury lub platformy e-learningowej.	– z pomocą nauczyciela lub na podstawie podręcznika udostępnia zasoby umieszczone w chmurze lub platformie.	– samodzielnie udostępnia zasoby na platformie lub chmurze w wybrany przez siebie sposób.	– samodzielnie udostępnia zasoby, korzystając z różnych sposobów oferowanych przez oprogramowanie chmury lub platformy.	– wykonuje czynności opisane w podręczniku także w innych chmurach i platformach.
III. Bazy danych					
15. Jak gromadzić informacje, czyli komputerowe bazy danych	– podaje przykłady baz danych, z których korzysta np. w swoim telefonie; – definiuje pojęcie bazy danych.	– wymienia wszystkie elementy bazy danych opisane w podręczniku.	– omawia przeznaczenie poszczególnych elementów bazy danych.	– samodzielnie opisuje czynności występujące w trakcie projektowania bazy danych.	– opisuje zależności między poszczególnymi elementami bazy danych.
16. Tabele i formularze, czyli jak utworzyć bazę danych	– wymienia przykładowe dane, jakie powinna zawierać baza uczestników projektu.	– uzasadnia wybór danych dla bazy uczestników projektu; – na podstawie podręcznika tworzy tabele bazy danych zgodnie z danymi zdefiniowanymi dla uczestników projektu; – rozumie znaczenie tabel w bazie danych.	– samodzielnie tworzy kolejne tabele projektu; – korzystając z kreatora odnośników, tworzy je dla pól tabeli, wybierając je spośród wskazanych w podręczniku; – umieszcza dane w tabelach; – omawia znaczenie klucza; – prawidłowo posługuje się pojęciami związanymi	– tworzy klucze; – tworzy odnośniki dla pól tabel; – tworzy formularze.	– czynności opisane dla Access wykonuje także w innym systemie baz danych (np. LibreOffice Base).

			z bazami danych.		
17. Relacje i pytania, czyli jak uczynić bazę użyteczną	– wyjaśnia znaczenie relacji między tabelami bazy.	– wyjaśnia różnice między rodzajami relacji.	– tworzy klucz zewnętrzny; – używa kreatora relacji między tabelami; – wypełnia pola tabel powiązanych za pomocą kreatora; – tworzy relacje za pomocą narzędzia Relacje na podstawie opisu w podręczniku.	– tworzy relacje za pośrednictwem kreatora i narzędzia Relacje; – tłumaczy sens tworzenia relacji i efekt ich działania w przykładzie z ćwiczeń.	– czynności opisane dla Access wykonuje także w innym systemie baz danych (np. LibreOffice Base).
18. Kto pyta, nie błądzi, czyli jak korzystać z baz danych Access	– wyjaśnia znaczenie pojęcia <i>kwerenda</i> ; – wyjaśnia, na czym polega filtrowanie w informatyce.	– na podstawie podręcznika tworzy kwerendę, używając kreatora; – na podstawie podręcznika lub tutoriali modyfikuje kwerendę, dodając filtrowanie; – na podstawie podręcznika lub tutoriali sortuje dane i tworzy raport za pomocą kreatora.	– samodzielnie tworzy kwerendę, używając kreatora; – samodzielnie modyfikuje kwerendę, dodając filtrowanie; – samodzielnie sortuje dane i tworzy raport za pomocą kreatora.	– planuje kwerendę i ją opracowuje; – prawidłowo decyduje o wyborze filtrowania, sposobie sortowania i raportowania; – wykonuje eksport tabel do innych formatów (np. Excel).	– czynności opisane dla Access wykonuje także w innym systemie baz danych (np. LibreOffice Base).
19. Formularze, czyli ułatwiamy sobie wprowadzenie informacji do bazy danych	– z pomocą nauczyciela omawia przykład zamieszczony w podręczniku.	– wymienia nazwy różnych rodzajów formularzy i na podstawie podręcznika omawia ich cechy i zastosowanie; – na podstawie opisów tworzy różne rodzaje formularzy za pomocą kreatora.	– przy tworzeniu formularzy korzysta z różnych rodzajów list.	– konstruuje formularze z widoku projektu, wykorzystując formaty; – używa przycisków nawigacyjnych w formularzu.	– czynności opisane dla Access wykonuje także w innym systemie baz danych (np. LibreOffice Base).
20. Czy to możliwe, czyli baza danych w arkuszu Excel	– omawia analogie w tworzeniu tabel w Access i Excel.	– na podstawie podręcznika lub tutoriali tworzy w Excelu tabele	– samodzielnie tworzy w Excelu tabele z nagłówkami;	– planuje i wykonuje czynności tworzenia bazy w Excelu;	– czynności opisane dla Excels wykonuje także w innym arkuszu (np.

