

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów

opracowany na podstawie

Zawężonej podstawy programowej 2024

oraz materiału edukacyjnego:

Grażyna Koba, *Informatyka 1-3. Materiał edukacyjny dla szkół ponadpodstawowych. Zakres rozszerzony*

dostępnego na stronie internetowej ir.migra.pl

Autor: Robert Bednarz

Wymagania uwzględniają zmiany wynikające z zawężenia podstawy programowej dla szkoły ponadpodstawowej na podstawie rozporządzenia MEN z 2024 roku: *Rozporządzenie Ministra Edukacji zmieniającego rozporządzenie w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia.*

Spis treści

Komputer i sieci.....	4
Kompresja danych.....	4
Sztuczna inteligencja.....	4
Sieci komputerowe – funkcje urządzeń i protokoły.....	5
Arkusz kalkulacyjny.....	5
Stosowanie wybranych funkcji w arkuszu kalkulacyjnym.....	5
Grafika komputerowa.....	6
Tworzenie i edytowanie obrazów rastrowych.....	6
Praca z warstwami obrazu i animacje.....	7
Więcej o tworzeniu i edytowaniu grafiki wektorowej.....	8
Więcej na temat programowania w językach C++ i Python.....	8
Prezentacja algorytmu liniowego w wybranej notacji.....	8
Prezentacja algorytmu z warunkami w wybranej notacji.....	9
Prezentacja algorytmu iteracyjnego w wybranej notacji.....	10
Funkcje w wybranym języku programowania.....	11
Tablice w wybranym języku programowania.....	12
Stosowanie instrukcji iteracyjnych.....	12
Wybrane struktury danych.....	13
Przetwarzanie plików.....	14
Programowanie wybranych algorytmów.....	15
Iteracyjna realizacja wybranych algorytmów, w tym algorytmów wyszukiwania.....	15
Elementy analizy algorytmów.....	16
Sito Eratostenesa, rozkład liczby na czynniki pierwsze oraz znajdowanie w ciągu podciągów.....	17
Stosowanie metody zachłannej i metody połowienia.....	17
Szyfrowanie.....	18

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

Projekty.....	19
Praca w chmurze i zadania projektowe.....	19

Komputer i sieci

Kompresja danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
rozumie, na czym polega kompresja danych i w jakim celu się ją stosuje; zna wzór na obliczenie współczynnika kompresji	wymienia i omawia rodzaje kompresji – bezstratną i stratną; potrafi obliczyć współczynnik kompresji; zna pojęcie <i>przepływności</i>	omawia algorytmy kompresji stratnej; omawia algorytmy kompresji bezstratnej (statyczne, słownikowe)	dokonuje kompresji informacji, m.in. stosuje w przykładowych ćwiczeniach algorytmy kompresji bezstratnej; objaśnia różnice między kompresją stratną i bezstratną tekstów, obrazów, dźwięków i filmów	wykonuje samodzielnie zadania dla zainteresowanych

Sztuczna inteligencja				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wie czym jest sztuczna inteligencja; potrafi wymienić obszary zastosowania sztucznej inteligencji (SI);	umie przeanalizować prosty system SI w arkuszu kalkulacyjnym; potrafi wytłumaczyć schemat sieci neuronowej; konstruuje zapytania w systemie ChatGPT (lub podobnym); tworzy proste grafiki w systemie tworzenia obrazu	konstruuje rozwinięte zapytania w systemie ChatGPT (lub podobnym); samodzielnie korzysta z systemów tworzenia obrazów	wskazuje zalety i wady z korzystania z systemów wykorzystujących SI; korzysta z różnych systemów tworzenia obrazów	wyszukuje w Internecie inne systemy wykorzystujące sztuczną inteligencję oraz ocenia ich przydatność w różnych dziedzinach; prowadzi dyskusje o zastosowaniu SI i potrafi prawidłowo uargumentować swoje zdanie

Sieci komputerowe – funkcje urządzeń i protokoły				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
zna terminologię sieciową (<i>sieć komputerowa, zasoby sieciowe, klient, serwer, switch, router, brama, konfiguracja sieciowa, protokoły sieciowe</i>); potrafi zamienić oktety zapisane w systemie dziesiętnym na system	potrafi wyznaczyć adresy hostów i maskę w podsieci; przedstawia strukturę adresu domenowego; zna podstawowe funkcje urządzeń sieciowych; wie do czego służy szafa rackowa	opisuje podstawowe funkcje urządzeń i protokoły stosowane w przepływie informacji, w tym DNS, DHCP; potrafi odnaleźć w systemie Windows plik <i>hosts</i> oraz dokonać jego modyfikacji; przeprowadza konfigurację	określa ustawienia sieciowe danego komputera i jego lokalizację w sieci (adres sieciowy, podsieci IPv4, protokoły DHCP, DNS i TCP, określa ustawienia sieciowe komputera); zna standardowe numery portów przypisane wybranym	przygotowuje prezentację multimedialną z opisem podstawowych protokołów stosowanych w przepływie informacji i w zarządzaniu siecią; samodzielnie konfiguruje urządzenia sieciowe; zabezpiecza urządzenia

