

WYMAGANIA NA POSZCZEGÓLNE OCENY Z INFORMATYKI W KLASIE VIII

K – konieczny – ocena dopuszczająca (2)

P – podstawowy – ocena dostateczna (3)

R – rozszerzający – ocena dobra (4)

D – dopełniający – ocena bardzo dobra (5)

W – wykraczający – ocena celująca (6)

PRZELICZNIK OCEN		OPISOWO
0-40%	niedostateczny	E – bardzo słabo lub brak pracy. Ocenę E otrzymuje uczeń, który nie opanował podstawowych umiejętności i wiadomości i braki te uniemożliwiają mu kontynuację nauki w klasie programowo wyższej .
41-55%	dopuszczający	D – słabo, musisz więcej pracować. Ocenę D trzymuje uczeń, który ma poważne braki w opanowaniu materiału nauczania, ale wykazuje elementarny poziom rozumienia wiadomości z zakresu podstawowych pojęć, terminów zasad działania. Braki te nie uniemożliwiają mu kontynuowania nauki w klasie programowo wyższej.
56-69%	dostateczny	C – nieźle, lecz z licznymi błędami, musisz bardziej się starać. Ocenę C otrzymuje uczeń, który opanował podstawy programowe pozwalające mu na rozumienie najważniejszych zagadnień omawianych na lekcjach. Potrafi przedstawić wiadomości w innej formie, niż je zapamiętał, wytłumaczyć wiadomości, zinterpretować je, streścić i uporządkować /dokonać prostego wnioskowania/.
70-74%	+ dostateczny	
75-84%	dobry	B – dobrze wykonałeś zadanie, ale zrobiłeś parę błędów. Ocenę B otrzymuje uczeń, który posiada wiadomości i umiejętności określone dla poziomu wymagań podstawowych, a ponadto: opanował umiejętności praktycznego posługiwania się wiadomościami według podanych wzorów, potrafi umiejętnie wykorzystać zdobytą wiedzę do samodzielnego opanowywania zagadnień wskazanych przez nauczyciela, dokonuje analizy i syntezy zjawisk w odniesieniu do sytuacji typowych.
85-90%	+ dobry	
91-100%	bardzo dobry	A – wspaniale, bezbłędnie lub prawie bezbłędnie. Ocenę A otrzymuje uczeń, który posiada wiadomości i umiejętności określone dla poziomu wymagań podstawowych i rozszerzających, a ponadto sprawnie posługuje się wiadomościami, stosuje je do formułowania problemów, dokonuje analizy i syntezy nowych zjawisk, umie formułować plan działania, tworzy oryginalne rozwiązania, samodzielnie rozwiązuje problemy, szuka źródeł informacji, integruje wiedzę uzyskaną ze źródeł różnego typu.
WYMAGANIA WYKRACZAJĄCE (ocena celująca): stosowanie znanych wiadomości i umiejętności w sytuacjach trudnych, nietypowych, złożonych.		

Wymagania na poszczególne oceny

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

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

Ocena			
Stopień dopuszczający K Uczeń:	Stopień dostateczny P Uczeń:	Stopień dobry R Uczeń:	Stopień bardzo dobry D 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 języku C++ z wykorzystaniem zmiennych - wyjaśnia działanie operatora modulo	- 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 - stosuje odpowiednie polecenie języka C++, aby wyświetlić tekst na ekranie - omawia różnice pomiędzy kodem źródłowym a kodem wynikowym - tłumaczy, czym jest środowisko programistyczne	- 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 C++	- 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 języku C++ - 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 algorytm badania podzielności liczb • wyjaśnia potrzebę wyszukiwania informacji 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	• wykonuje obliczenia w języku C++ • omawia działanie operatorów arytmetycznych • stosuje tablice w języku C++ 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 • implementuje grę w zgadywanie liczby • 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	• wykorzystuje instrukcję warunkową if oraz if else w programach • wykorzystuje iterację w konstruowanych algorytmach • wykorzystuje w programach instrukcję iteracyjną for • definiuje funkcje w języku C++ 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	• wyjaśnia, jakie błędy zwraca interpreter • czyta kod źródłowy i opisuje jego działanie • wyjaśnia różnice między instrukcją iteracyjną while a pętlą for • 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
--	--	--	--