

## Część 1,2, 3 – Informatyka rozszerzona

Egzamin teoretyczny i praktyczny: zadania do wykonania z zakresu: analiza algorytmów, tworzenie algorytmów, test z ogólnej wiedzy informatycznej, symulacja, przetwarzanie i tworzenie informacji, bazy danych.

Link do ćwiczeń: <https://cke.gov.pl/egzamin-maturalny/egzamin-maturalny-w-formule-2015/materialy-dodatkowe/materialy-dla-uczniow-i-nauczycieli/zbiory-zadan/>

### Zadania do wykonania:

1. Przygotuj fotomontaż na dowolny temat w programie Gimp. Pamiętaj o tworzeniu warstw. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
2. Przygotuj obrazek na dowolny temat z wykorzystaniem grafiki wektorowej w programie Inkscape. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
3. Przygotuj stronę internetową w języku HTML na dowolny temat. Ma ona zawierać: formatowanie tekstu, kolory, tło strony, odnośniki do innej własnej podstrony oraz innej strony w Internecie, tabelkę, obrazki, listę wypunktowaną lub numerowaną, kaskadowe arkusze stylów CSS. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
4. Przygotuj program w języku C++ wyszukujący największą liczbę w ciągu liczb. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.

Programy nie mogą być identyczne jak w podręczniku!

Do programowania w języku C++ pobierz program Dev C++ Lub Code Blocks.

**Prace należy przestać co najmniej tydzień przed egzaminem z pośrednictwem platformy Teams. Zostanie tam utworzone specjalne zadanie**

### Plan wynikowy (propozycja) cz.1

Lp.	Temat	Osiągnięcia uczniów	
		Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
1	Systemy operacyjne w środowisku sieciowym	<ul style="list-style-type: none"><li>– wymienia systemy operacyjne oraz ich zadania</li><li>– rozumie kwestie związane z bezpieczeństwem w przestrzeni cyfrowej</li><li>– zna zasady tworzenia mocnych haseł</li><li>– rozumie potrzebę stosowania kont użytkownika w systemie operacyjnym</li><li>– stosuje zasady bezpieczeństwa obowiązujące w pracowni komputerowej</li><li>– instaluje i aktualizuje oprogramowanie</li></ul>	<ul style="list-style-type: none"><li>– wyjaśnia, w jakim trybie (jądra czy użytkownika) powinien pracować program sterownika urządzenia w większości systemów operacyjnych</li><li>– zna procedurę wykonania kopii zapasowej dla systemu operacyjnego i wszystkich danych użytkownika komputera</li><li>– tworzy nośnik awaryjny uruchamiający komputer, gdy zainstalowany na nim system operacyjny nie działa prawidłowo</li><li>– wie, czym są fragmentacja i defragmentacja dysku</li></ul>

		<ul style="list-style-type: none"> <li>– zakłada i usuwa konto w środowisku aplikacji Google</li> <li>– pracuje w środowisku sieciowym</li> </ul>	<ul style="list-style-type: none"> <li>– sprawdza poziom fragmentacji dysku komputera i ocenia, czy wymagana jest jego defragmentacja</li> <li>– wie, jaka jest rola systemu plików jako części systemu operacyjnego</li> <li>– sprawdza, jaki system plików został przypisany do danego dysku</li> <li>– wie, w jaki sposób uruchomić tryb awaryjny w systemie Windows (od wersji Windows 7), zna poszczególne opcje dostępne dla trybu awaryjnego i wie, do czego służą</li> <li>– zna polecenia w trybie tekstowym Windows i posługuje się nimi</li> </ul>
2	Nowe technologie i oprogramowanie	<ul style="list-style-type: none"> <li>– rozumie pojęcia takie jak: sztuczna inteligencja, chmura obliczeniowa i posługuje się nimi</li> <li>– wymienia zastosowania automatyki i robotyki w życiu codziennym</li> <li>– wskazuje zalety i sposoby wykorzystania druku 3D</li> </ul>	<ul style="list-style-type: none"> <li>– proponuje własne, dotąd nieznanne, sposoby na wykorzystanie nowych technologii</li> <li>– wyjaśnia zastosowanie nowych rozwiązań technologicznych w różnych dziedzinach życia</li> <li>– posługując się darmowymi aplikacjami do tworzenia rozszerzonej rzeczywistości, tworzy filmy, artykuły i infografiki</li> </ul>
3	Projektowane zestawu komputerowego	<ul style="list-style-type: none"> <li>– projektuje rozbudowę nowego zestawu komputerowego oraz oprogramowania</li> </ul>	<ul style="list-style-type: none"> <li>– projektując zestaw komputerowy, kieruje się podanymi kryteriami (cena, przeznaczenie) i uzasadnia swój wybór</li> </ul>
4	Sieci komputerowe – budowa i usługi	<ul style="list-style-type: none"> <li>– rozumie pojęcia: sieć, protokół sieciowy, topologia sieci</li> <li>– rozróżnia i poprawnie nazywa sieci komputerowe ze względu na ich zasięg i topologię</li> <li>– opisuje budowę sieci lokalnej i sieci Internet</li> <li>– rozumie pojęcia takie jak adres IP, host, router, maska podsieci, brama, DNS oraz omawia zasadę adresowania urządzeń w sieci Internet</li> <li>– wymienia różne usługi internetowe</li> <li>– potrafi opisać warstwowy model działania Internetu oraz wymienić zadania poszczególnych warstw</li> </ul>	<p>testuje prędkość połączenia z siecią Internet na wybranym urządzeniu i interpretuje otrzymany wynik</p> <p>zna polecenia tekstowe służące do diagnostyki sieci i korzysta z nich</p> <p>oblicza liczbę możliwych do zaadresowania hostów na podstawie adresów IP i masek podsieci</p> <p>rozumie, czym jest model warstwowy TCP/IP</p> <p>wyjaśnia sposoby działania usługi NAT</p>