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

dwójkowy		routera wg instrukcji; potrafi dokonać montażu urządzeń w szafie rackowej	usługom; analizuje wydajność sieci komputerowej w tym bezprzewodowej korzystając z odpowiedniej do tego celu aplikacji	sieciowe przed nieuprawnionym dostępem; samodzielnie wykonuje zadania dla zainteresowanych
----------	--	--	---	---

Arkusz kalkulacyjny

Stosowanie wybranych funkcji w arkuszu kalkulacyjnym				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
analizuje i realizuje algorytm liniowy w arkuszu kalkulacyjnym, stosując odpowiednie funkcje zna zastosowanie funkcji JEŻELI w arkuszu kalkulacyjnym; wymienia grupy funkcji arkusza kalkulacyjnego np. statystyczne, matematyczne, tekstowe; wie, czym są zależności funkcyjne w arkuszu kalkulacyjnym	potrafi podać i omówić przykłady prostych algorytmów liniowych; zapisuje prostą sytuację warunkową w arkuszu kalkulacyjnym; tworzy formułę w arkuszu kalkulacyjnym sprawdzając czy wprowadzona liczba jest liczbą pierwszą; stosuje wybrane funkcje arkusza (matematyczne, logiczne, statystyczne, tekstowe) do realizacji algorytmów; analizuje przykłady zastosowania funkcji	analizuje realizację algorytmu z warunkami w arkuszu kalkulacyjnym; stosuje i analizuje formuły, sprawdzające czy dana liczba jest liczbą pierwszą; wymienia przykłady fraktali jak: śnieżynka Kocha, dywan i trójkąt Sierpińskiego, paprotka Bransleya; umie przeprowadzić kilka kroków „gry w chaos”; stosuje w zadaniach funkcje SUMA.JEŻELI i SILNIA, PRAWY,FRAGMENT.TEKST U, DŁ w arkuszu	zapisuje warunki zagnieżdżone w arkuszu kalkulacyjnym; dobiera funkcje arkusza kalkulacyjnego w zależności od rozwiązywanego problemu, w tym rodzaju danych; samodzielnie szkicuje trójkąt Sierpińskiego; omawia algorytm „gry w chaos” w arkuszu kalkulacyjnym; samodzielnie realizuje ćwiczenia z wykorzystaniem wybranych wzorów matematycznych, np. silni i	stosuje zaawansowane funkcje arkusza do realizacji algorytmów liniowych, z warunkami i iteracyjnymi i rysowania fraktali; samodzielnie analizuje algorytm konstruowania paprotki Barnsleya; samodzielnie tworzy arkusze kalkulacyjne i stosuje poznane funkcje zgodnie z ich przeznaczeniem; wykonuje zadania dla zainteresowanych

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

	SUMA, JEŻELI, SILNIA, PRAWY, FRAGMENT, TEKST U, DŁ i stosuje je w zadaniach	kalkulacyjnym; tworzy formuły na obliczenie funkcji liniowej, kwadratowej, trygonometrycznej i logarytmicznej i tworzy odpowiednio wykresy tych funkcji	symbolu Newtona; wykorzystuje w rozwiązaniach funkcje matematyczne, tekstowe i statystyczne i sprawdza poprawność rozwiązania	
--	---	--	--	--

Grafika komputerowa

Tworzenie i edytowanie obrazów rastrowych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
zna podstawowe możliwości programu do edycji obrazu rastrowego (programu GIMP); potrafi utworzyć prosty rysunek, stosując różne narzędzia malarskie i korekcyjne oraz wybrać odpowiedni tryb ich pracy	wykonuje operacje na obszarach selekcji, m.in.: wycinanie, kopiowanie, kadrowanie, obrysowywanie; stosuje poznane możliwości programu GIMP do edycji zdjęć	zna przeznaczenie i odpowiednio stosuje wybrane klawisze modyfikujące (i kombinacje klawiszy); potrafi wykonać proste modyfikacje barw: zmianę jasności, odwracanie kolorów, zmianę nasycenia, odcieni kolorów	zna pojęcia: <i>histogram</i> , <i>krzywa barw</i> ; tworzy własny pędzel; wykorzystuje tryb Kolor ; operuje na kanałach barw; używa krzywej jasności i funkcji histogramu; korzysta z narzędzia klonowania; w razie konieczności wyszukuje informacje w Pomocy	samodzielnie zapoznaje się z dodatkowymi możliwościami programu GIMP, w tym opcjami tworzenia pędzli; rozwiązuje trudniejsze zadanie, korzystając z narzędzia klonowania i innych; wyszukuje w Internecie informacje na temat tworzenia zdjęć panoramicznych; uczestniczy w konkursach dotyczących grafiki komputerowej