		<p>z nagłówkami;</p> <ul style="list-style-type: none"> – na podstawie podręcznika lub tutoriali używa formularzy do wprowadzania danych do tabel; – na podstawie podręcznika lub tutoriali tworzy listę rozwijaną. 	<ul style="list-style-type: none"> – samodzielnie używa formularzy do wprowadzania danych do tabel; – samodzielnie tworzy listę rozwijaną; – stosuje sortowanie według wskazanych kryteriów; – wstawia narzędzia do paska Szybki dostęp. 	<ul style="list-style-type: none"> – dobiera kryteria sortowania; – przenosi tabele z bazy danych do Excela. 	LibreOffice Calc).
21. Wiedza w sieci, czyli internet mądrych ludzi	<ul style="list-style-type: none"> – wyszukuje informacje w domyślnej wyszukiwarce przeglądarki internetowej; – omawia znaczenie zachowania praw autorskich i podstawy licencji CC. 	<ul style="list-style-type: none"> – na podstawie podręcznika zmienia domyślną wyszukiwarkę w przeglądarce Firefox; – na podstawie podręcznika uszczegółowia dane w wyszukiwarce w celu zwiększenia prawdopodobieństwa dotarcia do szukanej informacji; – na podstawie podręcznika wyszukuje za pomocą obrazu. 	<ul style="list-style-type: none"> – uzasadnia celowość korzystania z różnych wyszukiwarek; – pozyskuje informacje z baz danych, takich jak europeana.eu; – tłumaczy podstawy i cel działania botów indeksujących; – tłumaczy na przykładzie konieczność uszczegółowienia zapytań do wyszukiwarki; – korzysta z zaawansowanych opcji wyszukiwarek; – ocenia wiarygodność źródeł, wymieniając świadczące o niej cechy. 	<ul style="list-style-type: none"> – planuje użycie odpowiedniej wyszukiwarki; – wybiera bazę danych do wyszukiwania konkretnych informacji; – trafnie dobiera słowa kluczowe, zarówno w opcjach głównych, jak i zaawansowanych wyszukiwarek; – zmienia wyszukiwarki domyślne w różnych przeglądarkach internetowych. 	– omawia cechy różnych wyszukiwarek internetowych i ocenia ich przydatność do konkretnego zadania.
22. Wirtualne serwery, czyli instalujemy Apache	<ul style="list-style-type: none"> – omawia pojęcie <i>serwer</i>; – uruchamia środowisko XAMP. 	<ul style="list-style-type: none"> – na podstawie podręcznika i doświadczeń z poprzednich lekcji instaluje i uruchamia środowisko XAMP. 	<ul style="list-style-type: none"> – samodzielnie instaluje środowisko XAMP, wybierając wskazane komponenty; – omawia zawartość XAMP i jego 	<ul style="list-style-type: none"> – decyduje o wyborze komponentów na podstawie informacji o późniejszym wykorzystaniu XAMP. 	– instaluje i korzysta także z innych środowisk symulujących działanie serwera oraz omawia ich cechy.

			zastosowanie.		
23. Język zapytań, czyli poznajemy język SQL	– wyjaśnia, do czego służy język SQL.	– na podstawie podręcznika wymienia i krótko charakteryzuje podstawowe elementy składni SQL – DML, DDL, DCL, DQL; – na podstawie podręcznika lub innych publikacji omawia typy danych języka SQL.	– na podstawie podręcznika omawia podstawowe elementy składni SQL; – omawia przeznaczenie niektórych typów danych wykorzystywanych w MySQL.	– samodzielnie charakteryzuje podstawowe elementy składni SQL.	– omawia nieopisane w podręczniku elementy składni SQL.
24. Zadajemy pytania, czyli kwerendy w języku SQL	– na podstawie wiedzy z poprzednich lekcji omawia definicję i zastosowanie kwerendy; – na podstawie podręcznika uruchamia elementy XAMP niezbędne do korzystania z bazy danych; – przy pomocy nauczyciela i na podstawie opisu wykonuje ćwiczenia z podręcznika; – na podstawie podręcznika lub innych źródeł omawia różnicę między SQL a MySQL.	– na podstawie podręcznika omawia definicję MySQL jako otwarto-źródłowego systemu zarządzania bazami danych; – na podstawie podręcznika uruchamia phpMyAdmin i jego narzędzia z pakietu XAMP; – na podstawie podręcznika i z niewielką pomocą nauczyciela tworzy nową bazę i tabelę w MySQL; – na podstawie podręcznika i z niewielką pomocą nauczyciela tworzy zapytania w SQL.	– na podstawie podręcznika wykonuje ćwiczenie i wyjaśnia poszczególne etapy tworzenia zapytań; – omawia przykładowy kod tworzenia tabeli w SQL; – na podstawie tabeli omawia operatory wykorzystywane podczas wyszukiwania informacji w bazie.	– samodzielnie tworzy bazę i zapytania w języku SQL; – samodzielnie korzysta z MySQL.	– samodzielnie tworzy programy w SQL inne niż w przykładach podanych w podręczniku.
25. Więcej pytań, czyli jeszcze o kwerendach w języku SQL	– uruchamia w XAMP przykładowe programy SQL z podręcznika lub przedstawione przez nauczyciela.	– na podstawie podręcznika omawia składnię polecenia SELECT; – na podstawie podręcznika omawia działanie	– samodzielnie omawia składnię polecenia SELECT; – samodzielnie omawia działanie klauzuli WHERE	– wykorzystuje zapytanie SELECT i operatory logiczne podczas tworzenia zapytań; – łączy tabele w celu	– operuje MySQL i układa programy w SQL w przykładach innych niż poznane w trakcie lekcji.