5	Model warstwowy sieci komputerowych	<ul style="list-style-type: none"> <li>– opisuje warstwowy model sieci komputerowej oraz model sieci Internet</li> <li>– opisuje podstawowe funkcje urządzeń i protokoły stosowane w przepływie informacji i w zarządzaniu siecią</li> </ul>	
6	Protokoły IPv4 i IPv6	<ul style="list-style-type: none"> <li>– konfiguruje przykładową lokalną sieć komputerową oraz bezprzewodowy dostęp do sieci Internet</li> </ul>	
7	Konfiguracja sieci komputerowej	<ul style="list-style-type: none"> <li>– wyjaśnia, od czego zależą sprawne funkcjonowanie sieci komputerowej oraz szybki dostęp do jej usług i zasobów</li> <li>– określa parametry wybranego osprzętu sieciowego, szerokość pasma oraz możliwości serwera</li> </ul>	
8	E-usługi	<ul style="list-style-type: none"> <li>– poprawnie definiuje pojęcie e-usługi</li> <li>– wymienia różne zastosowania usług elektronicznych</li> <li>– charakteryzuje problemy oraz wymienia zalety związane z wykorzystaniem e-usług</li> </ul>	<ul style="list-style-type: none"> <li>– opisuje zabezpieczenia wybranych e-usług (w tym systemu ePUAP)</li> <li>– określa możliwości rozwoju dla wybranych e-usług, z których korzysta</li> <li>– wymienia narzędzia dostępne w sieci, które umożliwiają utworzenie wybranych e-usług</li> </ul>
9	Korzystanie z e-zasobów i współpraca zdalna	<ul style="list-style-type: none"> <li>– rozumie pojęcie informacji</li> <li>– korzysta z zasobów internetowych, wyszukując potrzebne informacje</li> <li>– wymienia etapy rozwoju technologii komputerowych</li> <li>– korzysta z różnych wyszukiwarek internetowych</li> <li>– wykorzystuje zasoby sieciowe do poszerzania własnej wiedzy (e-learning)</li> <li>– zna podstawy prawa autorskiego</li> <li>– stosuje zasady netykiety i korzysta z niej w komunikacji zdalnej</li> </ul>	<ul style="list-style-type: none"> <li>– wie, czym jest pozycjonowanie serwisów internetowych</li> <li>– wyjaśnia sposób tworzenia wybranych e-zasobów oraz wskazuje zalety i wady poszczególnych rozwiązań</li> <li>– zna i stosuje zapisy ustawy o prawie autorskim i prawach pokrewnych</li> </ul>
10	Rozbudowane dokumenty tekstowe	<ul style="list-style-type: none"> <li>– korzysta z programu Microsoft Word</li> <li>– stosuje style nagłówkowe (korzysta z gotowych, tworzy własne i modyfikuje je)</li> <li>– stosuje numeracje i wypunktowania, dostosowując ich styl</li> </ul>	<ul style="list-style-type: none"> <li>– tworzy styl według wzoru</li> <li>– pracuje nad dokumentem wspólnie z innymi osobami w trybie śledzenia zmian</li> </ul>

		<ul style="list-style-type: none"> <li>– formatuje elementy dokumentu odpowiedzialne za automatyczne spisy (treści, tabel, ilustracji)</li> <li>– wstawia w dokumencie spisy treści, tabel, ilustracji</li> <li>– poprawnie operuje nagłówkiem i stopką dokumentu</li> <li>– tworzy strony tytułowe</li> <li>– współpracuje przy edycji dokumentu z innymi użytkownikami, korzystając z opcji recenzji dokumentu</li> </ul>	
11	Sztuka prezentacji	<ul style="list-style-type: none"> <li>– korzysta z programu Microsoft PowerPoint</li> <li>– zna zasady zachowania się podczas występów publicznych</li> <li>– opracowuje plan prezentacji</li> <li>– zna narzędzia i pomoce wizualne wykorzystywane podczas prelekcji</li> <li>– prezentuje poprawnie sformatowaną treść slajdów</li> <li>– stosuje efekty i multimedia w prezentacji</li> </ul>	<ul style="list-style-type: none"> <li>– dodaje do slajdów swój komentarz głosowy i zapisuje prezentację jako film</li> </ul>
P1	Nowoczesne technologie w służbie człowiekowi – projekt zespołowy	<ul style="list-style-type: none"> <li>– aktywnie uczestniczy w realizacji projektów informatycznych</li> <li>– przyjmuje różne role w zespole realizującym projekt</li> <li>– prezentuje efekty wspólnej pracy</li> <li>– uzupełnia swoją wiedzę, korzystając z zasobów udostępnionych na platformie do e-nauczania</li> </ul>	<ul style="list-style-type: none"> <li>– przyjmuje rolę lidera odpowiedzialnego za zespół i projekt</li> </ul>
12	Moja cyfrowa tożsamość	<ul style="list-style-type: none"> <li>– definiuje pojęcie cyfrowej tożsamości</li> <li>– zna problemy zarządzania zasobami cyfrowymi</li> <li>– bezpiecznie kreuje swój wizerunek w przestrzeni medialnej</li> <li>– rozumie pojęcie wirtualnej komunikacji i komunikuje się z innymi w środowisku wirtualnym</li> <li>– dostrzega zalety i wady komunikacji wirtualnej oraz posługiwania się cyfrową tożsamością</li> <li>– rozumie pojęcie hejtu i dostrzega jego destrukcyjny wpływ</li> <li>– rozumie zagrożenia wynikające z upraszczania komunikacji za pośrednictwem sieci</li> <li>– zna narzędzia wirtualnej komunikacji</li> </ul>	<ul style="list-style-type: none"> <li>– wie, czym jest zautomatyzowane profilowanie i przetwarzanie danych</li> <li>– zna prawa przysługujące osobom, których dane są wykorzystywane</li> </ul>

