

Wymagania edukacyjne z informatyki dla klasy:
pierwszej, drugiej, trzeciej, czwartej
czteroletniego liceum (zakres rozszerzony)

Wymagania programowe na poszczególne oceny przygotowane na podstawie treści zawartych w podstawie programowej oraz w podręczniku „Informatyka na czasie 1, 2, 3” wydawnictwo Nowa Era, zakres podstawowy. Numer dopuszczenia MEN 990/2019

KLASA 1

2	3	4	5	6
Systemy operacyjne w środowisku sieciowym				
<ul style="list-style-type: none"> • wymienia urządzenia mobilne zaliczane do systemów komputerowych, • wymienia elementy budowy systemu operacyjnego, • rozumie pojęcie „ścieżka dostępu” w kontekście systemów plików, • sprawdza i wymienia atrybuty pliku, • opisuje, jak uruchomić system BIOS na komputerze, • wyjaśnia konieczność tworzenia bezpiecznych haseł, • wymienia metody zabezpieczania danych na komputerze, • uruchamia Menedżera zadań w systemie Windows, • wymienia problemy, jakie można napotkać podczas korzystania z komputera, • wyjaśnia pojęcie sztucznej inteligencji, • opisuje, czym jest chmura obliczeniowa, • wymienia zastosowania automatów i robotów, • podaje przykłady wykorzystania druku 3D, 	<ul style="list-style-type: none"> • wymienia urządzenia wchodzące w skład sieci komputerowej, • identyfikuje wersję systemu operacyjnego swojego smartfona (komputera), • wyjaśnia różnicę pomiędzy bezwzględną i względną ścieżką dostępu, • określa różnicę pomiędzy BIOS a UEFI, • rozumie pojęcie serwera, • opisuje zasady bezpiecznego korzystania z systemu operacyjnego, • wyjaśnia, jak założyć konto użytkownika w używanym przez siebie systemie operacyjnym, • konstruuje bezpieczne hasła, • kopiuje dane, aby wykonać kopię zapasową na zewnętrznym nośniku, • uruchamia komputer w trybie awaryjnym, • sprawdza obciążenie procesora, • wyjaśnia pojęcia fragmentacji i defragmentacji dysku, • wyjaśnia różnicę pomiędzy systemami plików FAT32 oraz 	<ul style="list-style-type: none"> • opisuje, czym jest model warstwowy systemu komputerowego, • wymienia i wyjaśnia zadania systemu operacyjnego, • instaluje i aktualizuje oprogramowanie, • podczas zamykania aplikacji umiejętnie korzysta z Menedżera zadań w systemie Windows, • korzysta z narzędzi oczyszczania dysku, • opisuje procedurę wykonywania kopii zapasowej dla systemu operacyjnego w szkolnej pracowni, • opisuje zastosowania rzeczywistości wirtualnej i rozszerzonej • dobiera kryteria wyboru elementów zestawu komputerowego w zależności od jego przeznaczenia, • wymienia i opisuje urządzenia sieciowe, • opisuje sieci komputerowe ze względu na zasięg ich działania, • wyjaśnia budowę adresów MAC i sprawdza je na 	<ul style="list-style-type: none"> • opisuje każdą z warstw modelu systemu komputerowego, • charakteryzuje poszczególne elementy systemu operacyjnego, • opisuje działanie systemu operacyjnego, • modyfikuje uprawnienia konta użytkownika systemu operacyjnego, • wykonuje defragmentację dysku, • wymienia i opisuje zastosowania sieci internet, • wyjaśnia pojęcie i budowę ramki jako porcji informacji w transmisji danych, • opisuje sposób adresowania urządzeń w sieci internet, • wyjaśnia sposób komunikacji między urządzeniami tej samej oraz różnych sieci, • omawia procesy enkapsulacji i dekapulacji danych w transmisji sieciowej, • opisuje protokoły wykorzystywane podczas transmisji danych w sieci, 	<ul style="list-style-type: none"> • obsługuje różne systemy operacyjne, • korzysta z poleceń trybu tekstowego Windows, • kopiuje pliki w trybie tekstowym Windows za pomocą ścieżek względnych i bezwzględnych, • dokonuje istotnych zmian w BIOS, • wyjaśnia zasadę działania sztucznego neuronu i sieci neuronowej, • projektuje modele warstwowe skomplikowanych procesów życia codziennego,