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

Praca z warstwami obrazu i animacje				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
poprawia jakość zdjęcia, stosując wybrane filtry; wie, na czym polega praca z warstwami	wykonuje przekształcenia obrazu (obroty, odbicia), tworząc obrazy w grafice rastrowej; stosuje filtry, m.in. poprawia ostrość obrazu, wykonuje proste fotomontaże, korzystając z warstw	stosuje filtry, m.in. wykonuje efekt zamglenia, usuwa szumy; wykonuje fotomontaże, korzystając z kilku warstw; wie, czym jest oderwane zaznaczenie i jak zakotwiczyć warstwę; przygotowuje animacje składające się z kilkunastu klatek, stosując pracę na warstwach; zapisuje animację w formacie GIF	samodzielnie tworzy rysunki, powiela warstwy i stosuje możliwości przekształceń, aby utworzyć kolejną klatkę animacji; wykorzystując dodatkowe możliwości programu GIMP, rozwiązuje różne zadania, edytując obrazy, tworząc fotomontaże korzystając z poznanych możliwości programu GIMP, projektuje dwuwymiarową wizualizację lub animację	wyszukuje w Internecie dodatkowe informacje na temat korzystania z filtrów w programie GIMP; wymyśla temat fotomontażu i przygotowuje animacje według własnych pomysłów, korzystając z różnych możliwości programu do tworzenia animacji; projektuje dwuwymiarową wizualizację lub animację, wykorzystując dodatkowe możliwości programu GIMP

Więcej o tworzeniu i edytowaniu grafiki wektorowej				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

zna cechy grafiki wektorowej; wie, czym różni się grafika wektorowa od grafiki rastrowej; potrafi wymienić nazwy programów do tworzenia grafiki wektorowej; wyjaśnia czym jest ścieżka; tworzy proste kształty; zapisuje plik do podanego formatu	potrafi korzystać z narzędzi <i>Kaligrafia, Narzędzie węzłów</i>; używa podstawowych skrótów klawiszowych; stosuje właściwe formaty do zapisu plików graficznych; wie, czym jest efekt ścieżki; wymienia znane operacje logiczne	korzystając z poznanych narzędzi, tworzy rysunek według podanej specyfikacji; zmienia kształty korzystając z tzw. węzłów edycyjnych; zamienia kształt w ścieżkę; stosuje efekty ścieżki do zmiany wyglądu obiektu; tworzy projekt graficzny według podanych założeń	samodzielnie modyfikuje kształt i zmienia jego właściwości; potrafi łączyć działanie kilku narzędzi w tworzeniu i modyfikacji obiektów; wykorzystuje efekty przekształceń do uzyskania kształtów według swojego pomysłu; wykonuje na obiektach operacje logiczne; samodzielnie tworzy projekt graficzny	tworzy i edytuje obrazy wektorowe z wykorzystaniem zaawansowanych funkcji programu; stosuje wybrane narzędzia edycji obrazu i narzędzia pomocnicze, m.in. wybrane operacje na ścieżkach, klonowanie i inne możliwości zaawansowanego modyfikowania obiektów; wykorzystuje program Inkscape i Tinkercad do drukowania modelu 3D
--	---	--	--	---

Programowanie w języku Python i C++

Prezentacja algorytmu liniowego w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
zna sposoby prezentacji algorytmów; testuje działanie algorytmu liniowego zapisanego w postaci listy kroków; zapisuje prosty algorytm liniowy w wybranym języku programowania	zna pojęcie specyfikacji zadania i potrafi zapisać specyfikację zadania; zna i stosuje zasady tworzenia listy kroków; pisze programy komputerowe realizujące dane algorytmy na podstawie ich list kroków	przedstawia dokładną specyfikację dowolnego zadania; objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych; realizuje przykładowy algorytm liniowy w wybranym	potrafi przeprowadzić szczegółową analizę poprawności listy kroków; analizuje działanie algorytmu dla przykładowych danych; potrafi wskazać i poprawić błędy w liście kroków; samodzielnie pisze program realizujący algorytm liniowy	zapisuje specyfikację zadania i przedstawia w postaci listy kroków rozwiązanie trudniejszego zadania; w języku programowania (C++ lub Python) pisze trudniejsze programy realizujące algorytmy przedstawione w postaci list kroków;