13	Przemiany społeczne a technologie	<ul style="list-style-type: none"> <li>– rozumie i wymienia czynniki przemian społecznych</li> <li>– dostrzega możliwości wynikające z przemian gospodarczych</li> <li>– określa obszary w społeczeństwie, na które wpływa rozwój technologii</li> <li>– rozumie potrzebę stosowania regulacji prawnych i norm etycznych</li> <li>– wskazuje pozytywne i negatywne skutki rozwoju technologii informacyjnej</li> <li>– zna wyzwania, przed którymi stoi edukacja</li> <li>– operuje pojęciami: e-zasoby, e-usługi, e-learning</li> <li>– rozumie pojęcie mediów i przestrzeni medialnej w kontekście IT</li> <li>– wskazuje możliwości zapobiegania negatywnym skutkom rozwoju technologii</li> </ul>	<ul style="list-style-type: none"> <li>– wymyśla rozwiązania technologiczne, których nie ma jeszcze na rynku</li> </ul>
14	Cyber-bezpieczeństwo	<ul style="list-style-type: none"> <li>– rozpoznaje zagrożenia związane z oprogramowaniem komputerowym</li> <li>– dba o przestrzeganie podstawowych zasad bezpieczeństwa, korzystając z urządzeń mobilnych czy komputera</li> <li>– bezpiecznie korzysta z bankowości elektronicznej</li> <li>– umiejętnie i w bezpieczny sposób weryfikuje własną tożsamość, korzystając z e-usług</li> <li>– rozumie związek ochrony danych osobowych z cyberbezpieczeństwem</li> <li>– właściwie zachowuje się w sytuacji cyberprzemocy</li> <li>– stosuje pojęcia związane z bezpieczeństwem w internecie</li> </ul>	<ul style="list-style-type: none"> <li>– wymienia symptomy wskazujące na zainfekowanie komputera złośliwym oprogramowaniem</li> <li>– wie, czym jest infrastruktura krytyczna i jak się ją chroni</li> </ul>
15	Podstawy kryptografii	<ul style="list-style-type: none"> <li>– objaśnia rolę technik uwierzytelniania, kryptografii i podpisu elektronicznego</li> </ul>	<ul style="list-style-type: none"> <li>– wyjaśnia, czym są: steganografia, kryptologia, kryptografia, kryptogram, kryptoanaliza</li> <li>– zna zasadę Kerckhoffs'a</li> <li>– wyjaśnia, jak kolejne certyfikaty protokołu SSL (DV, OV i EV) zwiększają poziom bezpieczeństwa usług sieciowych</li> <li>– wyjaśnia znaczenie problemu bezkolizyjności funkcji hashujących i ich znaczenie dla kryptoanalizy</li> </ul>

			<ul style="list-style-type: none"> <li>– prezentuje przykład zaawansowanej kryptoanalizy</li> </ul>
16	Tworzenie stron internetowych	<ul style="list-style-type: none"> <li>– korzysta z różnych przeglądarek internetowych</li> <li>– zna strukturę strony WWW</li> <li>– definiuje podstawowe znaczniki HTML</li> <li>– korzysta z atrybutów znaczników</li> <li>– zna reguły stosowania arkuszy stylów w połączeniu z kodem HTML</li> <li>– stosuje narzędzia wspierające pisanie kodu źródłowego</li> <li>– wyszukuje informacje w sieci i korzysta z zasobów witryn internetowych na temat tworzenia stron WWW</li> </ul>	<ul style="list-style-type: none"> <li>– tworzy rozbudowaną stronę WWW z podstronami, tabelą, elementami graficznymi, formatując jej wygląd za pomocą stylów CSS</li> </ul>
17	Grafika 2D i 3D	<ul style="list-style-type: none"> <li>– rozróżnia pojęcia grafiki rastrowej i wektorowej</li> <li>– stosuje właściwe narzędzia do edycji zdjęć w wybranym programie graficznym</li> <li>– wykonuje różne operacje na obrazie w grafice rastrowej</li> <li>– zna różne formaty graficzne dla plików i korzysta z nich</li> <li>– modeluje proste obiekty w grafice 3D za pomocą wybranego oprogramowania</li> <li>– rysuje za pomocą narzędzi grafiki wektorowej</li> </ul>	<ul style="list-style-type: none"> <li>– tworzy bryły obrotowe i kompozycje obiektów na scenie</li> <li>– tworzy wektorowe modele sfotografowanego przez siebie wybranego obiektu</li> <li>– projektuje bardziej skomplikowane trójwymiarowe modele</li> <li>– tworzy trójwymiarowe modele dowolnego budynku</li> </ul>
18	JavaScript na stronach WWW	<ul style="list-style-type: none"> <li>– projektuje modele dwuwymiarowe i trójwymiarowe</li> <li>– wykorzystuje różne formaty obrazów</li> <li>– przekształca pliki graficzne, uwzględniając wielkość i jakość obrazów</li> <li>– tworzy stronę internetową zgodnie ze standardami, wzbogaconą listami i elementami dynamicznymi</li> <li>– korzysta z oprogramowania i serwisów przeznaczonych do tworzenia stron</li> <li>– publikuje własną stronę w internecie</li> <li>– programuje elementy strony internetowej współpracujące z siecią bazą danych</li> </ul>	<ul style="list-style-type: none"> <li>– wie, czym jest hosting</li> <li>– posługuje się systemem CMS</li> <li>– sprawdza dostępność nazw dla domeny</li> <li>– wie, czym są WordPress, Joomla!, Drupal</li> </ul>
19	Animacja komputerowa	<ul style="list-style-type: none"> <li>– tworzy i edytuje dwuwymiarowe oraz trójwymiarowe wizualizacje i animacje</li> <li>– stosuje właściwe formaty plików graficznych</li> </ul>	

P2	Responsywna strona WWW w systemie CMS – projekt zespołowy	<ul style="list-style-type: none"> <li>– aktywnie uczestniczy w realizacji projektów informatycznych</li> <li>– przyjmuje różne role w zespole realizującym projekt</li> <li>– prezentuje efekty wspólnej pracy</li> <li>– publikuje własną stronę w internecie</li> </ul>	<ul style="list-style-type: none"> <li>– przyjmuje rolę lidera odpowiedzialnego za zespół i projekt</li> </ul>
20	Jak pobierać dane do arkusza kalkulacyjnego	<ul style="list-style-type: none"> <li>– gromadzi w tabeli arkusza kalkulacyjnego dane pochodzące z różnych źródeł</li> <li>– korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych</li> <li>– filtruje dane według kilku kryteriów</li> </ul>	
21	Wyciągamy wiedzę z danych	<ul style="list-style-type: none"> <li>– dobiera wykresy do rodzaju danych</li> <li>– analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych</li> </ul>	
22	Zaawansowane formuły	<ul style="list-style-type: none"> <li>– stosuje zaawansowane funkcje arkusza kalkulacyjnego w zależności od rodzaju danych</li> </ul>	
23	Konsekwencje zaokrąglania liczb	<ul style="list-style-type: none"> <li>– wyjaśnia, jakie może być źródło błędów pojawiających się w obliczeniach komputerowych: błąd zaokrąglenia, błąd przybliżenia</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania w arkuszu kalkulacyjnym z wykorzystaniem funkcji służących do zaokrąglania liczb</li> <li>– zna właściwości formatu walutowego</li> <li>– projektuje obliczenia walutowe</li> </ul>
24	Korespondencja seryjna	<ul style="list-style-type: none"> <li>– do rozwiązania problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki</li> <li>– definiuje korespondencję seryjną</li> <li>– wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach</li> <li>– definiuje relacje</li> <li>– stosuje filtrowanie</li> <li>– formułuje kwerendy</li> <li>– tworzy i modyfikuje formularze</li> <li>– drukuje raporty</li> </ul>	