<ul style="list-style-type: none"> • zna i opisuje zagrożenia wynikające z rozwoju technologii, • wyjaśnia pojęcia: „sieci komputerowe” i „urządzenia sieciowe”, • opisuje przeznaczenie warstwowych modeli sieci, • wyjaśnia przeznaczenie protokołu IP, • wyjaśnia pojęcie sztucznej inteligencji, • opisuje, czym jest chmura obliczeniowa, • wymienia zastosowania automatów i robotów, • podaje przykłady wykorzystania druku 3D, • zna i opisuje zagrożenia wynikające z rozwoju technologii, 	<p>NTFS,</p> <ul style="list-style-type: none"> • definiuje pojęcie systemu operacyjnego, • wyjaśnia różnicę pomiędzy wirtualną a rozszerzoną rzeczywistością • nazywa różne porty urządzeń sieciowych, • wymienia korzyści wynikające z korzystania z warstwowych modeli sieci, • opisuje budowę adresu IPv4 w wersjach dziesiętnej i binarnej, • rozróżnia typy domen (krajowe, funkcjonalne), • wyjaśnia pojęcie systemu DNS, • opisuje budowę adresu URL, • wyjaśnia, czym są e-usługi, 	<p>komputerze z systemem Windows,</p> <ul style="list-style-type: none"> • wyjaśnia pojęcia: adres IP, maska podsieci, • schematycznie przedstawia i omawia model warstwowy TCP/IP, • określa relacje między podmiotami rynku e-usług, • korzysta z wybranych e-usług, 	<ul style="list-style-type: none"> • podłącza i konfiguruje urządzenia sieciowe, • projektuje domową sieć komputerową, • opisuje sposób tworzenia i budowę domeny internetowej, • konfiguruje urządzenie do pracy w internecie i omawia ten proces, • wymienia i omawia protokoły usług internetowych, • diagnozuje stan połączeń internetowych, 	
Edytor tekstu i prezentacje				
<ul style="list-style-type: none"> • korzysta w podstawowym zakresie z formatowania tekstów w edytorze tekstowym, • wymienia etapy pracy nad dobrym wystąpieniem publicznym, • wymienia programy komputerowe do tworzenia prezentacji, 	<ul style="list-style-type: none"> • korzysta z szablonów w edytorze tekstów, • poprawnie stosuje style nagłówkowe, • generuje losowe bloki tekstowe, • ustawia marginesy w dokumencie 	<ul style="list-style-type: none"> • dzieli tekst na kolumny, • wymienia cechy dobrej prezentacji, • tworzy ciekawe przejścia między slajdami, 	<ul style="list-style-type: none"> • korzysta z automatycznej numeracji tytułów oraz tworzy spis treści, • tworzy spisy ilustracji i tabel, • pracuje z dokumentem wspólnie z innymi osobami, korzystając z narzędzi pracy grupowej, • wykorzystuje opcje recenzji 	<ul style="list-style-type: none"> • korzysta z różnych narzędzi (w tym mobilnych) podczas prezentacji, • bierze udział w projektach zespołowych jako odpowiedzialny lider projektu, • wypełnia wszystkie zadania wynikające z roli

<ul style="list-style-type: none"> • wyjaśnia pojęcia: wykluczenie i włączenie cyfrowe, 			<p>dokumentu,</p> <ul style="list-style-type: none"> • wygłasza prelekcję wybrany temat zgodnie z zasadami dobrego wystąpienia, • tworzy dokładny plan wystąpienia na dowolny temat, • stosuje efekty na slajdach prezentacji, • umieszcza filmy i ścieżki audio w prezentacji, • prezentuje kompletny projekt na forum klasy, 	<p>powierzonej mu w projekcie,</p>
<p>Społeczeństwo w internecie</p>				
<ul style="list-style-type: none"> • wyjaśnia pojęcie cyfrowej tożsamości, • wymienia sposoby uwierzytelniania użytkowników e-usług, • wskazuje miejsca występowania e-zasobów, • rozróżnia wyszukiwarki od przeglądarek internetowych, • podaje przykłady negatywnych zachowań w sieci internet, • wyjaśnia znaczenie kryptografii dla bezpieczeństwa danych, 	<ul style="list-style-type: none"> • wyjaśnia pojęcia: prawo autorskie, licencja, • rozróżnia i definiuje pojęcia wolnego i otwartego oprogramowania, • wyjaśnia pojęcie licencji Creative Commons, • wymienia wiarygodne źródła informacji w sieci internet, • wyjaśnia, jak sprawdzić właściciela serwisu internetowego, • omawia pojęcia związane z kryptografią, • opisuje pojęcie cyfrowej tożsamości, • wymienia zasady komunikacji w sieci internet (netykieta), 	<ul style="list-style-type: none"> • podaje cechy różnych rodzajów licencji oprogramowania, • stosuje symbole i wyrażenia w wyszukiwarkach internetowych, • wymienia zasady ochrony danych osobowych, • opisuje zastosowania technologii komputerowej w różnych dziedzinach życia, • opisuje rodzaje ataków sieciowych, • omawia działanie protokołu SSL, 	<ul style="list-style-type: none"> • wyjaśnia zasady stosowania prawa autorskiego, • wykorzystuje narzędzia współpracy zdalnej, • wyjaśnia, jak zwiększyć swoje bezpieczeństwo w sieci poprzez stosowanie różnych technik, • omawia kryptoanalizę na wybranym przez siebie przykładzie, 	<ul style="list-style-type: none"> • wyjaśnia, czym są: steganografia, kryptologia, kryptografia, kryptogram, kryptoanaliza • prezentuje przykład zaawansowanej kryptoanalizy

