

Wymagania na poszczególne oceny z informatyki dla klasy 8

Uwaga! Wymagania na każdą ocenę wyższą niż **dopuszczająca** obejmują również wymagania na oceny **niższe**.

Wymagania konieczne (ocena dopuszczająca) Uczeń:	Wymagania podstawowe (ocena dostateczna) Uczeń:	Wymagania rozszerzające (ocena dobra) Uczeń:	Wymagania dopełniające (ocena bardzo dobra) Uczeń:	Wymagania wykraczające (ocena celująca) Uczeń:
DZIAŁ 1. Arkusz kalkulacyjny				
<ul style="list-style-type: none"> omawia zastosowanie oraz budowę arkusza kalkulacyjnego określa adres komórki wprowadza dane różnego rodzaju do komórek arkusza kalkulacyjnego formatuje zawartość komórek (wyrównanie tekstu oraz wygląd czcionki) 	<ul style="list-style-type: none"> określa zasady wprowadzania danych do komórek arkusza kalkulacyjnego dodaje i usuwa wiersze oraz kolumny w tabeli 	<ul style="list-style-type: none"> tworzy proste formuły obliczeniowe wyjaśnia, czym jest adres względny 	<ul style="list-style-type: none"> kopiuje utworzone formuły obliczeniowe, wykorzystując adresowanie względne 	<ul style="list-style-type: none"> samodzielnie tworzy i kopiuje skomplikowane formuły obliczeniowe
<ul style="list-style-type: none"> rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym 	<ul style="list-style-type: none"> stosuje w arkuszu podstawowe funkcje: (SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora 	<ul style="list-style-type: none"> wykorzystuje funkcję JEŻELI do tworzenia algorytmów z warunkami w arkuszu kalkulacyjnym ustawia format danych komórki odpowiadający jej zawartości w formułach stosuje adresowanie względne, bezwzględne i mieszane 	<ul style="list-style-type: none"> korzysta z biblioteki funkcji, aby wyszukiwać potrzebne funkcje stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych 	<ul style="list-style-type: none"> stosuje zaawansowane funkcje arkusza w tabelach stworzonych na własne potrzeby
<ul style="list-style-type: none"> wstawia wykres do arkusza kalkulacyjnego 	<ul style="list-style-type: none"> omawia i modyfikuje poszczególne elementy wykresu 	<ul style="list-style-type: none"> dobiera odpowiedni wykres do rodzaju danych 	<ul style="list-style-type: none"> tworzy wykres dla więcej niż jednej serii danych 	<ul style="list-style-type: none"> tworzy rozbudowane wykresy dla wielu serii danych
<ul style="list-style-type: none"> korzysta z arkusza kalkulacyjnego w celu stworzenia kalkulacji wydatków 	<ul style="list-style-type: none"> zapisuje w tabeli arkusza kalkulacyjnego dane otrzymane z prostych doświadczeń i przedstawia je na wykresie 	<ul style="list-style-type: none"> sortuje oraz filtruje dane w arkuszu kalkulacyjnym 	<ul style="list-style-type: none"> tworzy prosty model (na przykładzie rzutu sześcienną kostką do gry) w arkuszu kalkulacyjnym 	<ul style="list-style-type: none"> przygotowuje rozbudowane arkusze kalkulacyjne korzysta z arkusza kalkulacyjnego do

			<ul style="list-style-type: none"> • stosuje filtry niestandardowe 	analizowania doświadczeń z innych przedmiotów
DZIAŁ 2. Programowanie w języku C++				
<ul style="list-style-type: none"> • definiuje pojęcia: algorytm, program, programowanie • podaje kilka sposobów przedstawienia algorytmu 	<ul style="list-style-type: none"> • wymienia różne sposoby przedstawienia algorytmu: opis słowny, schemat blokowy, lista kroków • poprawnie formułuje problem do rozwiązania • wyjaśnia różnice między interaktywnym a skryptowym trybem pracy • stosuje odpowiednie polecenie języka C++, aby wyświetlić tekst na ekranie • omawia różnice pomiędzy kodem źródłowym a kodem wynikowym • tłumaczy, czym jest środowisko programistyczne 	<ul style="list-style-type: none"> • wymienia przykładowe środowiska programistyczne • wyjaśnia, czym jest specyfikacja problemu • opisuje etapy rozwiązywania problemów • opisuje etapy powstawania programu komputerowego • zapisuje proste polecenia języka C++ 	<ul style="list-style-type: none"> • pisze proste programy w trybie skryptowym języka C++ 	<ul style="list-style-type: none"> • zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności
<ul style="list-style-type: none"> • tłumaczy, do czego używa się zmiennych w programach • pisze proste programy w trybie skryptowym języka C++ z wykorzystaniem zmiennych 	<ul style="list-style-type: none"> • wykonuje obliczenia w języku C++ • omawia działanie operatorów arytmetycznych • stosuje tablice w języku C++ oraz operatory logiczne 	<ul style="list-style-type: none"> • wykorzystuje instrukcję warunkową <code>if</code> oraz <code>if else</code> w programach • wykorzystuje iterację w konstruowanych algorytmach • wykorzystuje w programach instrukcję iteracyjną <code>for</code> • definiuje funkcje w języku C++ i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości 	<ul style="list-style-type: none"> • buduje złożone schematy blokowe służące do przedstawiania skomplikowanych algorytmów • konstruuje złożone sytuacje warunkowe (wiele warunków) w algorytmach • pisze programy zawierające instrukcje warunkowe, pętle oraz funkcje • wyjaśnia, jakie błędy zwraca interpreter • czyta kod źródłowy i opisuje jego działanie 	<ul style="list-style-type: none"> • pisze programy w języku C++ do rozwiązywanie zadań matematycznych • tworzy program składający się z kilku funkcji wywoływanych w programie głównym