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

		języku programowania na podstawie ich list kroków; testuje program dla różnych danych		uczestniczy w konkursach/olimpiadach informatycznych
--	--	--	--	--

Prezentacja algorytmu z warunkami w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
określa sytuacje warunkowe; podaje przykłady zadań, w których występują sytuacje warunkowe; potrafi zapisać w liście kroków nietrudny algorytm z warunkiem prostym	analizuje działanie algorytmu z warunkami; zna i stosuje zasady tworzenia listy kroków algorytmu z warunkami; pisze program w wybranym języku programowania realizujący algorytm z warunkami prostymi; zna i stosuje instrukcję warunkową	przedstawia algorytm z warunkami w postaci listy kroków; analizuje listę kroków i schemat blokowy algorytmu z warunkami, testując go dla wybranych danych; potrafi zapisać warunek złożony; korzystając z przykładu, zapisuje w postaci programu algorytm z warunkami złożonymi; zna i omawia warunek istnienia trójkąta	zapisuje algorytmy z pętlą zagnieżdżoną; testuje działanie algorytmu z warunkami zagnieżdżonymi zapisanego w postaci listy kroków na wybranych przykładach danych; zapisuje w postaci programu algorytm z warunkami zagnieżdżonymi	wymyśla samodzielnie problem z warunkami zagnieżdżonymi, zapisuje jego specyfikację, listę kroków i program w wybranym języku programowania; korzystając z dodatkowych źródeł, znajduje inny, niż podany w podręczniku, sposób sprawdzenia, czy z danych trzech odcinków można zbudować trójkąt i zapisuje ten algorytm w postaci programu komputerowego; w wybranym języku programowania pisze trudniejsze programy realizujące algorytmy z warunkami (w tym zagnieżdżonymi) przedstawione w postaci list kroków

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

Prezentacja algorytmu iteracyjnego w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
analizuje listę kroków algorytmu iteracyjnego, testując go dla wybranych danych	zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego; podaje przykłady algorytmów iteracyjnych; tworzy algorytm z warunkiem prostym i pętlą; testuje rozwiązanie dla wybranych danych; zna i stosuje instrukcję iteracyjną for w wybranym języku programowania; zapisuje prosty algorytm iteracyjny w postaci programu w wybranym języku programowania	analizuje algorytmy, w których występują powtórzenia (iteracje); ocenia zgodność algorytmu ze specyfikacją; analizuje listę kroków algorytmu z pętlą zagnieżdżoną, testując go dla wybranych danych; zapisuje algorytm iteracyjny w postaci programu w wybranym języku programowania	zapisuje w postaci listy kroków algorytmów z warunkami i iteracyjne; pisze listę kroków algorytmu z pętlą zagnieżdżoną; ocenia zgodność algorytmu ze specyfikacją problemu; wie, kiedy należy zastosować pętlę zagnieżdżoną; zapisuje w postaci programu wybrany algorytm z pętlą zagnieżdżoną; testuje program dla wybranych danych	przestrzega zasad zapisu algorytmów w zadanej postaci (notacji); stosuje listy kroków w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia; wskazuje podobieństwa i różnice dotyczące działania instrukcji warunkowych i iteracyjnej for w różnych językach programowania; zapisuje trudniejsze algorytmy iteracyjne w wybranym języku programowania; zapisuje programy w czytelnej postaci – stosuje wcięcia, komentarze; uczestniczy w konkursach/olimpiadach informatycznych