	<ul style="list-style-type: none"> wymienia zagrożenia wynikające ze złej komunikacji w sieci, opisuje wpływ rozwoju technologii na zmiany w społeczeństwie, wymienia i opisuje rodzaje szkodliwego oprogramowania 			
Strony WWW i grafika komputerowa				
<ul style="list-style-type: none"> zapisuje plik, nadając mu rozszerzenie .html, rozdziela sekcje HEAD i BODY oraz opisuje różnicę między tymi częściami kodu, wymienia podstawowe znaczniki formatowania tekstu w języku HTML, opisuje budowę znacznika HTML, wyjaśnia pojęcie responsywności strony WWW, uruchamia stronę WWW na smartfonie, określa różnicę pomiędzy grafiką rastrową a wektorową, zapisuje wynik swojej pracy w różnych formatach graficznych, wyjaśnia, jak uruchomić środowisko do grafiki 3D online, 	<ul style="list-style-type: none"> wyjaśnia, czym są e-zasoby, tworzy stronę tytułową w dokumencie tekstowym, wyjaśnia, jak przygotować dobre wystąpienie, zna narzędzia, dzięki którym można dobrać zestaw pasujących do siebie kolorów, opisuje podstawową strukturę strony w języku HTML, tworzy nagłówki w języku HTML, wstawia komentarze w kodzie HTML, tworzy listy uporządkowane i nieuporządkowane, rozumie cel pozycjonowania stron WWW, skaluje i kadruje obraz, dostosowując go do zadanego rozmiaru, wymienia podstawowe 	<ul style="list-style-type: none"> tworzy modele 3D z prostych brył 3D i ich przekształceń, umieszcza zdjęcia na stronie WWW, tworzy linki do zasobów zewnętrznych oraz miejsc w obrębie jednej strony, poprawnie i na różne sposoby korzysta z opisu kolorów w języku HTML, wyjaśnia działanie hostingu stron internetowych, wykorzystuje język JavaScript podczas tworzenia stron internetowych wymienia podstawowe narzędzia programu GIMP, korzysta z warstw podczas pracy z programem GIMP, pracuje na warstwach w programie do grafiki wektorowej, 	<ul style="list-style-type: none"> korzysta ze ścieżek względnych i bezwzględnych w kodzie HTML, poprawnie tworzy tabele o dowolnej strukturze, dołącza style kaskadowe do dokumentu HTML, tworzy ciekawą stronę WWW i publikuje ją w internecie, poprawnie używa narzędzia do rysowania krzywych Béziera, wycina dowolne elementy z obrazu rastrowego, tworzy w programach do grafiki wektorowej infografiki według wzoru, tworzy bryły obrotowe 3D na podstawie ich przekroju, tworzy trójwymiarowe animacje, 	<ul style="list-style-type: none"> tworzy style opisujące wygląd strony WWW, dołącza do strony elementy odpowiedzialne za jej responsywność, buduje stronę z wykorzystaniem systemu CMS i publikuje ją w internecie, tworzy złożone modele 3D.

	<p>narzędzia programu Inkscape,</p> <ul style="list-style-type: none">• tworzy dwuwymiarowe animacje,			
--	---	--	--	--