25	Makropolecenia VBA	<ul style="list-style-type: none"> <li>– definiuje makropolecenia</li> <li>– zna możliwości wbudowanego języka programowania</li> </ul>	
26	Programowanie własnych funkcji	<ul style="list-style-type: none"> <li>– stosuje zasady programowania strukturalnego i obiektowego do rozwiązywania problemów</li> <li>– wykorzystuje wbudowany język programowania</li> </ul>	

**Rozkładu materiału nauczania informatyki w zakresie rozszerzonym dla liceum ogólnokształcącego i technikum – Informatyka na czasie, część 2 (propozycja)**

**Plan wynikowy (propozycja) cz.2**

Lp.	Temat	Osiągnięcia uczniów	
		Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
1	Od problemu do programu	<ul style="list-style-type: none"> <li>– definiuje pojęcie specyfikacja algorytmu, określa dane i wyniki</li> <li>– planuje kolejne kroki rozwiązania problemu</li> <li>– omawia różne sposoby przedstawiania algorytmów (opis słowny, lista kroków, schemat blokowy, pseudokod)</li> <li>– programuje i testuje rozwiązanie problemu</li> <li>– sprawdza działanie algorytmów dla różnych danych</li> <li>– tworzy algorytmy działania na liczbach całkowitych</li> <li>– stosuje w języku C++ podstawowe konstrukcje programistyczne (operacje wejścia i wyjścia, instrukcja warunkowa, operatory matematyczne i logiczne)</li> <li>– tworzy w języku C++ programy wykonujące działania na liczbach całkowitych</li> </ul>	<ul style="list-style-type: none"> <li>– dobiera struktury i typy danych do rodzaju problemu</li> <li>– wyszukuje optymalne rozwiązania problemów</li> <li>– ocenia efektywność algorytmu</li> <li>– objaśnia dobrany do danego problemu algorytm, uzasadnia jego poprawność i wybór</li> </ul>
2	Systemy liczbowe i reprezentacja danych w komputerze	<ul style="list-style-type: none"> <li>– definiuje pojęcie pozycyjnego systemu liczbowego</li> <li>– wymienia systemy liczbowe stosowane w informatyce</li> <li>– definiuje pojęcia bit i bajt</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku, z arkuszy maturalnych z lat poprzednich lub konkursów i olimpiad informatycznych</li> </ul>



		<ul style="list-style-type: none"> <li>– dokonuje konwersji między pozycyjnymi systemami liczbowymi, wykorzystując przy tym zależności między systemami binarnym i ósemkowym oraz binarnym i heksadecymalnym</li> <li>– omawia sposób reprezentowania liczb całkowitych w komputerze</li> <li>– wymienia typy danych służące do zapisu liczb całkowitych (short int, int, long int, long long int, unsigned), stosuje je w pisanych programach</li> <li>– opisuje, jak w komputerze reprezentowane są znaki i napisy (char, string), odwołuje się do znaku w napisie za pomocą indeksu</li> <li>– wyjaśnia, czym jest tablica kodów ASCII</li> <li>– omawia działanie operacji logicznych i reprezentację ich wyników w komputerze (wynik może przyjmować wartość prawda – 1 lub fałsz – 0, co zajmuje 1 bajt pamięci)</li> <li>– opisuje istotę cyfrowej reprezentacji w komputerze obrazów, dźwięków i animacji</li> </ul>	
3	Algorytmy zamiany reprezentacji liczb między systemami liczbowymi	<ul style="list-style-type: none"> <li>– tworzy programy do konwersji między liczbami w systemach binarnym i decymalnym</li> <li>– pisze programy konwertujące liczbę dziesiętną na liczbę w podanym systemie pozycyjnym</li> <li>– posługuje się środowiskiem programistycznym, strukturami danych oraz językiem programowania w stopniu umożliwiającym implementację omawianych algorytmów</li> <li>– stosuje binarną reprezentację liczby w algorytmie szybkiego podnoszenia do potęgi</li> </ul>	<ul style="list-style-type: none"> <li>– pisze programy zamieniające liczby z systemu decymalnego na system heksadecymalny</li> <li>– pisze programy o podwyższonym stopniu trudności z wykorzystaniem algorytmów zamiany: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych</li> <li>– posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</li> <li>– do rozwiązania problemu dobiera optymalny algorytm i struktury danych</li> <li>– uzasadnia poprawność zaproponowanego rozwiązania</li> <li>– korzysta z dostępnych bibliotek w tworzonych przez siebie programach</li> <li>– tworzy własne funkcje rozwiązujące problemy</li> </ul>