		<p>klauzuli WHERE i operatorów logicznych;</p> <ul style="list-style-type: none"> – na podstawie podręcznika omawia efekt łączenia tabel w celu wyszukiwania informacji; – na podstawie podręcznika analizuje umieszczone w nim przykłady w SQL. 	<p>i operatorów logicznych;</p> <ul style="list-style-type: none"> – samodzielnie omawia efekt łączenia tabel w celu wyszukiwania informacji; – samodzielnie analizuje umieszczone w podręczniku przykłady w SQL; – sporadycznie korzystając ze źródeł wiedzy, wykonuje ćwiczenia z podręcznika. 	<p>przeszukania większej ich liczby;</p> <ul style="list-style-type: none"> – wyjaśnia, jak szukać informacji w tabelach niepołączonych ze sobą. 	
26. Podzapytania, czyli instrukcje modyfikujące dane w języku SQL	<ul style="list-style-type: none"> – na podstawie podręcznika wyjaśnia pojęcie <i>podzapytania</i>. 	<ul style="list-style-type: none"> – wyjaśnia pojęcie <i>podzapytania</i>; – na podstawie podręcznika omawia znaczenie podzapytań. 	<ul style="list-style-type: none"> – na podstawie podręcznika wykonuje opisane w nim ćwiczenie; – wyjaśnia istotę podzapytań wierszowych i skorelowanych; – wyjaśnia istotę podzapytań o charakterze tabeli tymczasowej; – omawia przykłady programów SQL realizujących podzapytania. 	<ul style="list-style-type: none"> – układa programy podzapytań w SQL do przykładów z podręcznika lub podanych przez nauczyciela. 	<ul style="list-style-type: none"> – samodzielnie układa programy podzapytań w SQL.
27. Modyfikacje, czyli zarządzanie bazą danych	<ul style="list-style-type: none"> – na podstawie podręcznika wyjaśnia działanie poleceń: UPDATE, ALTER TABLE, DELETE, DELETE FROM. 	<ul style="list-style-type: none"> – na podstawie podręcznika omawia składnię poleceń: UPDATE, ALTER TABLE, DELETE, DELETE FROM. 	<ul style="list-style-type: none"> – z pomocą nauczyciela lub na podstawie podręcznika modyfikuje elementy bazy danych. 	<ul style="list-style-type: none"> – samodzielnie, według zaleceń modyfikuje elementy (zmienia strukturę, usuwa rekordy, tabele) i strukturę bazy danych. 	<ul style="list-style-type: none"> – samodzielnie określa zakres modyfikacji bazy ze względu na zaistniałe potrzeby i modyfikuje elementy i strukturę bazy danych.
28. Bezpieczeństwo bazy, czyli tworzymy kopie zapasowe	<ul style="list-style-type: none"> – omawia zasady dbania o bezpieczeństwo baz danych; – podaje, czym jest kopia 	<ul style="list-style-type: none"> – na podstawie podręcznika omawia istotę najczęściej spotykanych ataków na bazę (pasywne 	<ul style="list-style-type: none"> – omawia na przykładach definicje ataków pasywnych i aktywnych; – omawia zasady, jakimi 	<ul style="list-style-type: none"> – tworzy kopię bezpieczeństwa, używając odpowiednich poleceń; – wyjaśnia pojęcie 	<ul style="list-style-type: none"> – planuje zabezpieczenie bazy danych.

	zapasowa.	i aktywne); – wymienia zasady, jakimi powinien się kierować administrator bazy w celu zabezpieczenia jej przed nieuprawnionym dostępem lub skasowaniem danych.	powinien się kierować administrator bazy w celu zabezpieczenia jej przed nieuprawnionym dostępem lub skasowaniem danych; – omawia działanie poleceń: CHECK TABLE, REPAIR TABLE, RESET MASTER, PURGE MASTER, SHOW BINLOG EVENTS; – omawia składnię poleceń do odzyskiwania danych z kopii bezpieczeństwa.	przyrostowej kopii bezpieczeństwa; – odzyskuje dane z kopii bezpieczeństwa.	
IV. Programowanie stron internetowych					
29. HTML, czyli przeglądarka interpretuje język programowania stron	– omawia przeznaczenie języka HTML; – omawia rolę znaczników w języku HTML.	– omawia rolę przeglądarki internetowej w kontekście języka HTML; – instaluje, uruchamia i korzysta ze wskazanego edytora (np. Notepad+) skonfigurowanego dla HTML5; – korzysta z opisów najczęściej używanych znaczników (np. z podręcznika).	– na podstawie podręcznika lub innych wiarygodnych źródeł konfiguruje edytor do układania programów w HTML5; – na podstawie podręcznika lub innych wiarygodnych źródeł układa proste programy stron, np. wyświetlające tekst z zachowaniem prawidłowej struktury programu; – samodzielnie objaśnia rolę CSS w projektowaniu wyglądu strony.	– dobiera odpowiedni edytor do edycji programów w HTML5 i uzasadnia swój wybór; – konfiguruje edytor; – tworzy proste strony internetowe zawierające sformatowany tekst.	– tworzy rozbudowane strony internetowe w języku HTML5.
30. Budujemy stronę, czyli tabele, listy i inne elementy dobrej strony	– omawia znaczenie list w programie HTML5; – uruchamia w przeglądarce gotowy	– na podstawie podręcznika uruchamia programy z przykładów; – na podstawie przykła-	– na podstawie podręcznika lub innych wiarygodnych źródeł tworzy w kodzie HTML listy	– samodzielnie tworzy w kodzie HTML listy uporządkowane i nieuporządkowane oraz	– tworzy rozbudowane strony internetowe w języku HTML5.