KLASA 2

2	3	4	5	6
Arkusz kalkulacyjny				
<ul style="list-style-type: none"> wprowadza dane różnego typu do arkusza kalkulacyjnego, omawia zastosowania korespondencji seryjnej, wyjaśnia relacje w bazach danych. 	<ul style="list-style-type: none"> pobiera dane do arkusza kalkulacyjnego ze źródeł zewnętrznych, filtruje dane w arkuszu kalkulacyjnym, tworzy różne wykresy w arkuszu kalkulacyjnym w zależności od rodzaju danych, bierze udział w projektach informatycznych jako członek zespołu. 	<ul style="list-style-type: none"> przeprowadza analizę danych zgromadzonych w arkuszu kalkulacyjnym, omawia błąd zaokrąglenia i błąd przybliżenia w obliczeniach komputerowych, dobiera środowisko informatyczne do rodzaju rozwiązywanego problemu, wyszukuje informacje zgromadzone w bazach danych, w bazach danych wykorzystuje kwerendy, filtrowanie 	<ul style="list-style-type: none"> wykorzystuje zaawansowane formuły, opracowując dane w arkuszu kalkulacyjnym, stosuje funkcje zaokrąglające liczby, korzysta z możliwości obliczeń walutowych, 	
Algorytmy na liczbach całkowitych i tekstach				
<ul style="list-style-type: none"> definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja, wymienia sposoby reprezentacji algorytmów, korzysta z środowiska programistycznego: pisze w nim kod, kompiluje i 	<ul style="list-style-type: none"> przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu, dodaje liczbybinarne, konwertuje liczby między pozycyjnymi systemami liczbowymi, wykonuje działania arytmetyczne na liczbach w systemach liczbowych o 	<ul style="list-style-type: none"> określa specyfikację algorytmu (dane, wynik), pisze programy o różnym stopniu trudności, szacuje ich efektywność, przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, pseudokodu, dobiera typy danych do 	<ul style="list-style-type: none"> charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania, pisze programy o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku, optymalizuje rozwiązania, pisze programy 	<ul style="list-style-type: none"> charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,

<p>uruchamia program, odczytuje i zapisuje pliki,</p> <ul style="list-style-type: none"> • pisze programy o niewielkim stopniu trudności, • omawia pojęcia: złożoność obliczeniowa algorytmu, algorytm naiwny, algorytm optymalny, złożoność pesymistyczna, złożoność oczekiwana (średnia), • korzysta z podstawowych funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych, • wymienia podstawowe typy danych, operacje arytmetyczne i logiczne, • definiuje pojęcie systemów liczbowych, • wyjaśnia, czym jest tablica kodów ASCII, • wymienia systemy liczbowe używane w informatyce, • konwertuje liczby między systemami binarnym i decymalnym, • dodaje pisemnie liczby binarne, • wyjaśnia, czym są palindrom i anagram, podaje przykłady, • podaje definicje liczby pierwszej i liczby złożonej, • implementuje w języku C++ 	<p>różnych podstawach,</p> <ul style="list-style-type: none"> • przedstawia liczby w kodzie U2, • definiuje pojęcie zdania logicznego, charakteryzuje podstawowe operacje logiczne (koniunkcja, alternatywa, negacja) oraz operatory logiczne, • charakteryzuje wybrane typy zmiennych służących do zapisu liczb całkowitych w języku C++: short int, int, long int, long long int, • pisze programy wykonujące działania na liczbach całkowitych, • korzysta z biblioteki string do operacji na łańcuchach znaków, • wykonuje operacje na napisach, wykorzystując słowa kluczowe: size, find, substr, erase, toupper, tolower, • wczytuje napisy ze spacjami, wykorzystując słowo kluczowe getline, • tworzy algorytmy sprawdzające, czy napis jest palindromem, • wyjaśnia różnicę między parametrami formalnym i aktualnym, a także między zmiennymi lokalną i globalną, 	<p>realizacji problemu,</p> <ul style="list-style-type: none"> • stosuje zmienne typu unsigned w tworzonych programach, • pisze programy konwertujące liczby między systemem dziesiętnym i binarnym, • implementuje w języku C++ algorytmy wykonujące działania arytmetyczne na liczbach w różnych systemach, • w algorytmach zamiany wykorzystuje zależności między systemami binarnym, ósemkowym i heksadecymalnym, • omawia sposób reprezentacji obrazów w komputerze, korzystając z takich pojęć jak: piksel, model RGB, kanał alfa, • wyjaśnia, na czym polega digitalizacja (dyskretyzacja) dźwięku, • wyjaśnia zasadę tworzenia animacji, • implementuje w języku C++ algorytmy sprawdzające, czy napis jest palindromem, • implementuje w języku C++ i optymalizuje algorytm sprawdzający, 	<p>konwertujące liczby między różnymi systemami pozycyjnymi,</p> <ul style="list-style-type: none"> • w programach wykonujących działania na liczbach w różnych systemach pozycyjnych wykorzystuje bibliotekę string i strukturalne typy danych, • wykorzystuje rozwinięcie binarne liczb dziesiętnej w algorytmie szybkiego podnoszenia do potęgi, • wykonuje operacje arytmetyczne na liczbach w różnych systemach, implementuje je w języku C++, • stosuje różne sposoby przekazywania parametrów do funkcji, uzasadnia ich użycie, • pisze funkcje typu logicznego, np. sprawdzając, czy napis jest palindromem, • szyfruje dane wczytane z pliku z uwzględnieniem polskich znaków diakrytycznych, • pisze program odczytujący informację ukrytą za pomocą szyfru Cezara z wykorzystaniem analizy 	<ul style="list-style-type: none"> • pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych lub oznaczone trzema gwiazdkami w podręczniku, • wyszukuje palindromy lub anagramy w plikach tekstowych, • tworzy palindromy z napisów, dopisując minimalną liczbę znaków, • pisze program rozkładający liczbę złożoną na dwie liczby pierwsze (hipoteza Goldbacha), • implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję, • pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere'a) i różnych kluczy,
---	--	---	--	---