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

Funkcje w wybranym języku programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wie, czym są podprogramy; w wybranym języku programowania definiuje prostą funkcję niezwracającą wartości bez parametrów. wie, jak wywołać funkcję niezwracającą wartości bez parametrów w programie; wywołuje taką funkcję w programie	wie, na czym polega programowanie strukturalne; pisze programy, stosując funkcje; definiuje funkcje niezwracające wartości bez parametrów i z parametrami; stosuje te funkcje w programach; wyjaśnia pojęcia zmienna lokalna i zmienna globalna	wymienia modele programowania; rozumie i stosuje zasady programowania strukturalnego; wyjaśnia, czym są parametry funkcji; wyjaśnia, czym jest funkcja zwracająca wartość, porównuje do funkcji niezwracającej wartości i wskazuje różnice; wie, na czym polega wywołanie funkcji z parametrami i wywołuje taką funkcję w programach; definiuje funkcje zwracające wartości bez parametrów i z parametrami oraz stosuje je w programach; wie, co to jest zasięg zmiennej i jakie ma znaczenie; deklaruje odpowiednio zmienne lokalne i globalne w programach	omawia modele programowania; rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą; wyjaśnia, czym różni się programowanie zstępujące od wstępującego; definiuje funkcje niezwracające wartości i zwracające wartość bez parametrów i z parametrami oraz stosuje je w programach; wyjaśnia, na czym polega przesłanianie zmiennych globalnych; wyjaśnia, na czym polega przesłanianie parametrów funkcji; modyfikuje programy, stosuje zmienne globalne i lokalne, objaśnia otrzymane wyniki	porównuje modele programowania, wskazując różnice; sprawnie definiuje i stosuje funkcje niezwracające wartości i zwracające wartość bez parametrów i z parametrami w programach; rozwiązuje przykładowe zadania z matury i olimpiady informatycznej; potrafi, na przykładzie programu utworzonego według własnego pomysłu, wyjaśnić różnice w stosowaniu zmiennych lokalnych i globalnych, omówić zasięg zmiennych i przesłanianie zmiennych; potrafi, na przykładzie programu utworzonego według własnego pomysłu, wyjaśnić przesłanianie parametrów funkcji

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

Tablice w wybranym języku programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
analizuje i omawia gotowe przykłady programów, w których są użyte tablice lub listy; potrafi wyjaśnić, dlaczego do rozwiązania niektórych zadań należy użyć tablic (list)	zna pojęcia: <i>tablica</i>, <i>lista</i>, <i>zmienna indeksowana</i>; na bazie przykładów, deklaruje tablicę i/lub listę, wczytuje i wyprowadza elementy tablicy i/lub listy, definiując odpowiednie funkcje w wybranym języku programowania	rozdziela struktury danych proste i złożone oraz podaje ich przykłady; wczytuje i wyprowadza elementy tablicy i/lub listy; definiuje odpowiednie funkcje; wie, jak odwołać się do elementu tablicy i/lub listy; potrafi zastosować tablicę (listę) w zadaniach oraz modyfikować program, znaleźć błędy i je poprawić	rozumie, na czym polega dobór struktur danych do algorytmu i tworzy programy, dobierając odpowiednie struktury danych do programu; definiuje odpowiednie funkcje; definiuje różne listy (w tym dwuwymiarowe) w języku Python; tworzy programy, dobierając odpowiednie struktury danych do programu	omawia podobieństwa i różnice w definiowaniu tablic w języku C++ i list w języku Python; stosuje w programach listy jednowymiarowe i dwuwymiarowe, odpowiednio dobierając określoną strukturę danych (tu: rodzaj listy) do algorytmu; rozwiązuje przykładowe zadania z konkursów i olimpiady informatycznej, w których należy zastosować tablice i/lub listy i funkcje; dobiera najlepszy algorytm i odpowiednie struktury danych do rozwiązania postawionego problemu

Stosowanie instrukcji iteracyjnych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
analizuje i omawia działanie gotowych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli for; analizuje i omawia działanie gotowych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli while	zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego, podaje ich przykłady; zna postać i działanie instrukcji iteracyjnej while w języku Python i stosuje ją w tworzonych programach komputerowych	zna sposoby zakończenia iteracji, określa kroki iteracji; stosuje instrukcję while w programach komputerowych; zna różne sposoby określania liczby iteracji w języku Python, np. poprzez powtarzanie poleceń zawartych wewnątrz pętli dla konkretnych wartości lub poprzez podanie liczby powtórzeń z zastosowaniem funkcji range()	rozumie różnicę pomiędzy instrukcją for a instrukcją while w wybranym języku programowania; modyfikuje programy, zamieniając pętlę for na while i odwrotnie; ocenia program po zmianie (styl, czytelność); pisze programy, stosując poznane instrukcje iteracyjne; dobiera odpowiednią technikę algorytmiczną i odpowiednie struktury danych do rozwiązywanego problemu	potrafi samodzielnie zastosować odpowiedni rodzaj instrukcji pętli w tworzonym programie; potrafi samodzielnie dobrać odpowiednią instrukcję while lub do...while; omawia podobieństwa i różnice w działaniu wszystkich omówionych instrukcji pętli w języku C++; sprawnie pisze trudniejsze programy, stosując poznane instrukcje iteracyjne; wymyśla samodzielnie problem iteracyjny, formułuje zadanie, pisze jego specyfikację i listę kroków oraz zapisuje w wybranym języku programowania