	program strony w HTML5 (np. z przykładu z podręcznika).	dów omawia rolę list, definicji i tabel; – tłumaczy istotę i przeznaczenie hipertekstu.	uporządkowane i nieuporządkowane oraz omawia ich znaczenie; – na podstawie podręcznika lub innych wiarygodnych źródeł tworzy listy definicji z dodanymi nagłówkami; – na podstawie podręcznika lub innych wiarygodnych źródeł tworzy tabele w HTML.	omawia ich znaczenie; – samodzielnie tworzy listy definicji z dodanymi nagłówkami; – samodzielnie tworzy tabele w HTML; – planuje użycie odpowiednich konstrukcji do projektowanej strony; – używa atrybutów w konstrukcji hipertekstu.	
31. Tabele i grafika, czyli kolejne składowe stron internetowych w HTML	– wymienia zasady przygotowania grafiki do publikacji na stronie z uwzględnieniem wymiarów i rozdzielczości.	– na podstawie podręcznika wstawia przygotowaną przez nauczyciela grafikę do gotowego kodu strony; – na podstawie podręcznika tłumaczy znaczenie wymiarów i skalowania dla szybkości wczytywania strony.	– określa parametry zdjęcia, biorąc pod uwagę jego miejsce na stronie; – zmienia parametry zdjęcia przeznaczonego na stronę za pomocą prostych narzędzi systemu Windows.	– planuje miejsce na grafikę; – przygotowuje grafikę do publikacji na stronie; – umieszcza zdjęcia w planowanym miejscu i formacie na stronie w HTML; – tworzy odnośnik z elementu graficznego umieszczonego na stronie w HTML.	– tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem elementów graficznych.
32. Składnia stylów, czyli jak CSS pomaga w programowaniu wyglądu strony	– objaśnia rolę CSS w kreowaniu wyglądu strony.	– objaśnia istotę pliku stylów i jego połączenie z HTML.	– na podstawie podręcznika lub innych wiarygodnych źródeł omawia zasady łączenia CSS z HTML; – na podstawie podręcznika lub innych wiarygodnych źródeł układa proste pliki stylów CSS, formatujące sposób wyświetlania tekstu;	– samodzielnie omawia zasady łączenia CSS z HTML; – samodzielnie układa proste pliki stylów CSS, formatujące sposób wyświetlania tekstu; – samodzielnie wyjaśnia znaczenie nagłówków i bloków w stosowaniu CSS;	– tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem CSS.

			<ul style="list-style-type: none"> – na podstawie podręcznika lub innych wiarygodnych źródeł wyjaśnia znaczenie nagłówków i bloków w stosowaniu CSS; – na podstawie podręcznika lub innych wiarygodnych źródeł wyjaśnia znaczenie sekcji i selektora. 	<ul style="list-style-type: none"> – samodzielnie wyjaśnia znaczenie sekcji i selektora; – projektuje użycie CSS do formatowania stylu tekstu na stronie. 	
33. Pliki stylów, czyli CSS w akcji	<ul style="list-style-type: none"> – objaśnia działanie i znaczenie odsyłaczy na stronie w HTML; – omawia rolę menu na stronie. 	<ul style="list-style-type: none"> – na podstawie podręcznika lub innych wiarygodnych źródeł omawia znaczenie selektorów i ich atrybutów w pliku CSS; – na podstawie podręcznika lub innych wiarygodnych źródeł omawia pojęcie <i>walidacja</i>. 	<ul style="list-style-type: none"> – na podstawie podręcznika lub innych wiarygodnych źródeł tworzy pliki HTML i CSS z menu z przyciskami, wykorzystując przykłady; – na podstawie podręcznika lub innych wiarygodnych źródeł używa pseudoklas z przykładu do określenia interakcji klawisza z kursorem; – na podstawie podręcznika lub innych wiarygodnych źródeł rozmieszcza elementy strony, tworząc jej prosty layout. 	<ul style="list-style-type: none"> – samodzielnie tworzy pliki HTML i CSS z menu z przyciskami, wykorzystując przykłady; – samodzielnie używa pseudoklas z przykładu do określenia interakcji klawisza z kursorem; – samodzielnie rozmieszcza elementy strony, tworząc jej prosty layout; – zmienia wygląd i atrybuty menu z przykładu; – zmienia położenie elementów strony; – stosuje różną orientację elementów menu; – przeprowadza walidację strony. 	<ul style="list-style-type: none"> – tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem CSS zawierające menu i efekty interakcji z kursorem.
34. Treści mogą się zmieniać, czyli elementy dynamiczne na stronie internetowej	<ul style="list-style-type: none"> – wskazuje elementy istniejących stron, które zmieniają się dynamicznie; – na podstawie podręcz- 	<ul style="list-style-type: none"> – omawia sposoby wklejania w kod strony gotowych odnośników do elementów dynamicznych; 	<ul style="list-style-type: none"> – tworzy element dynamiczny zewnętrzny za pomocą darmowego edytora aplikacji (np. learningapps.com); 	<ul style="list-style-type: none"> – planuje umieszczenie na stronie widżetów i innych elementów dynamicznych; – dokonuje prawidłowego 	<ul style="list-style-type: none"> – tworzy strony z własnymi elementami dynamicznymi.