<p>algorytm zliczający dzielniki danej liczby,</p> <ul style="list-style-type: none"> • omawia geometryczną interpretację algorytmu Euklidesa, 	<ul style="list-style-type: none"> • implementuje w języku C++ algorytm naiwny sprawdzający, czy liczba jest pierwsza, • implementuje w języku C++ algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem, 	<p>czy liczba jest pierwsza,</p> <ul style="list-style-type: none"> • pisze program rozkładający liczby na czynniki pierwsze, • stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW, • wykorzystuje algorytm Euklidesa do działań na ułamkach, • szyfruje dane wczytane z pliku tekstowego, • zapisuje w postaci programu rozszerzony algorytm Euklidesa, wyjaśnia jego działanie i zastosowanie, 	<p>częstości znaków w tekście,</p> <ul style="list-style-type: none"> • wykorzystuje algorytm Euklidesa do działań na ułamkach, stosując struktury lub pary (typ pair), 	
Rozwiązywanie problemów z wykorzystaniem struktur danych				
<ul style="list-style-type: none"> • w pisanych programach korzysta ze strukturalnych typów danych: napisów, struktur, tablic, • definiuje pojęcia: kryptologia, kryptografia, kryptoanaliza, tekst jawny, klucz, szyfrogram, • rozróżnia szyfry podstawieniowe i przestawieniowe, • omawia szyfr Cezara jako przykład szyfru podstawieniowego • wyjaśnia, na czym polega 	<ul style="list-style-type: none"> • przedstawia w postaci algorytmu problem wyszukiwania anagramów, • przy pisaniu programów stosuje własne funkcje różnych typów, w tym funkcję typu void, • pisze program szyfrujący napis szyfrem Cezara, • omawia algorytm sita Eratostenesa, • przedstawia algorytm znajdowania spójnych podciągów, wyznaczania najdłuższego z 	<ul style="list-style-type: none"> • stosuje różne sposoby przekazywania parametrów do funkcji: przez wartość, referencję lub wskaźnik, • pisze programy sprawdzające, czy dwa napisy są anagramami, wykorzystując funkcję sort z biblioteki STL, • stosuje algorytm wyszukiwania binarnego i oszacowuje jego złożoność czasową, • pisze programy sortujące 	<ul style="list-style-type: none"> • dobiera struktury danych i metody do rodzaju problemu, • sprawdza, czy napisy są anagramami, stosując sortowanie lub zliczanie znaków, • przy testowaniu liczby na pierwszość stosuje funkcję typu logicznego, • wyszukuje liczby bliźniacze, 	<ul style="list-style-type: none"> • stosuje w programach algorytmy sortowania inne niż omawiane na lekcjach (np. heapsort), • w projektach zespołowych przyjmuje rolę lidera.

<p>łamanie szyfru,</p> <ul style="list-style-type: none">• omawia algorytm zliczania znaków w tekście,	<p>nich oraz podciągu o największej sumie elementów,</p>	<p>metodami prostymi z zastosowaniem funkcji typu void,</p> <ul style="list-style-type: none">• pisze program realizujący algorytm sita Eratostenesa,		
--	--	---	--	--