Wybrane struktury danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wie, czym są <i>typ danych</i> i <i>format danych</i>; wymienia podstawowe proste i złożone typy danych; zna sposoby wprowadzania i wyprowadzania danych w języku Python; potrafi wytłumaczyć, czym są dane losowe; wyjaśnia, czym jest sprawdzanie poprawności danych w programie komputerowym	korzysta z wybranych zbiorów i podzbiorów typów w języku Python; potrafi określić do jakiego typu należą zadeklarowane zmienne w kodzie źródłowym; zna wybrane elementy formatowania wykorzystywane przez funkcję <code>print()</code>; wymienia funkcje do generowania danych losowych w języku Python; zna funkcje standardowe, które można użyć do sprawdzenia poprawności danych	stosuje funkcje <code>getsizeof()</code> i <code>type()</code> w języku Python; używa wbudowanych funkcji; objaśnia sposób formatowania poszczególnych wyników wykorzystanych przez funkcję <code>print()</code>; wie, do czego służą f-łańcuchy w języku Python; analizuje przykłady stosowania danych losowych; analizuje przykłady sprawdzające poprawność danych liczbowych w języku Python	oblicza i wyświetla wielkości pamięci zajmowane przez zmienne wybranych typów; oblicza i wypisuje minimalne i maksymalne wartości dające się zapamiętać w zmiennej; samodzielnie tworzy program z wykorzystaniem formatowania wyprowadzanych wyników w języku Python; korzysta z f-łańcuchów w języku Python; tworzy program symulacji rzutów sześcienną kostką do gry; tworzy programy sprawdzające poprawność wprowadzonych danych liczbowych w języku Python	wykonuje zadania dla zainteresowanych; potrafi wybrać odpowiednie struktury danych do wykonania zadań; samodzielnie bez pomocy nauczyciela rozwiązuje ćwiczenia i zadania; przygotowuje program według własnego pomysłu wykorzystujący poznane struktury danych

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

Przetwarzanie plików				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wie, czym nazywamy strumień danych; wie, czym jest kod ASCII, zna liczbę pozycji w tabeli; wymienia nazwy innych stron kodowych; omawia przykłady błędów w programach komputerowych	zna cel wykorzystania metody <code>seek()</code> w języku Python; zna konstrukcje do obsługi błędów w języku Python	wykorzystuje klasy i funkcje (zależnie od języka programowania) do odczytu i zapisu danych; analizuje gotowy program odczytujący dane z pliku i wyświetlający je na ekranie; analizuje gotowy program zapisujący dane do pliku; stosuje funkcje do obsługi błędów w języku Python	używa funkcji obsługi plików w programach komputerowych; analizuje gotowy program z obsługą błędów; samodzielnie modyfikuje programy zgodnie z treścią ćwiczeń	wykonuje zadania dla zainteresowanych; potrafi wybrać odpowiednie struktury danych do wykonania zadań; samodzielnie bez pomocy nauczyciela rozwiązuje ćwiczenia i zadania

Programowanie wybranych algorytmów

Iteracyjna realizacja wybranych algorytmów, w tym algorytmów wyszukiwania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
potrafi odróżnić algorytm liniowy od algorytmu iteracyjnego; omawia algorytm znajdowania elementu najmniejszego oraz analizuje listę kroków;	zna przykładowe algorytmy na liczbach naturalnych; wie, na czym polega metoda wyszukiwania liniowego i przez połowienie; zapisuje algorytm znajdowania	potrafi omówić algorytm naiwny i optymalny jednoczesnego znajdowania największego i najmniejszego elementu w zbiorze; zna iteracyjną postać algorytmu Euklidesa z resztą z dzielenia,	określa liczbę porównań w algorytmie naiwnym i optymalnym znajdowania największego i najmniejszego elementu w zbiorze, porównuje otrzymane wyniki; pisze listę kroków algorytmu	podaje przykłady problemów, w których można zastosować wyszukiwanie liniowe lub przez połowienie; pisze trudniejsze programy komputerowe, w których wykorzystuje poznane

