

WYMAGANIA EDUKACYJNE

PRZEDMIOT: INFORMATYKA

POZIOM: KLASA 2

ZAKRES: PODSTAWOWY

Wymagania na oceny śródroczne (I półrocze) oraz na oceny roczne obejmują wymagania z działów:

I. Tworzenie treści internetowych

II. Relacyjne bazy danych

III. Algorytmika i programowanie w języku C++/Python

Wymagania na oceny śródroczne (I półrocze) obejmują wymagania z działów od I do II włącznie, zaś na oceny roczne obejmują wszystkie wymagania z działów od I do III włącznie (cały rok szkolny).

1. Ustala się następujące progi procentowe na oceny bieżące:

- | | |
|------------------|------------|
| ▪ niedostateczny | 0% – 39% |
| ▪ dopuszczający | 40% – 50% |
| ▪ dostateczny | 51% – 70% |
| ▪ dobry | 71% – 85% |
| ▪ bardzo dobry | 86% – 95% |
| ▪ celujący | 96% – 100% |

2. Ocenie podlegają następujące obszary aktywności uczniów:

- praca na lekcji,
- prace pisemne (sprawdziany, kartkówki, zadanie domowe),
- wypowiedzi ustne (materiał bieżący),
- praca indywidualna i grupowa,
- projekt edukacyjny,
- prezentacje multimedialne,
- działania pozalekcyjne.

3. Zakłada się stosowanie następujących form sprawdzania wiadomości i umiejętności uczniów:

- sprawdzian,
- test,
- kartkówka (zapowiedziana, niezapowiedziana),
- odpowiedź ustna,
- praca na lekcji, praca w grupie,
- zadanie domowe,
- aktywność na lekcji.

4. Informacje o sprawdzaniu wiadomości i umiejętności z materiału bieżącego:

- materiał bieżący obejmuje trzy ostatnie lekcje,
- wiadomości i umiejętności sprawdzane są w następujący sposób:
 - a. odpowiedź ustna,
 - b. odpowiedź pisemna, zadanie praktyczne,
 - c. kartkówka – zapowiedziana i niezapowiedziana,
- można korzystać z „numerków szczęścia”,
- z „numerków szczęścia” nie korzystają uczniowie, którzy posiadają godziny nieusprawiedliwione w poprzednim miesiącu,
- „numerki szczęścia” nie obowiązują na zapowiedzianej kartkówce,
- osoby nieobecne podczas kartkówki zapowiedzianej są zobowiązane do jej nadrobienia – termin ustalony z nauczycielem,
- w przypadku stwierdzenia niesamodzielnego rozwiązywania zadań podczas kartkówki nauczyciel ma prawo unieważnić pracę, co jest równoważne z wystawieniem oceny niedostatecznej.

5. Informacje o sprawdzianach:

- sprawdzian może być poprzedzony jest lekcją powtórzeniową,
- sprawdzian jest zapowiadany co najmniej tydzień przed terminem, a informacja o nim jest odnotowana w dzienniku lekcyjnym,
- nie przekłada się sprawdzianów,
- osoby nieobecne podczas sprawdzianu są zobowiązane do jego uzupełnienia w terminie ustalonym z nauczycielem (wyjątek: osoby, które były przez dłuższy czas nieobecne w szkole [np. 2 tygodnie] i nie zdążyły nadrobić materiału, a uzyskały zgodę nauczyciela uczącego na pisanie w innym terminie),
- prace pisemne (sprawdziany, testy itp.) oddawane są w terminie do 21 dni,
- każda ocena ze sprawdzianu może być poprawiana w ustalonym terminie,
- prace pisemne są przechowywane przez nauczyciela; uczeń i jego rodzice mają możliwość wglądu do prac pisemnych,
- kartkówki nie podlegają poprawie,
- uczeń jest zobowiązany przygotować się do lekcji z 3 ostatnich tematów,
- uczeń może dwa razy na semestr skorzystać z „nieprzygotowania” i nie być brany pod uwagę przy wyborze do odpowiedzi ustnej,
- uczeń może uzyskiwać za aktywność na lekcji „plusy”. Po zebraniu odpowiedniej liczby plusów uczeń otrzymuje ocenę celującą z aktywności,