KLASA3

2	3	4	5	6
Rozwiązywanie problemów z wykorzystaniem struktur danych				
<ul style="list-style-type: none"> wypisuje liczby pierwsze z danego przedziału, stosując metodę sita Eratostenesa, wyszukuje w ciągu liczb spójne podciągi (nierosnący, niemalejący, stały), wskazuje najdłuższe, oblicza ich sumę, 	<ul style="list-style-type: none"> omawia algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic, przedstawia w postaci listy kroków algorytmy sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie, określa operacje dominujące, implementuje w języku C++ algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi, omawia rozszerzony algorytm Euklidesa, formułuje algorytm wydawania reszty minimalną liczbą monet, znajdowania drogi metodami zachłanną i dynamiczną, 	<ul style="list-style-type: none"> implementuje w języku C++ algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic, implementuje w języku C++ algorytmy wyszukujące spójne podciągi o różnych cechach, 	<ul style="list-style-type: none"> stosuje zaawansowane funkcje środowiska i języka programowania (np. z biblioteki STL), optymalizuje program realizujący algorytm sita Eratostenesa i szacuje jego złożoność czasową, wyszukuje spójne podciągi w plikach tekstowych, stosując optymalne algorytmy (w tym programowanie dynamiczne), wyjaśnia ich działanie, pisze programy wyszukujące lidera i idola w zbiorze, optymalizuje je, szacuje złożoność czasową, 	<ul style="list-style-type: none"> stosuje zaawansowane algorytmy i struktury danych do wyszukiwania spójnych podciągów,
Metody algorytmiczne				
<ul style="list-style-type: none"> wyjaśnia, na czym polega metoda „dziel i zwyciężaj”, 	<ul style="list-style-type: none"> implementuje w języku C++ algorytmy wyszukiwania liniowego i liniowego z 	<ul style="list-style-type: none"> stosuje algorytmy sortowania szybkiego i przez scalanie, 	<ul style="list-style-type: none"> pisze program wyszukujący jednocześnie minimum i maksimum w zbiorze z 	<ul style="list-style-type: none"> implementuje w języku C++ algorytm wyszukiwania

<ul style="list-style-type: none"> • wczytuje dane z pliku tekstowego, zapisuje wyniki w pliku, • omawia algorytmy wyszukiwania liczby w zbiorach uporządkowanym i nieuporządkowanym, • stosuje funkcję losującą w tworzonych programach, • omawia metody sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie na przykładowych danych, • definiuje pojęcia iteracji i rekurencji, • omawia zasadę złotego podziału, • opisuje rozszerzony algorytm Euklidesa, • omawia metody zachłanne na przykładzie problemu kasjera, harmonogramu sali, porównuje metody zachłanną i dynamiczną, 	<p>wartownikiem, porównuje ich efektywność,</p>	<ul style="list-style-type: none"> • stosuje metodę zachłanną w programach – problem kasjera, wyszukiwanie drogi, • porównuje algorytmy iteracyjne i rekurencyjne (liczbę wykonywanych operacji), szacuje ich złożoność czasową, 	<p>wykorzystaniem metody „dziel i zwyciężaj” oraz podaje wzór na liczbę wykonywanych operacji,</p> <ul style="list-style-type: none"> • szacuje złożoność obliczeniową programów sortujących, modyfikuje funkcje sortujące, zmieniając porządek sortowania, • wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach, • pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję, • do implementacji rozszerzonego algorytmu Euklidesa stosuje zarówno iterację, jak i rekurencję, • stosuje metody zachłanną i dynamiczną w problemach kasjera i wyszukiwania drogi, wskazuje wady i zalety obu metod, szacuje złożoność czasową, 	<p>binarnego w wersji rekurencyjnej,</p> <ul style="list-style-type: none"> • pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary), • stosuje zaawansowane algorytmy wyszukiwania, np. najlepszego wyboru (trwałych par), stosując rekurencję, • pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,
<p>Rozwiązywanie problemów z wykorzystaniem dynamicznych struktur danych</p>				
<ul style="list-style-type: none"> • pisze programy o niewielkim stopniu 	<ul style="list-style-type: none"> • wyróżnia operacje, które można wykonywać na 	<ul style="list-style-type: none"> • pisze programy o różnym stopniu trudności, szacuje 	<ul style="list-style-type: none"> • charakteryzuje sytuacje algorytmiczne, proponuje 	<ul style="list-style-type: none"> • charakteryzuje skomplikowane sytuacje