	nika wskazuje różnice między statyczną a dynamiczną stroną internetową.	– na podstawie podręcznika umieszcza w kodzie strony elementy dynamiczne.	– pobiera i umieszcza na stronie kod uruchamiający widżety lub aplikacje internetowe z learningapps.com.	wyboru widżetów; – projektuje elementy dynamiczne (np. krzyżówki) w learningapps.com.	
35. Widoczna w internecie, czyli jak opublikować stronę	– omawia funkcje domeny internetowej; – wskazuje przynajmniej jedną firmę (stronę internetową) zajmującą się rejestracją domen.	– omawia proces wczytywania strony internetowej do przeglądarki i rolę domeny; – omawia znaczenie niektórych domen (np. .com, .pl, .edu); – podaje, jakie usługi musi uruchomić lub zamówić, by strona była widoczna w sieci.	– na podstawie podręcznika lub innych wiarygodnych źródeł opisuje na przykładzie proces rezerwacji domeny; – na podstawie podręcznika lub innych wiarygodnych źródeł zakłada konta na darmowym serwerze z usługą hostingową (np. cba.pl); – na podstawie podręcznika lub innych wiarygodnych źródeł wysyła pliki strony do serwera www.	– samodzielnie opisuje na przykładzie proces rezerwacji domeny; – samodzielnie zakłada konta na darmowym serwerze z usługą hostingową (np. cba.pl); – samodzielnie wysyła pliki strony do serwera www; – sprawdza zajętość domen, którymi jest zainteresowany; – korzysta z klienta FTP nie tylko do przesyłania plików strony.	– podaje, jak zarządzać domeną i usługą hostingową na przykładzie swojego konta.
36. CMS, czyli system zarządzania treścią strony internetowej	– omawia definicję CMS-u; – podaje, do czego służy CMS i wymienia kilka najpopularniejszych (w tym Joomla i WordPress).	– omawia proces przygotowań do instalacji CMS-u; – na podstawie podręcznika lub innych wiarygodnych źródeł instaluje środowisko serwerowe (np. Laragon), a w nim CMS (np. WordPress); – na podstawie podręcznika lub innych wiarygodnych źródeł loguje się do panelu sterowania CMS-em (np.	– samodzielnie instaluje środowisko serwerowe (np. Laragon), a w nim CMS (np. WordPress); – samodzielnie loguje się do panelu sterowania CMS-em (np. WordPress).	– dobiera odpowiedni do tematu strony CMS i uzasadnia swój wybór; – wyjaśnia konieczność instalacji bazy danych dla CMS-a; – używa wirtualnego środowiska serwera do testowania działania różnych stron internetowych.	– buduje ciekawe strony internetowe za pomocą różnych CMS-ów.

		WordPress).			
37. Panel i skórki, czyli tworzymy stronę w CMS	<ul style="list-style-type: none"> – wyjaśnia znaczenie szablonu (skórki) w systemie CMS; – przegląda oferowane w sieci skórki dla WordPressa. 	<ul style="list-style-type: none"> – odróżnia motywy od szablonów; – omawia znaczenie i funkcję motywów w szablonach; – na podstawie podręcznika wypełnia treścią stronę zbudowaną na bazie szablonu. 	<ul style="list-style-type: none"> – na podstawie podręcznika lub innych wiarygodnych źródeł instaluje w CMS-ie wskazany szablon; – na podstawie podręcznika lub innych wiarygodnych źródeł dobiera motyw i zmienia go w razie potrzeby; – na podstawie podręcznika lub innych wiarygodnych źródeł modyfikuje motyw skórki; – na podstawie podręcznika lub innych wiarygodnych źródeł wypełnia treścią stronę, tworząc nowe wpisy; – na podstawie podręcznika lub innych wiarygodnych źródeł umieszcza na stronie elementy graficzne. 	<ul style="list-style-type: none"> – samodzielnie instaluje w CMS-ie wskazany szablon; – samodzielnie dobiera motyw i zmienia go w razie potrzeby; – samodzielnie modyfikuje motyw skórki; – samodzielnie wypełnia treścią stronę, tworząc nowe wpisy; – samodzielnie umieszcza na stronie elementy graficzne; – trafnie i z uzasadnieniem dobiera szablon i motyw do treści przyszłej strony; – administruje stroną, wykorzystując panel sterowania; – dodaje i usuwa użytkowników CMS-a i świadomie nadaje im uprawnienia; – umieszcza na stronie filmy i inne elementy multimedialne. 	<ul style="list-style-type: none"> – planuje i realizuje własne projekty stron na bazie CMS.
38. Szybkie i łatwe, czy programy do tworzenia stron internetowych	<ul style="list-style-type: none"> – podaje, jak zapisać tekstowy dokument Word w formacie HTML; – wymienia kilka systemów darmowych blogów. 	<ul style="list-style-type: none"> – na podstawie podręcznika lub innych wiarygodnych źródeł eksportuje do formatu HTML dokumenty Word i Excel; 	<ul style="list-style-type: none"> – samodzielnie eksportuje do formatu HTML dokumenty Word i Excel; – samodzielnie zakłada konto w systemie darmowych blogów 	<ul style="list-style-type: none"> – prawidłowo i estetycznie rozmieszcza elementy graficzne, wzory itp. w eksportowanym dokumencie; – testuje w środowisku 	<ul style="list-style-type: none"> – prowadzi własnego bloga na ciekawy temat.

		<ul style="list-style-type: none"> – na podstawie podręcznika lub innych wiarygodnych źródeł zakłada konto w systemie darmowych blogów wskazanym przez nauczyciela. 	<ul style="list-style-type: none"> wskazanym przez nauczyciela; – formatuje dokument Word I Excel w taki sposób, by po eksporcie do HTML powstała estetyczna i funkcjonalna strona; – wypełnia treścią blog utworzony w darmowym systemie blogów. 	<ul style="list-style-type: none"> serwerowym (np. Lagon) strony utworzone podczas eksportu dokumentów do HTML, zachowując układ folderów. 	
39. Pierwsze skrypty, czyli poznajemy PHP	<ul style="list-style-type: none"> – omawia specyfikę języka PHP (skrypt umieszczony w HTML i na odwrót); – uruchamia środowisko Notepad++. 	<ul style="list-style-type: none"> – na podstawie podręcznika omawia umieszczenie skryptów PHP w dokumencie HTML; – uruchamia gotowe przykłady z podręcznika lub podane przez nauczyciela. 	<ul style="list-style-type: none"> – omawia rolę funkcji w języku PHP, w tym: print(), echo(), printf(); – omawia działanie przykładowego skryptu, np. z podręcznika. 	<ul style="list-style-type: none"> – układa prosty skrypt wyświetlający tekst; – umieszcza skrypt na serwerze, uruchamia go w dokumencie HTML i testuje w przeglądarce. 	<ul style="list-style-type: none"> – tworzy proste skrypty PHP.
40. Stałe i zmienne, czyli typy danych w języku PHP	<ul style="list-style-type: none"> – na podstawie podręcznika wyjaśnia, czym są stałe i zmienne w języku PHP. 	<ul style="list-style-type: none"> – wymienia znaki, jakie mogą być użyte w nazwach zmiennych i stałych w języku PHP; – na podstawie podręcznika omawia typy zmiennych używanych w języku PHP; – na podstawie podręcznika analizuje przykładowy skrypt PHP, w którym zastosowano zmienne i stałe; – wymienia operatorów w języku PHP i omawia ich działanie. 	<ul style="list-style-type: none"> – analizuje przykładowy skrypt ze wskazaniem funkcji, jakie pełnią zmienne i stałe; – omawia sposób definiowania zmiennych i stałych w PHP; – prawidłowo używa operatorów. 	<ul style="list-style-type: none"> – układa nieskomplikowane skrypty według specyfikacji, w których używa zmiennych i stałych. 	<ul style="list-style-type: none"> – układa na podstawie własnych specyfikacji i pomysłów proste skrypty PHP.