- w przypadku stwierdzenia niesamodzielnego rozwiązywania zadań podczas sprawdzianu nauczyciel ma prawo unieważnić pracę, co jest równoważne z wystawieniem oceny niedostatecznej.
6. O podwyższenie przewidywanej rocznej oceny klasyfikacyjnej z zajęć edukacyjnych może ubiegać się uczeń, który:
- a. Przystępował do wszystkich przewidzianych przez nauczyciela sprawdzianów i prac pisemnych
 - b. Skorzystał ze wszystkich oferowanych przez nauczyciela form poprawy

Wymagania edukacyjne niezbędne do otrzymania przez ucznia poszczególnych śródrocznych i rocznych ocen klasyfikacyjnych z informatyki w klasie drugiej

Nowa informatyka na czasie 2, zakres podstawowy

Wymagania na poszczególne oceny				
dopuszczający	dostateczny	dobry	bardzo dobry	celujący
2	3	4	5	6
I. TWORZENIE TREŚCI INTERNETOWYCH				
<i>Uczeń:</i> – zapisuje plik, nadając mu rozszerzenie .html, – rozróżnia sekcje HEAD i BODY oraz opisuje różnicę między tymi częściami kodu, – wymienia podstawowe znaczniki formatowania tekstu w języku HTML, – opisuje budowę znacznika HTML, – wyjaśnia pojęcie responsywności strony WWW, – uruchamia stronę WWW. – rozumie model działania aplikacji do wizualnego generowania stron i systemów zarządzania treścią CMS, – potrafi zbudować prostą stronę w aplikacji do wizualnego generowania stron, – opracowuje interesujące treści internetowe dostosowane do potrzeb	<i>Uczeń:</i> – rozumie i rozróżnia podstawowe elementy składni języka HTML, – opisuje podstawową strukturę strony w języku HTML, – tworzy nagłówki w języku HTML, – wstawia komentarze w kodzie HTML, – tworzy listy uporządkowane i nieuporządkowane, – rozumie cel pozycjonowania stron WWW, – korzysta z legalnych źródeł do pozyskiwania treści i materiałów dla tworzonych stron WWW, – potrafi przetestować w przeglądarce internetowej stronę WWW zbudowaną za pomocą aplikacji, – montuje materiały wykorzystując specjalistyczne oprogramowanie (np. Stream z pakietu Office 365),	<i>Uczeń:</i> – umieszcza zdjęcia na stronie WWW, – tworzy linki do zasobów zewnętrznych oraz miejsc w obrębie jednej strony, – poprawnie i na różne sposoby korzysta z opisu kolorów w języku HTML – korzysta z różnych przeglądarek internetowych. – potrafi w prostym narzędziu tekstowym poprawnie zapisać podstawowe znaczniki, w tym stosować hiperłącza, – biegle potrafi tworzyć i modyfikować znaczniki z rozbudowanymi atrybutami, – potrafi stosować grafikę, tabele i inne elementy w odniesieniu do responsywności strony internetowej, – potrafi weryfikować legalność pozyskiwanych	<i>Uczeń:</i> – korzysta ze ścieżek względnych i bezwzględnych w kodzie HTML, – poprawnie tworzy tabele o dowolnej strukturze, – dołącza style kaskadowe do dokumentu HTML, – tworzy ciekawą stronę WWW i publikuje ją w internecie, – rozumie pojęcia tabeli, listy i stylów oraz potrafi modyfikować przygotowane wcześniej kody źródłowe zawierające wskazane pojęcia, – testuje działanie strony WWW na różnych urządzeniach (komputer, smartfon), – tworzy podcasty i publikacje wideo na wybrane tematy, wymagające dużego nakładu pracy (np. promocja czy jubileusz szkoły, szkolny festiwal kultury lub nauki) lub korzysta z zaawansowanych narzędzi,	<i>Uczeń:</i> – tworzy style opisujące wygląd strony WWW, – dodaje do strony elementy odpowiedzialne za jej responsywność, – buduje stronę z wykorzystaniem systemu CMS i publikuje ją w Internecie, – tworzy stronę z tabelami, elementami graficznymi, – potrafi przetestować działanie napisanego kodu strony WWW w przeglądarce internetowej, – potrafi tworzyć i stosować arkusze stylów, – potrafi stworzyć złożoną stronę, sterowaną zaprojektowanym menu i ją opublikować oraz przetestować na różnych urządzeniach (komputer, smartfon), – rozumie, na czym polega zaawansowane zarządzanie treścią w systemach CMS,