<p>trudności,</p> <ul style="list-style-type: none"> • wyjaśnia, co to jest notacja infiksowa, notacja prefiksowa, odwrotna notacja polska, drzewo wyrażenia algebraicznego, • definiuje pojęcie dynamicznej struktury danych, • definiuje dynamiczne struktury danych takie jak: stos, kolejka, lista, vector, • wymienia rodzaje list, • definiuje graf, wymienia elementy i rodzaje grafów, wymienia sposoby reprezentacji grafu (macierz sąsiedztwa, lista sąsiedztwa), 	<p>dynamicznych strukturach danych (stosie, kolejce, liście, typie vector),</p> <ul style="list-style-type: none"> • omawia zastosowanie dynamicznych struktur danych na różnych przykładach, • zapisuje wyrażenia algebraiczne bez użycia nawiasów, w tym w postaci odwrotnej notacji polskiej, • oblicza wartość wyrażenia arytmetycznego zapisanego w odwrotnej notacji polskiej, • omawia algorytm znajdowania wyjścia z labiryntu z wykorzystaniem iteracji i rekurencji, • stosuje typ vector do reprezentacji grafu w postaci list sąsiedztwa, • omawia algorytm przeszukiwania grafu w głąb (DFS), • omawia algorytm przeszukiwania grafu wszerz (BFS), • wyjaśnia, do czego służy algorytm Dijkstry, • wyjaśniam różnicę między przekazywaniem parametrów do funkcji przez wartość i przez referencję, 	<p>ich efektywność,</p> <ul style="list-style-type: none"> • dobiera typy danych do rozwiązania problemu, • do przeglądania grafu stosuje algorytm przeszukiwania w głąb (DFS) oraz algorytm przeszukiwania grafu wszerz (BFS), • omawia algorytm Dijkstry, 	<p>sposoby ich rozwiązania,</p> <ul style="list-style-type: none"> • pisze programy o podwyższonym stopniu trudności: rozwiązuje zadania oznaczone trzema gwiazdkami w podręczniku, • optymalizuje rozwiązania, • stosuje zaawansowane funkcje środowiska i języka programowania, • dobiera struktury danych i metody do rodzaju problemu, • szacuje złożoność algorytmów, • implementuje algorytmy grafowe – BFS, DFS, algorytm Dijkstry, • w reprezentacji liczb rzeczywistych w komputerze stosuje reprezentację stało- lub zmiennoprzecinkową zgodnie ze specyfikacją algorytmu, minimalizując błędy w obliczeniach, 	<p>algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych,</p> <ul style="list-style-type: none"> • pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych, <p>optymalizuje programy, szacuje ich efektywność</p> <ul style="list-style-type: none"> • wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach, np. sprawdzanie spójności grafu,
---	--	--	---	---

--	--	--	--	--

KLASA 4

2	3	4	5	6
Relacyjne bazy danych				
<ul style="list-style-type: none"> zna podstawowe pojęcia dotyczące relacyjnych baz danych, wie, co to jest język SQL, zna podstawowe klauzule tego języka, zna zasady tworzenia zapytań do bazy z wykorzystaniem języka SQL, 	<ul style="list-style-type: none"> wyszukuje informacje w bazach danych, tworzy kwerendy 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, tworzy konta użytkowników i przydziela im uprawnienia do wybranej bazy, używając języka SQL, formułuje zapytania zwracające określone dane, sortuje wyniki zapytań, 	<ul style="list-style-type: none"> projektuje i tworzy proste bazy danych, 	<ul style="list-style-type: none"> projektuje zaawansowane relacyjne bazy danych, zarządza nimi, tworzy tabele pomostowe, formularze, kwerendy formułuje zapytania w języku SQL, stosując selekcję, sortowanie, projekcję oraz agregowanie danych, 	<ul style="list-style-type: none"> projektuje rozbudowane relacyjne bazy danych, zarządza nimi, wykorzystując zaawansowane narzędzia oraz klauzule języka SQL,
Algorytmy numeryczne				
<ul style="list-style-type: none"> omawia różnice między stałoprzecinkową a zmiennoprzecinkową reprezentacją liczb rzeczywistych w komputerze, wymienia rodzaje błędów w obliczeniach komputerowych, rozróżnia błąd względny i bezwzględny, 	<ul style="list-style-type: none"> wykorzystuje pliki tekstowe do wczytywania danych i zapisywania wyników, omawia algorytm znajdujący rozwinięcie binarne nieskracalnego ułamka właściwego, zapisuje liczby w postaci znormalizowanej, definiuje liczby pojedynczej precyzji i liczby podwójnej 	<ul style="list-style-type: none"> znajduje reprezentację liczby zapisanej w systemie dziesiętnym jako liczby pojedynczej i liczby podwójnej precyzji, świadomie używa typów <code>float</code> i <code>double</code> w zadaniach, stosuje schemat Hornera do zamiany liczby w systemie pozycyjnym o wybranej 	<ul style="list-style-type: none"> stosuje schemat Hornera do szybkiego podnoszenia do potęgi, implementuje algorytmy numeryczne: znajdowania miejsc zerowych funkcji oraz obliczania pierwiastka kwadratowego metodą bisekcji, obliczania pierwiastka kwadratowego metodą 	<ul style="list-style-type: none"> charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych, pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów

