

Wymagania edukacyjne z informatyki klasa II zakres rozszerzony

Ocenę celującą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- tworzy złożone modele 3D.
- charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- 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,
- bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- w projektach zespołowych przyjmuje rolę lidera.

Ocenę bardzo dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- 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,
- wykorzystuje zaawansowane formuły, opracowując dane w arkuszu kalkulacyjnym,
- stosuje funkcje zaokrąglające liczby,
- korzysta z możliwości obliczeń walutowych,
- rozwiązuje problemy, wykorzystując programowanie strukturalne i obiektowe.
- charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,
- pisze programy o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku,
- optymalizuje rozwiązania,
- stosuje zaawansowane funkcje środowiska i języka programowania (np. z biblioteki STL),
- dobiera struktury danych i metody do rodzaju problemu,
- pisze programy konwertujące liczby między różnymi systemami pozycyjnymi,
- w programach wykonujących działania na liczbach w różnych systemach pozycyjnych wykorzystuje bibliotekę string i strukturalne typy danych,
- wykorzystuje rozwinięcie binarne liczby 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. sprawdzającą, czy napis jest palindromem,
- sprawdza, czy napisy są anagramami, stosując sortowanie lub zliczanie znaków,
- przy testowaniu liczby na pierwszość stosuje funkcję typu logicznego,
- wyszukuje liczby bliźniacze,
- wykorzystuje algorytm Euklidesa do działań na ułamkach, stosując struktury lub pary (typ pair),

- aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy.

Ocenę dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- wymienia podstawowe narzędzia programu GIMP,
- korzysta z warstw podczas pracy z programem GIMP,
- pracuje na warstwach w programie do grafiki wektorowej,
- 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, formularze i raporty,
- 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 realizacji problemu,
- 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,
- stosuje różne sposoby przekazywania parametrów do funkcji: przez wartość, referencję lub wskaźnik,
- implementuje w języku C++ algorytmy sprawdzające, czy napis jest palindromem,
- pisze programy sprawdzające, czy dwa napisy są anagramami, wykorzystując funkcję sort z biblioteki STL,
- implementuje w języku C++ i optymalizuje algorytm sprawdzający, czy liczba jest pierwsza,
- pisze program rozkładający liczby na czynniki pierwsze,
- stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW,
- wykorzystuje algorytm Euklidesa do działań na ułamkach,
- uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej.

Ocenę dostateczną otrzymuje uczeń, który spełnia kryteria oceny dopuszczającej, a ponadto:

- skaluje i kadruje obraz, dostosowując go do zadanego rozmiaru,
- wymienia podstawowe narzędzia programu Inkscape,
- tworzy dwuwymiarowe animacje,
- pobiera dane do arkusza kalkulacyjnego ze źródeł zewnętrznych,
- filtruje dane w arkuszu kalkulacyjnym,
- tworzy różne wykresy w arkuszu kalkulacyjnym w zależności od rodzaju danych,
- przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu,
- dodaje liczby binarne,
- konwertuje liczby między pozycyjnymi systemami liczbowymi,
- wykonuje działania arytmetyczne na liczbach w systemach liczbowych o różnych podstawach,

- 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,
- przedstawia w postaci algorytmu problem wyszukiwania anagramów,
- przy pisaniu programów stosuje własne funkcje różnych typów, w tym funkcję typu void,
- wyjaśnia różnicę między parametrami formalnym i aktualnym, a także między zmiennymi lokalną i globalną,
- implementuje w języku C++ algorytm naiwny sprawdzający, czy liczba jest pierwsza,
- implementuje w języku C++ algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem,
- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania.

Ocenę dopuszczającą otrzymuje uczeń, który:

- określa różnicę pomiędzy grafiką rastrową a wektorową,
- zapisuje wynik swojej pracy w różnych formatach graficznych,
- wyjaśnia, jak uruchomić środowisko do grafiki 3D online,
- wprowadza dane różnego typu do arkusza kalkulacyjnego,
- omawia zastosowania korespondencji seryjnej,
- wyjaśnia relacje w bazach danych.
- definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja,
- wymienia sposoby reprezentacji algorytmów,
- korzysta ze środowiska programistycznego: pisze w nim kod, kompiluje i uruchamia program, odczytuje i zapisuje pliki,
- 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,
- w pisanych programach korzysta ze strukturalnych typów danych: napisów, struktur, tablic,
- 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++ algorytm zliczający dzielniki danej liczby,
- omawia geometryczną interpretację algorytmu Euklidesa,
- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania o niewielkim stopniu trudności.

Ocenę niedostateczną otrzymuje uczeń, który:

- nie opanował podstawowych wiadomości i umiejętności, co uniemożliwia zdobywanie dalszej wiedzy,
- nie jest w stanie scharakteryzować podstawowych pojęć (algorytm, warunek, iteracja, rekurencja),
- nie zna prostych algorytmów,
- nie rozwiązuje najprostszych zadań,
- nie bierze czynnego udziału w lekcjach, nie wykonuje zadań, nie pisze programów, nie odrabia prac domowych.