potencjalnych odbiorców, wykorzystując zasadę 5W, – występuje przed kamerą i mikrofonem, przekazuje treści w sposób atrakcyjny dla odbiorców, utrzymuje ich uwagę, – uczestniczy w pracach zespołu i rozumie, jakie zadania wykonują poszczególni członkowie, – dba o ochronę prawa autorskiego publikowanych treści.	– wykazuje się kreatywnością w zakresie projektowania strony internetowej.	materiałów dla swojej strony internetowej, – dba o identyfikację wizualną, korzysta z narzędzi graficznych i multimedialnych do wzbogacania treści, – potrafi wykonać proste zadanie w zakresie przygotowania treści i zadanie techniczne w zakresie ich publikacji.	– potrafi współdziałać w zespole w zakresie projektowania i tworzenia strony internetowej, – skutecznie weryfikuje możliwości wykorzystania materiałów pod kątem praw autorskich, ochrony informacji oraz danych osobowych.	– wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku, – potrafi wykonać zaawansowane zadania techniczne.
--	--	--	--	---

II. RELACYJE BAZY DANYCH

<i>Uczeń:</i> – wyjaśnia podstawowe pojęcia związane z bazami danych: tabela, atrybut, rekord, pole, klucz główny, klucz obcy, relacja, – projektuje proste bazy danych, operuje w podstawowym zakresie poznanymi na lekcji narzędziami programu MS Access. – wymienia różne zastosowania baz danych.	<i>Uczeń:</i> – projektuje proste bazy danych, – operuje w podstawowym zakresie programem MS Access, – tworzy bazy danych w programie MS Access, – pomaga innym członkom grupy w wykonaniu ich zadań, – projektuje nieduże bazy danych.	<i>Uczeń:</i> – zarządza danymi w bazie danych w programie MS Access, – tworzy tabele i raporty w bazie danych i definiuje relacje między nimi, – podaje przykłady baz danych, w których podczas projektowania należy uwzględnić relacje typu: jeden do jednego, jeden do wielu, wiele do wielu, – modyfikuje dane zawarte w bazie danych, dodaje nowe atrybuty i określa typ danych w Widoku projektu, – tworzy tabele i definiuje relacje między nimi.	<i>Uczeń:</i> – pomaga w pracach innym uczestnikom projektu zespołowego, – tworzy formularze, kwerendy, drukuje i eksportuje raporty do plików w programie MS Access, – tworzy rozbudowane bazy danych, zawierające relacje typu: jeden do jednego, jeden do wielu, wiele do wielu, – modyfikuje bazy danych, poszerzając ich zastosowania.	<i>Uczeń:</i> – stosuje różne narzędzia do tworzenia relacyjnych baz danych, – projektuje rozbudowane bazy danych, – tworzy maski wprowadzenia.
--	--	---	---	--