<ul style="list-style-type: none"> • wyjaśnia działanie operatora modulo • wyjaśnia algorytm badania podzielności liczb 	<ul style="list-style-type: none"> • zapisuje w postaci listy kroków algorytm badania podzielności liczb naturalnych • wykorzystuje w programach instrukcję iteracyjną <code>while</code> 	<ul style="list-style-type: none"> • omawia algorytm Euklidesa w wersji z odejmowaniem i z dzieleniem – zapisuje go w wybranej postaci • wyjaśnia algorytm wyodrębniania cyfr danej liczby i zapisuje go w wybranej postaci 	<ul style="list-style-type: none"> • wyjaśnia różnice między instrukcją iteracyjną <code>while</code> a pętlą <code>for</code> • pisze programy obliczające NWD, stosując algorytm Euklidesa, oraz wypisujące cyfry danej liczby • wyjaśnia różnice między algorytmem Euklidesa w wersjach z odejmowaniem i z dzieleniem 	<ul style="list-style-type: none"> • pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby
<ul style="list-style-type: none"> • wyjaśnia potrzebę wyszukiwania informacji w zbiorze • określa różnice między wyszukiwaniem w zbiorach uporządkowanym i nieuporządkowanym • sprawdza działanie programów wyszukujących element w zbiorze 	<ul style="list-style-type: none"> • zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym, w tym elementu największego i najmniejszego • zapisuje algorytm wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia • implementuje grę w zgadywanie liczby 	<ul style="list-style-type: none"> • implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym • omawia funkcje zastosowane w realizacji algorytmu wyszukiwania metodą połowienia • implementuje algorytm wyszukiwania największej wartości w zbiorze 	<ul style="list-style-type: none"> • samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze metodą połowienia, w tym elementu największego i najmniejszego • implementuje algorytm wyszukiwania elementu w zbiorze metodą połowienia 	<ul style="list-style-type: none"> • samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania
<ul style="list-style-type: none"> • wyjaśnia potrzebę porządkowania danych • sprawdza działanie programu sortującego dla różnych danych 	<ul style="list-style-type: none"> • zapisuje w wybranej formie algorytm porządkowania metodami przez wybieranie oraz przez zliczanie • omawia implementację algorytmu sortowania przez wybieranie • stosuje pętle zagnieżdżone i wyjaśnia, jak działają 	<ul style="list-style-type: none"> • omawia implementację algorytmu sortowania przez zliczanie • omawia funkcje zastosowane w kodzie źródłowym algorytmów sortowania przez wybieranie oraz przez zliczanie 	<ul style="list-style-type: none"> • implementuje algorytmy porządkowania metodami przez wybieranie oraz przez zliczanie • wprowadza modyfikacje w implementacji algorytmów porządkowania przez wybieranie oraz przez zliczanie 	<ul style="list-style-type: none"> • samodzielnie modyfikuje i optymalizuje programy sortujące metodą przez wybieranie, metodą przez zliczanie
DZIAŁ 4. Projekty				
<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej 	<ul style="list-style-type: none"> • przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, przygotowuje zestawienia, drukuje wyniki 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, tworzy

wykonywać powierzone mu zadania o niewielkim stopniu trudności	<ul style="list-style-type: none"> wprowadza dane do zaprojektowanych tabel 	<ul style="list-style-type: none"> współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> zestawienia zawierające zaawansowane formuły, wykresy oraz elementy graficzne współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera
<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności testuje grę na różnych etapach współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> bierze udział w pracach nad wypracowaniem koncepcji gry współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> programuje wybrane funkcje i elementy gry opracowuje opis gry 	<ul style="list-style-type: none"> implementuje i optymalizuje kod źródłowy gry, korzystając z wypracowanych założeń 	<ul style="list-style-type: none"> rozbudowuje grę o nowe elementy współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera
<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności – znalezienie informacji w internecie, umieszczenie ich w chmurze 	<ul style="list-style-type: none"> współpracuje z innymi podczas pracy nad projektem analizuje zebrane dane tworzy projekt prezentacji multimedialnej 	<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania tworzy prezentację wg projektu zaakceptowanego przez zespół 	<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu analizuje i weryfikuje pod względem merytorycznym i technicznym przygotowaną prezentację 	<ul style="list-style-type: none"> współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera wzbogaca prezentację o elementy podnoszące jej walory estetyczne i merytoryczne
<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności bierze aktywny udział w dyskusji nad wyborem atrakcyjnego zawodu wymagającego kompetencji informatycznych 	<ul style="list-style-type: none"> gromadzi informacje dotyczące wybranych zawodów, umieszcza je w zaprojektowanych tabelach i dokumentach tekstowych 	<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu projektuje tabele do zapisywania informacji o zawodach weryfikuje i formatuje przygotowane dokumenty tekstowe 	<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu, weryfikuje opracowane treści i łączy wszystkie dokumenty w całość 	<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu, przyjmuje rolę lidera podczas dyskusji przyjmuje funkcję moderatora

Uczeń, który nie sprostał wymaganiom na ocenę dopuszczającą, otrzymuje ocenę niedostateczną.