

Wymagania na poszczególne oceny klasa 8

Wymagania na każdy stopień wyższy niż **dopuszczający** obejmują również wymagania na wszystkie stopnie niższe.

Wymagania na ocenę celującą obejmują stosowanie przyswojonych informacji i umiejętności w sytuacjach trudnych, złożonych i nietypowych.

Ocena			
Stopień dopuszczający Uczeń:	Stopień dostateczny Uczeń:	Stopień dobry Uczeń:	Stopień bardzo dobry Uczeń:
- 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) - rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym - wstawia wykres do arkusza kalkulacyjnego - korzysta z arkusza kalkulacyjnego w celu stworzenia kalkulacji wydatków - definiuje pojęcia: algorytm, program, programowanie - podaje kilka sposobów przedstawienia algorytmu - tłumaczy, do czego używa się zmiennych w programach - pisze proste programy w trybie skryptowym języka Python z wykorzystaniem zmiennych - wyjaśnia działanie operatora modulo - wyjaśnia algorytm badania podzielności liczb - wyjaśnia potrzebę wyszukiwania informacji	- określa zasady wprowadzania danych do komórek arkusza kalkulacyjnego - dodaje i usuwa wiersze oraz kolumny w tabeli - stosuje w arkuszu podstawowe funkcje: (SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora - omawia i modyfikuje poszczególne elementy wykresu - zapisuje w tabeli arkusza kalkulacyjnego dane otrzymane z prostych doświadczeń i przedstawia je na wykresie - wymienia różne sposoby przedstawienia algorytmu: opis słowny, lista kroków - poprawnie formułuje problem do rozwiązania - wyjaśnia różnice między interaktywnym a skryptowym trybem pracy - stosuje odpowiednie polecenie języka Python, aby wyświetlić tekst na ekranie - omawia różnice pomiędzy kodem źródłowym a kodem wynikowym - tłumaczy, czym jest środowisko	- tworzy proste formuły obliczeniowe - wyjaśnia, czym jest adres względny - 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 - dobiera odpowiedni wykres do rodzaju danych - sortuje oraz filtruje dane w arkuszu kalkulacyjnym - wymienia przykładowe środowiska programistyczne - wyjaśnia, czym jest specyfikacja problemu - opisuje etapy rozwiązywania problemów - opisuje etapy powstawania programu komputerowego - zapisuje proste polecenia języka Python - wykorzystuje instrukcję warunkową if oraz if else w programach - wykorzystuje iterację w konstruowanych	- kopiuje utworzone formuły obliczeniowe, wykorzystując adresowanie względne - korzysta z biblioteki funkcji, aby wyszukiwać potrzebne funkcje - stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych - tworzy wykres dla więcej niż jednej serii danych - tworzy prosty model (na przykładzie rzutu sześcienną kostką do gry) w arkuszu kalkulacyjnym - stosuje filtry niestandardowe - pisze proste programy w trybie skryptowym języka Python - 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 - wyjaśnia różnice między instrukcją iteracyjną while a pętlą for

w zbiorze • sprawdza działanie programów wyszukujących element w zbiorze • wyjaśnia potrzebę porządkowania danych • sprawdza działanie programu sortującego dla różnych danych • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, wykonując powierzone mu zadania o niewielkim stopniu trudności • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności • testuje grę na różnych etapach • współpracuje w grupie podczas pracy nad projektem	programistyczne • wykonuje obliczenia w języku Python • omawia działanie operatorów arytmetycznych • stosuje listy w języku Python oraz operatory logiczne • zapisuje w postaci listy kroków algorytm badania podzielności liczb naturalnych • wykorzystuje w programach instrukcję iteracyjną while • zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym, w tym elementu największego i najmniejszego • zapisuje w wybranej formie algorytm porządkowania metodą przez wybieranie • omawia implementację algorytmu sortowania przez wybieranie • stosuje pętle zagnieżdżone i wyjaśnia, jak działają • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej • wprowadza dane do zaprojektowanych tabel • bierze udział w pracach nad wypracowaniem koncepcji gry • współpracuje w grupie podczas pracy nad projektem	algorytmach • wykorzystuje w programach instrukcję iteracyjną for • definiuje funkcje w języku Python i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości • omawia algorytm Euklidesa i zapisuje go w wybranej postaci • wyjaśnia algorytm wyodrębniania cyfr danej liczby i zapisuje go w wybranej postaci • implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym • omawia funkcje zastosowane w kodzie źródłowym algorytmu sortowania przez wybieranie • przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy • współpracuje w grupie podczas pracy nad projektem • programuje wybrane funkcje i elementy gry • opracowuje opis gry	• pisze programy obliczające NWD, stosując algorytm Euklidesa, oraz wypisujące cyfry danej liczby • samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze • implementuje algorytm porządkowania metodą przez wybieranie • wprowadza modyfikacje w implementacji algorytmu porządkowania przez wybieranie • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, przygotowuje zestawienia, drukuje wyniki • współpracuje w grupie podczas pracy nad projektem • implementuje i optymalizuje kod źródłowy gry, korzystając z wypracowanych założeń
--	--	--	--