41. Instrukcje sterujące, czyli praktyczne wykorzystanie języka PHP	– wyjaśnia ogólnie, na czym polega działanie instrukcji warunkowych i wyboru.	– omawia działanie instrukcji warunkowych i wyboru w PHP przez analogię do podobnych instrukcji języka C++ /Python; – na podstawie opisu omawia działanie przykładowych skryptów PHP zawierających instrukcje warunkowe i wyboru.	– samodzielnie omawia działanie przykładowych skryptów PHP zawierających instrukcje warunkowe i wyboru, uruchamia je i testuje.	– układa według specyfikacji skryptu PHP zawierające instrukcje warunkowe i wyboru.	– układa na podstawie własnych specyfikacji i pomysłów skryptu PHP zawierające instrukcje warunkowe i wyboru.
42. Pętle, czyli jeszcze więcej możliwości wykorzystania języka PHP	– wyjaśnia ogólnie, na czym polega działanie pętli w algorytmie i w języku programowania.	– omawia działanie pętli: for, while, do while w PHP przez analogię do podobnych instrukcji języka C++ / Python; – wyjaśnia różnice między instrukcjami pętli.	– omawia działanie przykładowych skryptów PHP zawierających pętle; – analizuje przykłady skryptów zawierających pętle.	– układa według specyfikacji skryptu PHP zawierające pętle.	– układa na podstawie własnych specyfikacji i pomysłów skryptu PHP zawierające pętle.
43. Funkcje w języku PHP, czyli jak wzbogacić naszą stronę	– na podstawie podręcznika objaśnia, na czym polega użycie funkcji w programie komputerowym.	– na podstawie podręcznika wyjaśnia różnice między funkcjami wbudowanymi a własnymi; – na podstawie podręcznika wymienia nazwy i określa rolę jaką odgrywają niektóre funkcje wbudowane.	– samodzielnie wyjaśnia różnice między funkcjami wbudowanymi a własnymi; – samodzielnie wymienia nazwy i określa rolę, jaką odgrywają niektóre funkcje wbudowane; – analizuje przykłady skryptów złożonych w PHP z użyciem funkcji; – testuje przykłady skryptów PHP zawierających funkcje.	– omawia rolę niektórych funkcji wbudowanych; – tworzy skrypt PHP według specyfikacji z użyciem funkcji wbudowanych i testuje ich działanie.	– układa skrypty PHP zawierające funkcje własne.
44. Tablice, czyli jak sortować dane w języku	– omawia na realnym przykładzie istotę tablic.	– omawia różnice między zmiennymi tablicowymi	– samodzielnie analizuje działanie przykładowych	– układa według specyfikacji skryptu	– układa według własnych specyfikacji i pomysłów

PHP		w języku C++/Python a PHP; – omawia różnice między tablicami jedno- a wielowymiarowymi; – uruchamia przykładowe skrypty PHP zawierające tablice i na podstawie podręcznika omawia ich działanie.	skryptów PHP zawierających tablice.	zawierające tablice.	skrypty PHP zawierające tablice.
45. Pobieranie i przetwarzanie danych, czyli obsługa formularza w języku PHP	– omawia istotę i cele stosowania formularzy na stronach internetowych.	– na podstawie opisu omawia sedno metod POST i GET stosowanych w budowaniu formularzy; – uruchamia i testuje przykładowe skrypty z formularzami.	– samodzielnie analizuje działanie przykładowych skryptów PHP zawierających formularze; – proponuje zmiany w przykładowych formularzach w związku ze zmianą specyfikacji skryptu.	– układa według specyfikacji skrypty tworzące na stronie formularze, testuje je i modyfikuje; – używa w swoich skryptach obu metod i uzasadnia ich wybór.	– układa według własnych specyfikacji i pomysłów skrypty PHP zawierające zaprojektowane przez siebie formularze.
V. Algorytmy i programowanie					
46. Powtarzanie w pętli i wywołanie siebie, czyli iteracja i rekurencja w algorytmach	– omawia na realnych przykładach różnicę między rekurencją a iteracją.	– na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykładowe algorytmy iteracyjne i rekurencyjne; – na podstawie podręcznika lub innych wiarygodnych źródeł rozpoznaje procesy rekurencyjne i iteracyjne.	– samodzielnie analizuje przykładowe algorytmy iteracyjne i rekurencyjne; – samodzielnie rozpoznaje procesy rekurencyjne i iteracyjne; – wskazuje w przykładowych algorytmach miejsca, które decydują o iteracyjności lub rekurencyjności opisywanego procesu; – analizuje przykładowy program.	– układa algorytmy z podejściem iteracyjnym i rekurencyjnym; – układa programy na podstawie tych algorytmów; – wskazuje instrukcje, które decydują o iteracyjności lub rekurencyjności podejścia do realizacji algorytmu.	– rozwiązuje problemy obiema metodami i ocenia ich skuteczność.
47. Sortowanie	– na podstawie podręcz-	– na podstawie podręcz-	– samodzielnie omawia	– układa algorytm	– układa program