– ALGORYTMIKA I PROGRAMOWANIE W JĘZYKU C++/PYTHON				
<i>Uczeń:</i> – wyjaśnia podstawowe pojęcia: algorytm, lista kroków, kod źródłowy, kod wynikowy, translator, kompilator, interpreter, słowa kluczowe, instrukcja, zmienna, operator przypisania, – wymienia cechy poprawnego algorytmu, tworzy proste algorytmy, – wyjaśnia na przykładzie pojęcie specyfikacji problemu, – podaje przykłady algorytmów spotykanych w codziennym życiu, – zapisuje algorytm z warunkami w postaci listy kroków, – zapisuje algorytm z warunkami w wybranym języku programowania, – wyjaśnia na przykładach pojęcia iteracji i pętli, – tworzy programy wykorzystujące zmienne całkowitoliczbowe, – zapisuje dwucyfrową liczbę dziesiętną w systemie binarnym, – wyjaśnia pojęcia: NWD, NWW,	<i>Uczeń:</i> – wymienia zasady tworzenia kodu źródłowego w wybranym języku programowania, – stosuje podstawowe konstrukcje wybranego języka programowania: instrukcje wejścia i wyjścia, instrukcje warunkowe oraz pętle, – zapisuje algorytm w postaci kodu źródłowego, – uruchamia zapisany kod źródłowy, – omawia przynajmniej dwie cechy poprawnego algorytmu, – zapisuje wybrane algorytmy za pomocą kodu źródłowego, – używa zmiennych różnych typów, – stosuje instrukcje wejścia i wyjścia, – stosuje instrukcje iteracyjne w postaci listy kroków, – zapisuje dwa rodzaje pętli, – stosuje w programach pętle, zapisuje liczbę dziesiętną w systemie binarnym, – zapisuje w postaci dziesiętnej liczby binarne, – wyjaśnia pojęcia: liczby pierwsze i liczby złożone,	<i>Uczeń:</i> – znajduje błędy w kodzie źródłowym programu na podstawie informacji zwrotnych z kompilatora, – tworzy program sprawdzający warunek trójkąta, – posługuje się w programowaniu strukturą tablicy lub listy, – buduje algorytmy sprawdzające podzielność jednej liczby przez drugą, – bada podzielność liczb z użyciem języka programowania, – omawia wybraną metodę sprawdzania, czy liczba jest pierwsza, – zapisuje wybraną metodę sprawdzania pierwszości w postaci funkcji języka programowania, – tworzy program realizujący algorytm Euklidesa w wersji z dodawaniem, – tworzy program komputerowy dodający ułamki, – zapisuje liczby w systemie liczbowym: dziesiętnym, binarnym, ósemkowym i szesnastkowym,	<i>Uczeń:</i> – samodzielnie tworzy programy komputerowe w wybranym języku programowania do rozwiązywania zadań matematycznych i fizycznych – optymalnie wykorzystuje różne rodzaje pętli w tworzonych programach, – analizuje i poprawia błędy w kodach źródłowych programów napisanych przez inne osoby, – omawia poznane na lekcjach algorytmy i uzasadnia, dlaczego spełniają cechy dobrych algorytmów, – tworzy samodzielnie programy z wykorzystaniem poznanych na lekcjach algorytmów, również z użyciem funkcji, – wyjaśnia pojęcia liczb doskonałych, bliźniaczych, zaprzyjaźnionych, – tworzy programy realizujące działania na ułamkach, – zapisuje w kodzie programu wywołania funkcji, również w instrukcji wyjścia, – wyjaśnia pojęcie składni i błędów składniowych, znajduje	<i>Uczeń:</i> – ilustruje pojęcie sprawności (efektywności) algorytmu na przykładach, – rozwiązuje różne zadania przy użyciu własnych algorytmów i programów komputerowych, – tworzy algorytmy i programy komputerowe do konwersji między systemami liczbowymi, – wykonuje działania na ułamkach za pomocą własnych programów komputerowych, – omawia pojęcie zasięgu zmiennych w programowaniu – tworzy samodzielnie programy, wykorzystując poznane instrukcje wybranego języka programowania, – stosuje w swoich programach zagnieżdżone instrukcje warunkowe, – pisze programy rozwiązujące zadania matematyczne i fizyczne oraz problemy z napisami, – zapisuje algorytm konwersji między systemami liczbowymi w postaci programu komputerowego,