<ul style="list-style-type: none"> znajduje wartość wielomianu algorytmem naiwnym, wie, na czym polegają podstawowe metody obliczeń przybliżonych, wyjaśnia, co to jest fraktal, wskazuje przykłady struktur fraktalnych występujących w przyrodzie, 	<p>precyzji,</p> <ul style="list-style-type: none"> wykonuje działania na liczbach zmiennoprzecinkowych, wskazuje różnice między algorytmem stabilnym a algorytmem niestabilnym, znajduje pierwiastki równania kwadratowego algorytmem stabilnym i algorytmem niestabilnym, implementuje algorytm obliczający wartość wielomianu z zastosowaniem schematu Hornera, stosuje w algorytmach numerycznych metody: bisekcji, Newtona–Raphsona, trapezów, prostokątów, wyjaśnia sposób tworzenia fraktali, 	<p>podstawie na liczbę dziesiętną,</p>	<p>Newtona–Raphsona, implementuje w języku JavaScript algorytmy generujące fraktale danego stopnia,</p> <ul style="list-style-type: none"> zna algorytm Karpa–Rabina w programach wyszukujących wzorców w tekście, 	<p>informatycznych,</p> <ul style="list-style-type: none"> optymalizuje programy, szacuje ich efektywność
Zaawansowane algorytmy i techniki programistyczne				
<ul style="list-style-type: none"> wyjaśnia, na czym polega sortowanie leksykograficzne, wyszukuje wzorce w tekście algorytmem naiwnym, wskazuje różnice między kryptografią symetryczną i kryptografią asymetryczną, definiuje 	<ul style="list-style-type: none"> implementuje algorytm naiwny wyszukiwania wzorca w tekście, wyjaśnia, jak generuje się klucze publiczny i prywatny oraz szyfruje i deszyfruje informacje w algorytmie RSA, wyjaśnia, na czym polegają metoda zstępująca i metoda wstępująca, 	<ul style="list-style-type: none"> pisze program generujący klucz prywatny i klucz publiczny w algorytmie RSA, 	<ul style="list-style-type: none"> pisze programy szyfrujące i deszyfrujące informacje w algorytmie RSA, 	<ul style="list-style-type: none"> charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych, pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych,

<p>pojęcia klucz publiczny i klucz prywatny,</p> <ul style="list-style-type: none"> • wyjaśnia, do czego służy algorytm RSA, i wyróżnia główne etapy tego algorytmu (generowanie kluczy, szyfrowanie z kluczem publicznym oraz deszyfrowanie z kluczem prywatnym), 				<p>konkursów informatycznych,</p> <ul style="list-style-type: none"> • optymalizuje programy, szacuje ich efektywność
<p>Rozwiązywanie różnych problemów z wykorzystaniem komputera</p>				
<ul style="list-style-type: none"> • wyróżnia etapy pracy nad aplikacją internetową, rozróżnia technologie back-end i front-end, • definiuje pojęcie robota, omawia funkcje wybranych robotów i ich budowę, • rozróżnia pojęcia webcast, webinarium, screencast i podcast, • definiuje pojęcie grafiki informacyjnej, wymienia przykłady grafiki narracyjnej i wizualizacji danych, 	<ul style="list-style-type: none"> • wyjaśnia, na czym polega praca nad aplikacją internetową, instaluje i konfiguruje niezbędne oprogramowanie, przygotowuje bazę danych na potrzeby projektu, • opracowuje treści internetowe, korzystając z narzędzi graficznych i multimedialnych, dbając o identyfikację wizualną, • projektuje proste poprawne infografiki zawierające uporządkowane informacje, 	<ul style="list-style-type: none"> • tworzy podcasty i publikacje wideo, stosując elementy przyciągające uwagę użytkowników, montuje materiały, wykorzystując specjalistyczne oprogramowanie, • projektuje infografiki, umiejętnie stosując tekst i obraz, wykazuje się przy tym znajomością doboru barw i funkcji koloru, zwraca uwagę na dostosowanie treści do odbiorców, 	<ul style="list-style-type: none"> • tworzy interesujące podcasty i publikacje wideo, dba o właściwy format plików, stosuje kompresję, stosuje zasady pracy z kamerą i mikrofonem, • tworzy infografiki dostosowane do odbiorców, wykazując się dużymi umiejętnościami korzystania z narzędzi graficznych, 	<ul style="list-style-type: none"> • tworzy rozwinięte responsywne aplikacje internetowe wymagające dużego nakładu pracy i znajomości nowoczesnych technologii, • tworzy infografiki, korzystając z zaawansowanych narzędzi graficznych, • tworzy podcasty i publikacje wideo wymagające znajomości zaawansowanych narzędzi i dużego nakładu pracy,