**Wymagania edukacyjne na ocenę śródroczną oraz roczną**

**(zgodne z podstawą programową i Statutem Szkoły, dostosowane do specyfiki grupy)**

**Rok szkolny 2023/2024**

**zgodne z podstawą programową dla 5-letniego technikum**

|  |  |
| --- | --- |
| Przedmiot | Informatyka Rozszerzona |
| Klasa | 3t |
| Nauczyciel uczący | Katarzyna Maj |

**Program nauczania informatyki dla liceum ogólnokształcącego i technikum**

**Numer dopuszczenia: 1037/2/2020**

**Podręcznik: Informatyka na czasie 2. Zakres rozszerzony – Maciej Borowiecki**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Celujący** | **Bardzo dobry** | **Dobry** | **Dostateczny** | **Dopuszczający** |
| **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 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),
 | **Ocenę bardzo dobrą** otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:* 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++,
 | **Ocenę dobrą** otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:* określa specyfikację algorytmu (dane, wynik),
* pisze programy o różnym stopniu trudności, szacuje ich efektywność,
* przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, schematu blokowego, 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,
* omawia sposób reprezentacji obrazów w komputerze, korzystając z takich pojęć jak: piksel, model RGB, kanał alfa,
* wyjaśnia, na czym polega digitalizacja (dyskretyzacja) dźwięku,
* wyjaśnia zasadę tworzenia animacji,
 | **Ocenę dostateczną** otrzymuje uczeń, który spełnia kryteria oceny dopuszczającej, a ponadto:* przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu, schematu blokowego,
* 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++: shortint, int, longint, longlongint,
* 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,
 | **Ocenę dopuszczającą** otrzymuje uczeń, który:* 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,
 |
| Ocena roczna obejmuje zakres wiedzy i umiejętności z pierwszego półrocza oraz: |
| * implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję,
* pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere’a) i różnych kluczy,
* implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej,
* bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
* w projektach zespołowych przyjmuje rolę lidera.
 | * 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),
* szyfruje dane wczytane z pliku z uwzględnieniem polskich znaków diakrytycznych,
* pisze program odczytujący informację ukrytą za pomocą szyfru Cezara z wykorzystaniem analizy częstości znaków w tekście,
* aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach,prezentuje efekty wspólnej pracy.
 | * 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,
* szyfruje dane wczytane z pliku tekstowego,
* implementuje w języku C++ algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic,
* uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach.
 | * 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,
* pisze program szyfrujący napis szyfrem Cezara,
* omawia algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic,
 | * podaje definicje liczby pierwszej i liczby złożonej,
* implementuje w języku C++ algorytm zliczający dzielniki danej liczby,
* omawia geometryczną interpretację algorytmu Euklidesa,
* 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 algorytm zliczania znaków w tekście,
 |

**Nauczyciel przedmiotu uwzględnia zalecenia zawarte w opinii Poradni Psychologiczno – Pedagogicznej i dostosowuje do nich wymagania. Wymagania zgodne z podstawą programową dla szkół ponadpodstawowych.**

* 1. Sposobysprawdzaniawiedzyiumiejętności–m.in.
		+ Wykonywanie zadań praktycznych przy komputerze
		+ Odpowiedź ustna,
		+ pisemna,
		+ praca w grupie
	2. Sprawdzanie stopnia spełnienia wymagań,uwzględnia m. in.:
		+ Wzadaniach praktycznych: poprawność rozwiązania zadania, poprawność zastosowanej metody i narzędzi do rozwiązania zadania, poprawność merytoryczną,czas wykonania i terminowość oddania pracy,estetykę pracy
		+ W wypowiedzi ustnej i pracy pisemnej: precyzjęwypowiedzi, poprawność merytoryczną,wyczerpanie zagadnienia,puentę wypowiedzi;
		+ podczas pracy wgrupie– m.in.podział pracy zgodny z potrzebami uczniów,sposoby podejmowania decyzji,współdziałanie w grupie,postawę podczas pracy, formę prezentacji wyników pracy.