4	Czy to jest palindrom?	<ul style="list-style-type: none"> <li>– definiuje pojęcie palindromu</li> <li>– określa, czy dany napis lub liczba są palindromami</li> <li>– wykonuje operacje na napisach (wczytywanie napisów ze spacjami, sprawdzanie długości napisu, zamiana liter dużych na małe i odwrotnie, porównywanie znaków, znajdowanie oraz usuwanie fragmentów napisów)</li> <li>– definiuje własne funkcje w języku C++, wyjaśnia celowość ich stosowania, rozróżnia parametry formalne i aktualne</li> <li>– realizuje w języku C++ algorytmy sprawdzające, czy dany napis jest palindromem, oraz wyszukujące palindromy w zdaniach</li> <li>– opisuje popularne funkcje oraz metody stosowane dla zmiennych typu string (toupper, tolower, size, substr, erase)</li> </ul>	<ul style="list-style-type: none"> <li>– pisze programy dotyczące palindromów o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych</li> <li>– posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</li> <li>– optymalizuje algorytmy i ocenia ich efektywność</li> </ul>
5	Czy ta liczba jest pierwsza?	<ul style="list-style-type: none"> <li>– wymienia podstawowe własności liczb pierwszych</li> <li>– sprawdza, czy dana liczba jest pierwsza, stosując algorytm naiwny</li> <li>– rozkłada liczbę złożoną na czynniki pierwsze</li> <li>– wyznacza liczby bliźniacze</li> </ul>	<ul style="list-style-type: none"> <li>– implementuje algorytmy dotyczące liczb pierwszych o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych</li> <li>– posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</li> <li>– stosuje optymalny algorytm sprawdzający, czy liczba jest pierwsza, wykorzystując funkcję logiczną; uzasadnia jego efektywność</li> <li>– pisze program rozkładający liczbę złożoną na sumę dwóch liczb pierwszych (hipoteza Goldbacha)</li> </ul>
6	Działania na liczbach w systemach innych niż dziesiętny	<ul style="list-style-type: none"> <li>– wykonuje działania arytmetyczne na liczbach w różnych systemach pozycyjnych</li> <li>– wykonuje obliczenia na dowolnie dużych liczbach, wykorzystując napisy</li> <li>– wyjaśnia różnicę między operacjami na liczbach o podstawie od 1 do 9 i większej od 10</li> <li>– stosuje odejmowanie w dzieleniu pisemnym liczb binarnych</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje działania o podwyższonym stopniu trudności</li> <li>– pisze programy wykonujące operacje arytmetyczne na liczbach w różnych systemach pozycyjnych</li> <li>– optymalizuje programy, szacuje ich efektywność</li> </ul>

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

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

		<ul style="list-style-type: none"> <li>– wymienia i stosuje różne algorytmy znajdowania maksymalnej sumy elementów spójnych podciągów, oceniając ich złożoność obliczeniową</li> <li>– znajduje w zbiorze spójny podciąg o maksymalnej sumie i wypisuje jego elementy</li> </ul>	<ul style="list-style-type: none"> <li>– do rozwiązania problemu dobiera optymalny algorytm i struktury danych</li> </ul>
16	W poszukiwaniu lidera i idola	<ul style="list-style-type: none"> <li>– definiuje pojęcia idola w grupie i lidera w zbiorze</li> <li>– znajduje idola w grupie lub stwierdza jego brak</li> <li>– określa, czy w zbiorze jest lider</li> <li>– omawia i implementuje w języku C++ algorytmy szukania idola oraz lidera</li> <li>– ocenia złożoność obliczeniową stosowanych algorytmów i ich efektywność</li> <li>– stosuje tablice dwuwymiarowe w pisanych programach</li> <li>– stosuje funkcję sort z biblioteki STL do wyszukiwania lidera</li> </ul>	<ul style="list-style-type: none"> <li>– pisze programy o podwyższonym stopniu trudności</li> <li>– posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</li> <li>– do rozwiązania problemu dobiera optymalny algorytm i struktury danych</li> </ul>
17	Iteracja a rekurencja	<ul style="list-style-type: none"> <li>– opisuje zasadę działania rekurencji</li> <li>– implementuje w języku C++ algorytmy rekurencyjne, określa warunki brzegowe</li> <li>– porównuje iteracyjne i rekurencyjne wersje algorytmów</li> <li>– opisuje zasadę złotego podziału</li> <li>– oblicza <math>n</math>-ty wyraz ciągu Fibonacciego metodami iteracyjną i rekurencyjną</li> <li>– wyjaśnia, na czym polega rozszerzony algorytm Euklidesa, oraz implementuje go w języku C++</li> </ul>	<ul style="list-style-type: none"> <li>– pisze programy o podwyższonym stopniu trudności, np. sprawdzanie hipotezy Collatza</li> <li>– posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</li> <li>– do rozwiązania problemu dobiera optymalny algorytm i struktury danych</li> <li>– uzasadnia wybór iteracji lub rekurencji do rozwiązania problemu</li> <li>– szacuje złożoność czasową stosowanych algorytmów</li> <li>– oblicza liczbę wykonywanych operacji w algorytmach rekurencyjnych</li> </ul>
18	Metoda zachłanna	<ul style="list-style-type: none"> <li>– wyjaśnia, na czym polega metoda zachłanna, i wymienia przykłady jej stosowania</li> <li>– implementuje następujące algorytmy zachłanne: problem kasjera (wydawania reszty minimalną liczbą nominałów), problem telewizja/kinomana (optymalny harmonogram wykorzystania sali), pakowanie plecaka, wyszukiwanie optymalnej drogi</li> <li>– ocenia przydatność zastosowanych algorytmów</li> </ul>	<ul style="list-style-type: none"> <li>– pisze programy o podwyższonym stopniu trudności z wykorzystaniem algorytmów zachłannych, stosując rekurencję i algorytmy z nawrotami</li> <li>– posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</li> <li>– do rozwiązania problemu dobiera optymalny algorytm i struktury danych</li> </ul>

		<ul style="list-style-type: none"> <li>– stosuje własne kryterium porównania w funkcji sort z biblioteki STL</li> </ul>	<ul style="list-style-type: none"> <li>– objaśnia algorytm wybrany do rozwiązania problemu oraz ocenia jego efektywność i niedoskonałość</li> </ul>
19	Programowanie dynamiczne	<ul style="list-style-type: none"> <li>– wyjaśnia, na czym polega metoda dynamiczna</li> <li>– implementuje optymalne algorytmy dotyczące problemu kasjera, telewizza, znajdowania drogi oraz pakowania plecaka</li> <li>– stosuje metodę dynamiczną do znajdowania najdłuższego wspólnego podciągu</li> <li>– porównuje metody zachłanną i dynamiczną</li> </ul>	<ul style="list-style-type: none"> <li>– pisze programy o podwyższonym stopniu trudności wykorzystujące algorytmy dynamiczne</li> <li>– posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</li> <li>– do rozwiązania problemu dobiera optymalny algorytm i struktury danych</li> </ul>
20	Dziel i zwyciężaj, czyli sortujemy sprawniej	<ul style="list-style-type: none"> <li>– omawia metodę „dziel i zwyciężaj” oraz rekurencję</li> <li>– wyjaśnia, na czym polega algorytm sortowania szybkiego oraz przez scalanie i implementuje je</li> <li>– ocenia i porównuje złożoność czasową i obliczeniową algorytmów</li> </ul>	<ul style="list-style-type: none"> <li>– pisze programy o podwyższonym stopniu trudności wykorzystujące metodę „dziel i zwyciężaj” oraz algorytmy sortowania</li> <li>– posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym</li> <li>– do rozwiązania problemu dobiera optymalny algorytm i struktury danych, a także korzysta z funkcji bibliotecznych</li> </ul>

