**Wymagania z informatyki cz.3 ZP**

Przedmiotowy system oceniania (PSO) uczniów z przedmiotu Informatyka. Jest zgodny z podstawą programową oraz wewnątrzszkolnym systemem oceniania (WSO).

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

* charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych,
* bierze udział w konkursach informatycznych i zajmuje w nich punktowane miejsca,
* pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych lub oznaczone trzema gwiazdkami w podręczniku,
* optymalizuje programy, szacuje ich efektywność,
* wyszukuje w tekście anagramy i palindromy,
* pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere’a) i różnych kluczy (symetrycznych i asymetrycznych),
* pisze programy sortujące dane różnego typu (liczby, napisy, pary) oraz stosuje efektywne algorytmy sortowania (np. sortowanie szybkie, sortowanie przez scalanie),
* stosuje metody dynamiczną i zachłanną do rozwiązania problemów wydawania reszty i kinomana, wskazuje wady i zalety obu metod, szacuje ich złożoność czasową,
* programuje roboty tworzone na podstawie własnych projektów, steruje nimi za pomocą aplikacji mobilnych, wykazując się przy tym kreatywnością,
* tworzy podcasty i publikacje wideo wymagające znajomości zaawansowanych narzędzi i dużego nakładu pracy,
* przyjmuje rolę lidera w projektach zespołowych,
* tworzy rozbudowane infografiki, które skutecznie przekazują określone informacje,
* w dyskusjach panelowych przyjmuje funkcję eksperta.

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

* charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,
* realizuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku,
* optymalizuje rozwiązania,
* stosuje zaawansowane funkcje środowiska i języka programowania,
* dobiera struktury danych i metody do rodzaju problemu,
* szyfruje i deszyfruje dane, stosując popularne szyfry podstawieniowe i przestawieniowe,
* implementuje algorytmy sortowania bąbelkowego i przez wstawianie, zlicza kluczowe operacje (porównywanie i zamianę),
* wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach,
* implementuje algorytmy rekurencyjne, szacuje ich złożoność czasową,
* zastępuje iterację rekurencją i omawia konsekwencje takiej zamiany,
* programuje roboty, wykorzystując specjalistyczne narzędzia, tworzy własne projekty,
* tworzy interesujące podcasty i publikacje wideo,
* korzysta z różnych technik, tworząc infografikę,
* aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy,
* przyjmuje rolę moderatora lub eksperta w dyskusji panelowej.

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

* pisze programy o różnym stopniu trudności,
* dobiera typy danych do realizacji problemu,
* implementuje algorytmy tekstowe – w tym algorytmy porównywania i naiwnego wyszukiwania wzorca,
* wymienia metody łamania klasycznych szyfrów (atak siłowy, analiza częstości),
* pisze programy sortujące metodami prostymi (bąbelkowe i przez wstawianie), wskazuje operacje kluczowe,
* stosuje metodę zachłanną w przykładowych programach, wskazuje jej wady,
* porównuje algorytmy iteracyjne i rekurencyjne (liczbę wykonywanych operacji),
* implementuje w języku programowania algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi,
* unika błędów przybliżeń, stosuje całkowitoliczbowe typy danych,
* programuje roboty, wykorzystując specjalistyczne narzędzia (w tym symulatory online),
* tworzy podcasty i publikacje wideo,
* tworzy proste infografiki,
* uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej oraz dyskusji panelowej.

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

* wymienia sposoby przedstawiania informacji w komputerze,
* omawia i implementuje proste algorytmy przetwarzania tekstów,
* korzysta z funkcji i metod typu znakowego i napisów (łańcuchów znaków),
* implementuje przykładowe algorytmy szyfrowania (szyfry: kolumnowy, Cezara),
* przedstawia w postaci listy kroków lub schematu blokowego algorytmy sortowania prostego (bąbelkowe, przez wstawianie),
* definiuje rekurencję, algorytm rekurencyjny, warunki początkowe i wywołania rekurencyjne,
* definiuje rekurencyjnie ciągi liczbowe,
* formułuje algorytm wydawania reszty minimalną liczbą monet oraz rozwiązanie problemu kinomana z wykorzystaniem metody zachłannej,
* programuje roboty na wzór podanych przykładów,
* opracowuje treści internetowe z wykorzystaniem narzędzi graficznych i multimedialnych,
* wymienia sposoby porządkowania informacji oraz formułuje podstawowe zasady tworzenia infografik,
* uczestniczy w realizacji projektu zespołowego, wykonuje powierzone mu zadania.

**Ocenę dopuszczającą** otrzymuje uczeń, który:

* pisze programy o niewielkim stopniu trudności,
* wymienia sposoby zapisywania informacji w komputerze,
* definiuje pojęcia: kod liczbowy, UNICODE, ASCII,
* definiuje pojęcia: kryptologia, kryptografia, kryptoanaliza, tekst jawny, klucz, szyfrogram,
* rozróżnia szyfry podstawieniowe i przestawieniowe,
* omawia szyfr Cezara jako przykład szyfru podstawieniowego i szyfr kolumnowy jako przykład szyfru przestawieniowego,
* wyjaśnia, na czym polega łamanie szyfru,
* omawia metody sortowania prostego (bąbelkowe, przez wstawianie) na przykładowych danych,
* definiuje pojęcia iteracji i rekurencji,
* omawia zasadę złotego podziału,
* omawia metody zachłanne na przykładzie problemów wydawania reszty i kinomana,
* definiuje pojęcie robota, omawia jego budowę oraz wybrane parametry,
* 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 mu dalsze zdobywanie wiedzy,
* nie definiuje pojęć: kod liczbowy, UNICODE, ASCII, szyfrowanie, deszyfrowanie, sortowanie, algorytm zachłanny, rekurencja,
* nie implementuje prostych algorytmów tekstowych, szyfrowania, porządkowania,
* nie rozwiązuje najprostszych zadań,
* nie definiuje pojęcia robota ani nie opisuje jego budowy,
* nie opracowuje interesujących treści internetowych, nie posługuje się narzędziami graficznymi i multimedialnymi do wzbogacania treści,
* nie bierze czynnego udziału w lekcjach, nie wykonuje zadań, nie pisze programów, nie odrabia prac domowych,
* nie uczestniczy w projektach zespołowych.