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

analizuje i omawia gotową listę kroków algorytmu Euklidesa w jednej z wersji, testuje algorytm na podstawie listy	najmniejszego (największego) elementu w postaci programu; zna iteracyjną postać algorytmu Euklidesa w wersji z odejmowaniem; potrafi napisać listy kroków algorytmu Euklidesa w wersji z odejmowaniem; zna metodę „dziel i zwyciężaj”; zapisuje algorytm Euklidesa w wersji z odejmowaniem w postaci programu w wybranym języku programowania	pisze listę kroków tego algorytmu; zapisuje algorytm Euklidesa w wersji z resztą z dzielenia w postaci programu w wybranym języku programowania; wyjaśnia na przykładzie różnicę między wersją algorytmu Euklidesa z odejmowaniem a wersją z resztą z dzielenia	jednoczesnego znajdowania minimum i maksimum z wykorzystaniem metody dziel i zwyciężaj, zapisuje ten algorytm w postaci programu w języku Python; omawia zastosowanie schematu Hornera do obliczania wartości wielomianu, pisze listę kroków tego algorytmu i pisze program realizujący algorytm obliczania wartości wielomianu według schematu Hornera; programując w/w algorytmy, definiuje odpowiednie funkcje, dobiera struktury danych, dba o stosowanie podstawowych zasad programowania	algorytmy; korzystając z dodatkowych źródeł, wyszukuje informacje o zastosowaniu metody „dziel i zwyciężaj” oraz pisze program według własnego pomysłu pokazujący zastosowanie tej metody; samodzielnie zapoznaje się ze schematem Hornera i zapisuje go w postaci programu, dobierając poprawne struktury danych; korzystając z dodatkowych źródeł, omawia przykłady zastosowań algorytmu Euklidesa i poznaje trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, algorytm znajdowania liczb bliźniaczych, potrafi zapisać je w języku programowania
--	--	--	---	---

Elementy analizy algorytmów				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wymienia własności algorytmów; potrafi przeanalizować przebieg prostego algorytmu zapisanego w postaci listy	zna i omawia własności algorytmów; wie, kiedy algorytm jest poprawny; potrafi przeanalizować	wie, jak sprawdzić, czy algorytm jest skończony; potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją;	wie, kiedy algorytm jest skończony; potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy	potrafi samodzielnie ocenić poprawność i skończoność wybranych algorytmów; potrafi samodzielnie wymyśleć zadanie (problem), napisać

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

kroków dla przykładowych danych i ocenić w ten sposób jego poprawność	przebieg algorytmu (np. obliczania silni) zapisanego w postaci listy kroków dla przykładowych danych i ocenić w ten sposób jego poprawność; analizuje program wybranego algorytmu (np. obliczania silni) i ocenia jego poprawność	określa liczbę prostych działań zawartych w algorytmie; określa liczbę prostych działań zawartych w algorytmie; potrafi poprawić program, który jest niepoprawny	kroków dla przykładowych danych i ocenić w ten sposób jego skończoność; analizuje program realizujący wybrany algorytm i ocenia jego skończoność; modyfikuje program, aby działał poprawnie; sprawdza poznane własności algorytmów, rozwiązując zadania, m.in. uzasadnia skończoność algorytmu znajdowania największego wspólnego dzielnika (NWD) dwóch liczb naturalnych; oblicza liczbę operacji porównania w algorytmie wyboru minimum z tablicy zawierającej n losowo uporządkowanych liczb	specyfikację zadania, listę kroków i program realizujący to zadanie; korzysta samodzielnie z dodatkowej literatury fachowej; poznane dodatkowe możliwości wybranego języka programowania (np. standardowe funkcje) i stosuje w programach; rozwiązuje przykładowe zadania z olimpiady informatycznej
---	---	---	---	--

Sito Eratostenesa, rozkład liczby na czynniki pierwsze oraz znajdowanie w ciągu podciągów				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

zna i potrafi wytłumaczyć algorytm badania pierwszości liczby metodą sita Eratostenesa; omawia rozkład liczby na czynniki pierwsze; zna definicję podciagu, podciagu spójnego i podaje przykłady	wskazuje i nazywa konstrukcje programistyczne wykorzystane w przykładach programów generowania liczb pierwszych metodą sita Eratostenesa; analizuje opis algorytmu rozkładu liczby na czynniki pierwsze; podaje przykłady podciągów spójnych; zna definicję liczby doskonałej	analizuje realizację algorytmu generowania liczb pierwszych metodą sita Eratostenesa w języku Python; potrafi wytłumaczyć korzystając z listy kroków algorytm wyszukania liczby doskonałej; planuje znajdowanie najdłuższego spójnego podciagu niemalejącego; planuje znajdowanie spójnego podciagu o największej sumie	tworzy programy komputerowe w języku Python wyznaczające liczby pierwsze metodą sita Eratostenesa; tworzy programy komputerowe rozkładu liczby na czynniki pierwsze w języku Python; samodzielnie sprawdza, czy dana liczba jest liczbą doskonałą; pisze i modyfikuje programy znajdujące najdłuższy spójny podciąg niemalejący oraz o największej sumie	wykonuje zadania dla zainteresowanych, w tym znajdowanie liczb bliźniaczych; potrafi wybrać odpowiednie struktury danych do wykonania zadań; przygotowuje program według własnego pomysłu wykorzystujący poznane struktury danych
---	---	---	--	--