### Plan wynikowy (propozycja) cz.3

Lp.	Temat	Osiągnięcia uczniów	
		Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
<b>Rozdział 1. Rozwiązywanie problemów z wykorzystaniem dynamicznych struktur danych</b>			
1	Odwrotna notacja polska (ONP)	<ul style="list-style-type: none"> <li>– definiuje pojęcia: notacja infiksowa, sufiksowa, prefiksowa, drzewo wyrażenia algebraicznego</li> <li>– zapisuje wyrażenia algebraiczne bez użycia nawiasów, w tym w odwrotnej notacji polskiej (ONP)</li> <li>– zamienia zapis wyrażenia algebraicznego z notacji tradycyjnej na ONP i odwrotnie</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych oraz ze zbioru zadań CKE</li> <li>– optymalizuje programy, szacuje ich efektywność</li> </ul>

		<ul style="list-style-type: none"> <li>– definiuje pojęcie dynamicznej struktury danych oraz stosu jako przykładu takiej struktury</li> <li>– wymienia przykłady operacji, jakie można wykonać na stosie, używa struktury stos w programach</li> <li>– omawia i implementuje algorytm zamiany wyrażenia algebraicznego z notacji tradycyjnej na ONP</li> <li>– omawia i implementuje algorytm obliczania wartości wyrażenia arytmetycznego zapisanego w ONP z wykorzystaniem rekurencji oraz stosu</li> </ul>	
2	Znajdowanie drogi wyjścia z labiryntu	<ul style="list-style-type: none"> <li>– omawia kolejkę jako kolejny przykład dynamicznej struktury danych i ją deklaruje</li> <li>– wymienia przykładowe operacje na kolejce i je stosuje</li> <li>– wyjaśnia algorytm przeszukiwania z nawrotami</li> <li>– definiuje pojęcie manipulator strumienia</li> <li>– omawia algorytm znajdowania wyjścia z labiryntu z wykorzystaniem rekurencji</li> <li>– stosuje kolejkę w iteracyjnym algorytmie znajdującym najkrótszą drogę wyjścia z labiryntu</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– wskazuje różnice między algorytmami znajdowania wyjścia z labiryntu z wykorzystaniem rekurencji i iteracji</li> <li>– uzasadnia użycie kolejki w algorytmie wyznaczania najkrótszej drogi wyjścia z labiryntu</li> </ul>
3	Wykorzystanie list w rozwiązywaniu problemów	<ul style="list-style-type: none"> <li>– definiuje pojęcie listy, w tym listy jednokierunkowej, dwukierunkowej, cyklicznej, wyjaśnia, co oznacza sekwencyjny dostęp do danych na liście</li> <li>– wyróżnia przykładowe operacje na liście i je stosuje</li> <li>– omawia problem Flawiusza, przeprowadza jego symulację</li> <li>– definiuje pojęcia porządek leksykograficzny, sortowanie leksykograficzne, sortowanie kubełkowe, sortowanie stabilne</li> <li>– omawia i implementuje algorytm porządkujący słowa leksykograficznie</li> <li>– wykorzystuje pliki tekstowe do wczytywania danych do programu i zapisywania wyników</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– szacuje złożoność czasową poznanych algorytmów, optymalizuje je</li> </ul>
4	Grafy. Znajdowanie najkrótszej drogi	<ul style="list-style-type: none"> <li>– definiuje graf, wymienia elementy składowe grafu i rodzaje grafów</li> <li>– zna sposoby reprezentacji grafu: macierz sąsiedztwa, listy sąsiedztwa</li> <li>– stosuje typ <code>vector</code> do reprezentacji grafu w postaci list sąsiedztwa</li> <li>– stosuje metody dla klasy <code>vector</code></li> </ul>	<ul style="list-style-type: none"> <li>– implementuje algorytmy przeszukiwania grafu w głąb (DFS) oraz przeszukiwania grafu wszerz (BFS)</li> <li>– omawia i implementuje algorytm Dijkstry</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– szacuje czasową złożoność algorytmów</li> </ul>

		<ul style="list-style-type: none"> <li>– wczytuje dane do programu z plików tekstowych</li> <li>– definiuje własne nazwy typów zmiennych</li> <li>– omawia algorytm przeszukiwania grafu w głąb (DFS)</li> <li>– omawia algorytm przeszukiwania grafu wszerek (BFS)</li> <li>– wyjaśnia, do czego służy algorytm Dijkstry</li> </ul>	
<b>Rozdział 2. Algorytmy numeryczne</b>			
5	Reprezentacja liczb rzeczywistych w komputerze	<ul style="list-style-type: none"> <li>– znajduje rozwinięcie binarne nieskracalnego ułamka właściwego</li> <li>– wyjaśnia, które ułamki właściwe mają skończone rozwinięcie binarne, a które okresowe</li> <li>– omawia różnice między stałoprzecinkową a zmiennoprzecinkową reprezentacją liczb rzeczywistych w komputerze</li> <li>– wyjaśnia pojęcia: cecha, mantysa, postać znormalizowana</li> <li>– zapisuje liczby w postaci znormalizowanej</li> <li>– definiuje pojęcia: liczba pojedynczej precyzji, liczba podwójnej precyzji, kod z nadmiarem</li> <li>– wykonuje działania na liczbach zmiennoprzecinkowych</li> </ul>	<ul style="list-style-type: none"> <li>– znajduje reprezentację liczby zapisanej w systemie dziesiętnym jako liczby pojedynczej i podwójnej precyzji</li> <li>– świadomie używa typów <code>float</code> i <code>double</code> w zadaniach</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– interpretuje wpływ zastosowanych typów na uzyskane wyniki</li> </ul>
6	Błędy w obliczeniach	<ul style="list-style-type: none"> <li>– rozróżnia pojęcia błąd względny i błąd bezwzględny</li> <li>– omawia przyczyny i rodzaje błędów w obliczeniach komputerowych – błąd reprezentacji, zaokrąglenia, przybliżenia, obcięcia</li> <li>– wskazuje różnice między algorytmem stabilnym i niestabilnym</li> <li>– znajduje pierwiastki równania kwadratowego algorytmem stabilnym i algorytmem niestabilnym</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– optymalizuje programy, stosując algorytmy stabilne oraz unikając błędów w obliczeniach</li> </ul>
7	Obliczanie wartości wielomianu	<ul style="list-style-type: none"> <li>– wyjaśnia schemat Hornera</li> <li>– implementuje algorytm obliczający wartość wielomianu algorytmem naiwnym oraz algorytm obliczający wartość wielomianu z zastosowaniem schematu Hornera</li> </ul>	<ul style="list-style-type: none"> <li>– stosuje schemat Hornera do zamiany liczby w systemie pozycyjnym o wybranej podstawie na liczbę dziesiętną oraz do szybkiego podnoszenia do potęgi</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
8	Metody obliczeń przybliżonych	<ul style="list-style-type: none"> <li>– wyjaśnia, na czym polegają metody obliczeń przybliżonych</li> <li>– znajduje metodą bisekcji miejsce zerowe funkcji w zadanym przedziale</li> </ul>	<ul style="list-style-type: none"> <li>– implementuje algorytmy numeryczne: znajdowania miejsc zerowych funkcji oraz obliczania pierwiastka kwadratowego metodą bisekcji, obliczania pierwiastka kwadratowego metodą Newtona–Raphsona, obliczania pola obszaru zamkniętego metodą prostokątów i metodą</li> </ul>