– omawia na przykładzie działanie algorytmu Euklidesa.	– przedstawia metodę sprawdzania, czy liczba jest pierwsza, – bada podzielność liczb – zapisuje algorytm Euklidesa, – definiuje liczby złożone i liczby pierwsze oraz podstawowe twierdzenie arytmetyki, – podaje przykłady użycia liczb pierwszych, – omawia rozkład liczb na czynniki pierwsze, – omawia algorytmy sprawdzające podzielność liczb.	– wyjaśnia pojęcia: system pozycyjny, podstawa systemu liczbowego, dzielenie całkowite, reszta z dzielenia, – omawia algorytm konwersji liczb między systemami dziesiętnym i binarnym, – definiuje pojęcia: porządkowanie (sortowanie), wyszukiwanie sekwencyjne i wyszukiwanie binarne (połówkowe), – wyjaśnia znaczenie uporządkowania danych w procesie wyszukiwania, – wskazuje operacje kluczowe w algorytmach sortowania.	- i poprawia błędy w kodzie źródłowym programu, – bada podzielność wybranych liczb, programując poznane algorytmy w wybranym języku, – grupuje instrukcje w funkcje i wyjaśnia cel stosowania funkcji, – tworzy samodzielnie programy dla poznanych algorytmów, – pisze programy wykorzystujące poznane rodzaje liczb pierwszych, – wyjaśnia praktyczne znaczenie liczb pierwszych w informatyce.	– stosuje operację dzielenia całkowitego w rozwiązywaniu problemów, – tworzy algorytmy konwersji między różnymi systemami liczbowymi, – programuje algorytmy konwersji między różnymi systemami liczbowymi, stosując strukturę listy. – wykorzystuje strukturalne typy danych (listy) do przechowywania danych, stosuje pętle zagnieżdżone, – omawia oraz implementuje algorytm sortowania bąbelkowego (prostej zamiany) i przez wstawianie, zarówno nierosnąco, jak i niemalejąco, szacuje liczbę porównań oraz zamian w każdym z nich, – tworzy samodzielnie program sortujący z użyciem wartownika, – wykonuje zadania o podwyższonym stopniu trudności, – stosuje algorytmy sortowania o mniejszej złożoności czasowej (szybkie, przez scalanie).
--	--	---	--	--

Ocenę niedostateczną otrzymuje uczeń, który:

- nie opanował podstawowych wiadomości i umiejętności niezbędnych do dalszego zdobywania wiedzy,

- nie rozwiązuje najprostszych zadań,
- nie wykazuje zainteresowania treściami prezentowanymi na lekcjach, nie rozwiązuje ćwiczeń, zadań domowych,
- otrzymuje częściowe oceny niedostateczne, których nie poprawia.

Podział ze względu na tematy:

Numer lekcji	Temat	Liczba godzin	Zapisy podstawy programowej
Rozdział 1. Tworzenie treści internetowych			
1	Tworzenie stron internetowych	3	II.2, II.3.f, III.2, III.3, V.1
2	Wizualne tworzenie stron internetowych	2	II.2, II.3.f, II.4, III.2, III.3, V.1
3	Sztuka publikowania w sieci	3	II.3.a, II.3.b, II.3.e, II.4, III.2, III.3, IV.4, V.1
Rozdział 2. Relacyjne bazy danych			
4	Projektowanie bazy danych	2	I.1, II.2, II.3.d
5	Tworzenie bazy danych	3	I.1, II.2, II.3.d
Rozdział 3. Algorytmika i programowanie w języku C++/Python			
6	Od problemu do programu	4	I.1, I.3, II.1, II.2
7	Systemy liczbowe	3	I.1, I.2.a, I.3, II.1, II.2
8	Liczby pierwsze	3	I.1, I.2.a, I.3, II.1, II.2
9	Sortowanie liczb i wyszukiwanie	3	I.1, I.2.c, I.3, II.1, II.2
10	Tworzenie własnej strony internetowej – projekt zespołowy	4	II.3.a, II.3.e, II.3.f, II.4, III.2, III.3, IV.1, IV.5, V.1, V.4
Suma godzin			30