bąbelkowe, czyli każda liczba jest mniejsza od maksymalnej lub jej równa	nika lub innych wiarygodnych źródeł omawia budowę tablicy jednowymiarowej; – na podstawie podręcznika lub innych wiarygodnych źródeł omawia istotę sortowania bąbelkowego.	nika lub innych wiarygodnych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu blokowego; – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykład sprawdzający poprawność działania algorytmu.	istotę metody sortowania bąbelkowego; – samodzielnie omawia działanie przykładowego algorytmu opartego o metodę sortowania bąbelkowego; – samodzielnie sprawdza działanie algorytmu na przykładach.	sortowania bąbelkowego; – weryfikuje poprawność działania programu na przykładach.	sortujący metodą bąbelkową w innym języku niż C++/ Python (np. Java).
48. Przez wstawianie, czyli jeszcze o porządkowaniu liczb	– na podstawie podręcznika lub innych wiarygodnych źródeł omawia istotę sortowania przez wstawianie.	– na podstawie podręcznika lub innych wiarygodnych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu przez wstawianie; – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykład sprawdzający poprawność działania algorytmu.	– samodzielnie omawiana przykładzie istotę metody sortowania przez wstawianie; – samodzielnie omawia działanie przykładowego algorytmu opartego o metodę sortowania przez wstawianie; – samodzielnie sprawdza działanie algorytmu na przykładach.	– układa algorytm sortowania przez wstawianie w postaci listy kroków i schematu blokowego; – weryfikuje poprawność działania programu na przykładach.	– układa program sortujący metodą przez wstawianie w innym języku niż C++ / Python (np. Java).
49. Komputer porządkuje, czyli układamy programy sortujące	– na podstawie podręcznika lub innych wiarygodnych źródeł weryfikuje działanie przykładowych programów (np. z podręcznika).	– na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykładowe programy i wskazuje miejsca, w których są wykonywane działania najważniejsze dla metody.	– na podstawie podręcznika lub innych wiarygodnych źródeł układa programy sortujące; – na podstawie podręcznika uzasadnia użycie danych instrukcji w przykładowych programach.	– samodzielnie układa programy sortujące według obu metod; – samodzielnie weryfikuje poprawność działania tych programów; – wskazuje instrukcje i rozkazy realizujące istotę metody.	– proponuje inne od przykładowych rozwiązania programowe; – układa programy realizujące algorytmy sortowania bąbelkowego i przez wstawianie według własnego pomysłu.

50. Fibonacci i jego wzór, czyli generujemy kolejne liczby ciągu	<ul style="list-style-type: none"> – wie, kim był i kiedy żył Fibonacci; – zna jego zasługi dla rozwoju cywilizacji. 	<ul style="list-style-type: none"> – na podstawie podręcznika lub innych wiarygodnych źródeł omawia na przykładzie istotę ciągu Fibonacciego; – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje działanie przykładowego algorytmu obliczającego kolejne elementy ciągu, zapisanego w postaci schematu blokowego. 	<ul style="list-style-type: none"> – samodzielnie omawia na przykładzie istotę ciągu Fibonacciego; – samodzielnie analizuje działanie przykładowego algorytmu obliczającego kolejne elementy ciągu, zapisanego w postaci schematu blokowego; – układa algorytm obliczający określoną liczbę liczb ciągu Fibonacciego; – analizuje program w języku C++ / Python ułożony według przykładowego algorytmu. 	<ul style="list-style-type: none"> – układa program w języku C++ / Python obliczający n kolejnych elementów ciągu Fibonacciego; – testuje poprawność działania swojego programu na przykładzie. 	<ul style="list-style-type: none"> – proponuje rozwiązanie problemu obliczania n-tego elementu ciągu Fibonacciego.
51. Szukamy lidera, czyli porządkowanie zbioru metodą binarnego wyszukiwania	<ul style="list-style-type: none"> – omawia, na czym polega metoda dziel i zwyciężaj; – rozumie, na czym polega umieszczanie danych w tablicy. 	<ul style="list-style-type: none"> – na podstawie podręcznika opisuje zastosowanie wyszukiwania binarnego w sortowaniu. 	<ul style="list-style-type: none"> – omawia na przykładzie działanie algorytmu sortowania zbioru z wykorzystaniem wyszukiwania binarnego. 	<ul style="list-style-type: none"> – testuje algorytm sortowania zbioru z wykorzystaniem wyszukiwania binarnego na dowolnych danych; – układa program na podstawie algorytmu. 	<ul style="list-style-type: none"> – stosuje algorytm dla tablicy wielowymiarowej.
52. Przecinając oś, czyli wyznaczenie miejsc zerowych funkcji metodą połowienia	<ul style="list-style-type: none"> – podaje definicję miejsca zerowego funkcji i wskazuje je na wykresie. 	<ul style="list-style-type: none"> – na podstawie podręcznika omawia metodę połowienia. 	<ul style="list-style-type: none"> – analizuje algorytm metody połowienia z podręcznika. 	<ul style="list-style-type: none"> – na podstawie opisu metody opracowuje algorytm metody połowienia i na jego podstawie układa program w języku C++/Python; – testuje algorytm i program. 	<ul style="list-style-type: none"> – omawia inne metody znajdowania miejsca zerowego, np. metodę analityczną.
53. Prawie robi różnicę,	<ul style="list-style-type: none"> – podaje, czym jest 	<ul style="list-style-type: none"> – na podstawie podręcz- 	<ul style="list-style-type: none"> – omawia podstawy 	<ul style="list-style-type: none"> – na podstawie opisu 	<ul style="list-style-type: none"> – omawia inne metody