		<ul style="list-style-type: none"> <li>– oblicza pierwiastek kwadratowy metodą bisekcji i metodą Newtona–Raphsona, porównuje obie metody</li> <li>– oblicza pola obszarów zamkniętych metodą prostokątów oraz metodą trapezów</li> </ul>	<p>trapezów, znajdowania przybliżenia liczby pi oraz symulacji ruchów Browna metodą Monte Carlo</p> <ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
9	Algorytmy badające własności geometryczne	<ul style="list-style-type: none"> <li>– bada położenie punktów względem prostej i odcinka</li> <li>– sprawdza, czy dwa odcinki się przecinają</li> <li>– bada przynależność punktu do figury</li> </ul>	<ul style="list-style-type: none"> <li>– implementuje algorytmy badające własności geometryczne</li> <li>– w algorytmach badających własności geometryczne wykorzystuje macierz oraz regułę Sarrusa do obliczania wyznacznika macierzy</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
10	Fraktale	<ul style="list-style-type: none"> <li>– wyjaśnia, czym jest fraktal</li> <li>– wskazuje przykłady struktur fraktalnych występujących w przyrodzie</li> <li>– podaje przykłady fraktali (zbiór Cantora, drzewo binarne, płatek Kocha, dywan Sierpińskiego) i wyjaśnia sposób tworzenia tych fraktali</li> </ul>	<ul style="list-style-type: none"> <li>– implementuje w języku JavaScript algorytmy generujące fraktale danego stopnia</li> <li>– stosuje metodę IFS do tworzenia fraktali w arkuszu kalkulacyjnym</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
<b>Rozdział 3. Zaawansowane algorytmy i techniki programistyczne</b>			
11	Wyszukiwanie wzorca w tekście	<ul style="list-style-type: none"> <li>– definiuje problem wyszukiwania wzorca w tekście</li> <li>– wyszukuje wzorec w tekście algorytmem naiwnym</li> <li>– wyjaśnia, na czym polega metoda haszowania, i czym są: funkcja haszująca, klucz, hasz, kolizja</li> </ul>	<ul style="list-style-type: none"> <li>– omawia algorytm Karpa–Rabina do wyszukiwania wzorca w tekście z zastosowaniem funkcji haszującej</li> <li>– stosuje funkcję haszującą oraz algorytm Karpa–Rabina w programach wyszukujących wzorec w tekście</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– ocenia czasową złożoność obliczeniową algorytmów</li> </ul>
12	Szyfrowanie z kluczem publicznym. Algorytm RSA	<ul style="list-style-type: none"> <li>– wskazuje różnice między kryptografią symetryczną i kryptografią asymetryczną, definiuje pojęcia klucz prywatny i klucz publiczny</li> <li>– wyjaśnia, do czego służy algorytm RSA, i wyróżnia główne etapy tego algorytmu (generowanie kluczy, szyfrowanie z kluczem publicznym oraz deszyfrowanie z kluczem prywatnym)</li> <li>– wyjaśnia, jak generuje się klucze publiczny i prywatny oraz jak wykorzystuje się je do szyfrowania i deszyfrowania informacji w algorytmie RSA</li> </ul>	<ul style="list-style-type: none"> <li>– pisze program generujący klucz prywatny i klucz publiczny w algorytmie RSA</li> <li>– pisze programy szyfrujące i deszyfrujące informacje w algorytmie RSA</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>

13	Programowanie obiektowe	<ul style="list-style-type: none"> <li>– wyjaśnia, na czym polega programowanie strukturalne, czym charakteryzują się metoda zstępująca i metoda wstępująca</li> <li>– definiuje programowanie obiektowe i podstawowe pojęcia z nim związane</li> <li>– definiuje własne klasy korzystając ze specyfikatorów dostępu</li> <li>– deklaruje konstruktory w klasach</li> <li>– wyjaśnia, na czym polega przeciążanie metod</li> </ul>	<ul style="list-style-type: none"> <li>– definiuje i implementuje własne klasy, obiekty, atrybuty i metody</li> <li>– wyjaśnia, na czym polega polimorfizm i czym są metody wirtualne</li> <li>– stosuje hierarchię klas</li> <li>– wyjaśnia, na czym polega hermetyzacja danych i jakie jest zastosowanie operatora zasięgu</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
<b>Rozdział 4. Relacyjne bazy danych</b>			
14	Wprowadzenie do relacyjnych baz danych	<ul style="list-style-type: none"> <li>– wyjaśnia, czym jest baza danych, oraz definiuje podstawowe pojęcia z nią związane: rekord, pole rekordu, tabela bazy danych, atrybut, relacja, klucz główny, klucz obcy, redundancja, formularz, kwerenda, raport, system zarządzania bazą danych</li> <li>– rozróżnia typy relacji w bazach danych</li> <li>– wprowadza i modyfikuje dane w tabelach</li> <li>– wyszukuje informacje w bazach danych, stosując filtrowanie i kwerendy</li> <li>– stosuje formularze do wprowadzania i modyfikowania danych</li> <li>– eksportuje dane, wykorzystując raporty</li> <li>– importuje dane do tabel</li> </ul>	<ul style="list-style-type: none"> <li>– projektuje i tworzy relacyjne bazy danych</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
15	Wykorzystanie danych pochodzących z kwerend	<ul style="list-style-type: none"> <li>– definiuje pojęcia: tabela pomostowa, klucz złożony, kwerenda wybierająca, kwerenda krzyżowa</li> <li>– łączy dane w bazach za pomocą tabeli pomostowej</li> <li>– stosuje kwerendy krzyżowe i wybierające</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
16	Podstawy języka SQL	<ul style="list-style-type: none"> <li>– opisuje przeznaczenie języka SQL, wymienia podstawowe klauzule języka</li> <li>– wykorzystuje język SQL do tworzenia i usuwania baz danych, dodawania tabel do baz danych, usuwania tabel z baz, dodawania rekordów do tabel, importowania danych do tabel, edycji rekordów</li> <li>– wyjaśnia działanie baz danych w architekturze klient-serwer</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>