Stosowanie metody zachłannej i metody połowienia				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
omawia definicję algorytmu zachłannego; podaje przykłady wykorzystania algorytmu zachłannego	tłumaczy algorytm wydawania reszty metodą zachłanną; wie, na czym polega algorytm połowienia i demonstruje ten algorytm na przykładzie	analizuje na konkretnym przykładzie algorytm wydawania reszty metodą zachłanną; analizuje funkcje realizujące algorytm wydawania reszty metodą zachłanną i wyszukiwania elementu w zbiorze uporządkowanym	analizuje programy realizujące algorytm zachłanny wydawania reszty i wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia w języku Python; objaśnia oraz porównuje podstawowe metody i	pisze program realizujący algorytm wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia w języku Python; znajduje błędy w wykonanym programie i je poprawia wykonuje zadania dla

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

		metodą połowienia	techniki algorytmiczne oraz struktury danych zastosowane w w/w/ programach	zainteresowanych; przygotowuje program według własnego pomysłu wykorzystujący poznane techniki programistyczne
--	--	-------------------	--	---

Szyfrowanie				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wie, czym jest szyfrowanie z kluczem; zna różnice między algorytmem szyfrowania symetrycznego a algorytmem szyfrowania asymetrycznego	omawia schemat szyfrowania z kluczem; potrafi scharakteryzować szyfr Vigenère'a; wymienia zastosowanie szyfrowania w życiu codziennym	analizuje przykład stosowania szyfru Vigenère'a; określa wady i zalety szyfrowania z kluczem jawnym	dokonuje szyfrowania szyfrem Vigenere'a; opisuje wykorzystanie algorytmów szyfrowania w podpisie elektronicznym; samodzielnie wykonuje zadania	przygotowuje prezentację multimedialną dotyczącą algorytmu RSA; wykonuje zadania dla zainteresowanych

Projekty

Praca w chmurze i zadania projektowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
zna etapy pracy nad projektem i bierze udział	planuje temat projektu; bierze aktywny udział w pracy	korzystając z chmury, potrafi udostępnić pliki, linki do	korzysta z oprogramowania dostępnego w chmurze,	sprawnie posługuje się środowiskiem przeznaczonym

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.

w pracy grupowej jako członek zespołu; uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje	grupowej jako członek zespołu, gromadząc i selekcionując materiały do projektu; bierze udział w testowaniu projektu; uczestniczy w przygotowaniu dokumentacji projektu, korzystając z edytora tekstu, ewentualnie z wybranego szablonu	folderu, umożliwić współdzielenie danego folderu; realizuje projekt na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania, przygotowuje dokumentację projektu; współpracuje w grupie, wykonując projekt na temat projektowania zakupu nowego zestawu komputerowego oraz oprogramowania dla ucznia szkoły ponadpodstawowej oraz wpływu trendów w historycznym rozwoju pojęć i metod informatyki oraz technologii na możliwości rozwoju grafiki komputerowej	m.in. tworzy dokumenty w edytorze tekstu i umieszcza w chmurze, współdzieli je z innymi współużytkownikami; przygotowuje wybrane zadanie szczegółowe zgodnie z etapami przygotowania projektu; gromadzi materiały i inne pomoce, opracowuje dokumentację projektu, wykorzystując m.in. możliwości pracy w chmurze; udostępnia pliki, linki do folderu; wyszukuje informacje na e-platformach do e-nauczania; prezentuje wykonane zadanie projektowe i inicjuje dyskusje	do współpracy i realizacji projektów zespołowych, w tym środowiskiem w chmurze; potrafi pełnić funkcję koordynatora grupy; koordynuje wykonywanie zadań szczegółowych na poszczególnych etapach; zarządza folderami (współdzieleniem) i pracą nad dokumentami, w tym dokumentacją projektu; ustala sposób prezentacji projektu i wyznacza osobę (osoby) do prezentacji; w miarę możliwości współtworzy zasoby udostępniane na platformach do e-nauczania; przygotowuje projekt na wybrany przez siebie temat
--	---	--	---	---

Wymagania na poszczególne oceny szkolne dla klas 1-4 z poszczególnych działów.