czyli obliczanie przybliżonej wartości pierwiastka kwadratowego	przybliżenie wartości pierwiastka i zna konsekwencje jego stosowania.	nika omawia metodę Newtona-Robsona.	matematyczne metody Newtona-Robsona; – analizuje omówiony w podręczniku algorytm i program metody Newtona-Robsona.	metody samodzielnie opracowuje algorytm metody Newtona-Robsona; – układa program znajdujący przybliżoną wartość pierwiastka kwadratowego i go testuje.	obliczania wartości pierwiastka kwadratowego.
54. Szybkie potęgowanie liczb w wersji iteracyjnej, czyli jak obliczyć potęgę o wykładniku naturalnym	– podaje właściwości potęgi liczby 0; – własnymi słowami opisuje, czym jest potęgowanie liczb i jak się je wykonuje; – uruchamia program z podręcznika i testuje jego działanie; – podaje, na czym polega iteracyjne podejście do rozwiązywania problemów.	– na podstawie podręcznika omawia algorytm iteracyjny metody szybkiej potęgowania i porównuje ją z metodą kolejnego mnożenia przez podstawę potęgi.	– samodzielnie omawia metodę szybkiego potęgowania w wersji iteracyjnej; – analizuje algorytm i program w języku C++/Python metody szybkiego potęgowania.	– omawia na przykładzie zalety metody szybkiego potęgowania; – układa algorytm metody szybkiego potęgowania i na jego podstawie program w języku C++/Python w wersji iteracyjnej.	– porównuje podejście iteracyjne i rekurencyjne do obliczania potęgi metodą szybką.
55. Szybkie potęgowanie liczb w wersji rekurencyjnej, czyli jak obliczyć potęgę o wykładniku naturalnym	– uruchamia program z podręcznika i testuje jego działanie; – podaje, na czym polega rekurencyjne podejście do rozwiązywania problemów.	– na podstawie podręcznika omawia rekurencyjne podejście do metody potęgowania szybkiego.	– samodzielnie omawia metodę szybkiego potęgowania w wersji rekurencyjnej; – analizuje algorytm i program w języku C++/Python lub Java dla metody szybkiego potęgowania rekurencyjnego.	– układa program metody szybkiego potęgowania i na jego podstawie program w języku C++/Python lub Java w wersji rekurencyjnej.	– porównuje podejście iteracyjne i rekurencyjne do obliczania potęgi metodą szybką.
56. Suma binarna, czyli dodawanie w systemie dwójkowym	– omawia podstawy systemu dwójkowego; – dodaje do siebie	– zamienia reprezentacje liczb binarnych na dziesiętne i odwrotnie.	– układa algorytm dodawania binarnego i go testuje.	– układa program w języku C++/Python realizujący algorytm	– układa program na podstawie algorytmu także w innym języku (np.

	pisemnie niewielkie liczby binarne.			dodawania liczb binarnych i go testuje.	Java).
57. Różnica binarna, czyli odejmowanie w systemie dwójkowym	– odejmuje pisemnie niewielkie liczby binarne.	– opisuje metodę odejmowania liczb binarnych i tłumaczy rolę pożyczki na przykładzie.	– układa algorytm odejmowania binarnego i go testuje.	– układa program w języku C++ / Python realizujący algorytm odejmowania liczb binarnych i go testuje.	– układa program na podstawie algorytmu także w innym języku (np. Java).
58. Iloczyn binarny, czyli mnożenie w systemie dwójkowym	– mnoży pisemnie niewielkie liczby binarne.	– opisuje własnymi słowami na przykładzie metodę mnożenia liczb binarnych.	– układa algorytm mnożenia binarnego i go testuje.	– układa program w języku C++ / Python realizujący algorytm mnożenia liczb binarnych i go testuje.	– układa program na podstawie algorytmu także w innym języku (np. Java).
59. Iloraz binarny, czyli dzielenie w systemie dwójkowym	– dzieli pisemnie niewielkie liczby binarne.	– opisuje własnymi słowami na przykładzie metodę dzielenia liczb binarnych.	– układa algorytm dzielenia binarnego i go testuje.	– układa program w języku C++ / Python realizujący algorytm dzielenia liczb binarnych i go testuje.	– układa program na podstawie algorytmu także w innym języku (np. Java).
60. Metoda Monte Carlo, czyli jak obliczyć przybliżoną wartość liczby π	– omawia znaczenie liczby π na przykładach (np. obliczania obwodu koła).	– opisuje własnymi słowami na przykładzie, korzystając z rysunku pomocniczego, metodę Monte Carlo obliczania kolejnych pozycji liczby π ; – zna wzór na liczbę π z metody Monte Carlo.	– układa algorytm w dowolnym zapisie obliczający kolejne pozycje liczby π metodą Monte Carlo.	– układa program w języku C++ / Python realizujący algorytm Monet Carlo.	– układa program na podstawie algorytmu także w innym języku (np. Java).