		<ul style="list-style-type: none"> <li>– tworzy konta użytkowników i przydziela im uprawnienia do wybranej bazy</li> </ul>	
17	Zapytania w języku SQL	<ul style="list-style-type: none"> <li>– zna zasady tworzenia zapytań do bazy</li> <li>– formułuje zapytania zwracające określone dane</li> <li>– sortuje wyniki zapytań</li> <li>– eksportuje wyniki zapytania do pliku tekstowego</li> </ul>	<ul style="list-style-type: none"> <li>– formułuje zapytania w języku SQL, stosując selekcję, sortowanie, projekcję oraz agregowanie danych</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
18	Tworzenie aplikacji korzystającej z sieciowej bazy danych	<ul style="list-style-type: none"> <li>– wyróżnia etapy pracy nad aplikacją internetową</li> <li>– wyróżnia technologie back-end i front-end</li> <li>– wymienia kluczowe zadania w procesie tworzenia aplikacji</li> <li>– określa założenia projektu, jego funkcjonalność</li> <li>– instaluje i konfiguruje niezbędne oprogramowanie</li> <li>– przygotowuje bazę danych na potrzeby projektu</li> </ul>	<ul style="list-style-type: none"> <li>– tworzy aplikacje internetowe z przejrzystym interfejsem użytkownika korzystające z sieciowej bazy danych, samodzielnie je testuje i wprowadza poprawki</li> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> </ul>
P1	Pułapki cyfrowego świata	<ul style="list-style-type: none"> <li>– wyjaśnia, czym jest dokumentacja projektu, bierze czynny udział w jej tworzeniu</li> <li>– definiuje cel projektu</li> <li>– wie, czym jest dyskusja panelowa</li> <li>– aktywnie uczestniczy w realizacji projektu, wykorzystując specjalistyczne narzędzia do gromadzenia, opracowania i prezentacji danych oraz prowadzenia spotkań online</li> </ul>	<ul style="list-style-type: none"> <li>– przyjmuje rolę lidera odpowiedzialnego za zespół i projekt</li> <li>– przydziela zadania, nadzoruje pracę innych</li> <li>– przyjmuje funkcję eksperta lub moderatora</li> </ul>
<b>Rozdział 5. Rozwiązywanie różnych problemów z wykorzystaniem komputera</b>			
19	Sterujemy robotem	<ul style="list-style-type: none"> <li>– definiuje pojęcie robota</li> <li>– omawia budowę oraz wybrane parametry robotów (serwomotor, magnetometr, akcelerometr, diody, czujniki, wyświetlacz)</li> <li>– programuje roboty wykorzystując specjalistyczne narzędzia (aplikacje), w tym symulatory online</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– wykazuje się kreatywnością przy projektowaniu własnych projektów, takich jak np.: stacja pogodowa, gry logiczne i zręcznościowe, mierzenie odległości od przeszkód, loty synchroniczne (drony)</li> <li>– stosuje aplikacje mobilne do sterowania robotami</li> </ul>
20	Sztuka publikowania w sieci	<ul style="list-style-type: none"> <li>– opracowuje interesujące treści internetowe dostosowane do potrzeb potencjalnych odbiorców, wykorzystując zasadę 5W</li> <li>– dba o identyfikację wizualną i dźwiękową</li> <li>– stosuje elementy przyciągające uwagę użytkowników, np. lidy, hastagi, właściwie dobiera słowa</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– tworzy podcasty i publikacje wideo na wybrane tematy wymagające dużego nakładu pracy (np. promocja czy jubileusz szkoły, szkolny festiwal kultury lub nauki), korzysta z zaawansowanych narzędzi</li> </ul>

		<ul style="list-style-type: none"> <li>– korzysta z narzędzi graficznych i multimedialnych do wzbogacania treści</li> <li>– rozróżnia pojęcia webcast, webinarium, screencast i podcast</li> <li>– montuje materiały, wykorzystując specjalistyczne oprogramowanie (np. Stream z pakietu Office 365)</li> <li>– dba o właściwy rozmiar materiałów, stosując różne rodzaje kompresji oraz właściwy format plików</li> <li>– występuje przed kamerą i mikrofonem, przekazując treści w sposób atrakcyjny dla odbiorców, utrzymujący ich uwagę</li> </ul>	
21	Grafiki informacyjne	<ul style="list-style-type: none"> <li>– wymienia różne sposoby przedstawiania informacji</li> <li>– definiuje pojęcie grafiki informacyjnej, wymienia przykłady grafiki narracyjnej i wizualizacji danych</li> <li>– tworzy infografikę z wykorzystaniem języka piktogramów Isotype</li> <li>– poprawnie projektuje proste infografiki zawierające uporządkowane informacje (chronologicznie, według kategorii, przestrzenne czy hierarchiczne), umiejętnie stosuje tekst i obraz</li> <li>– wymienia, co powoduje wrażenie chaosu na infografice</li> <li>– przy projektowaniu świadomie dobiera barwy i wykorzystuje funkcje kolorów</li> <li>– w swoich projektach zwraca uwagę na dostosowanie treści do odbiorców</li> </ul>	<ul style="list-style-type: none"> <li>– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku</li> <li>– wykazuje się kreatywnością, tworząc infografiki dotyczące globalnych problemów współczesnego świata, lokalnych, szkolnej społeczności czy też środowisk młodzieżowych</li> </ul>