

Wymagania edukacyjne niezbędne do otrzymania przez ucznia poszczególnych śródrocznych i rocznych ocen klasyfikacyjnych z **informatyki** (zakres podstawowym) **w klasie drugiej**

1. Formy sprawdzania wiadomości i umiejętności podlegające ocenianiu bieżącemu:

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

2. 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**).

Wymagania na poszczególne oceny				
dopuszczający	dostateczny	dobry	bardzo dobry	celujący
2	3	4	5	6
I. TWORZENIE TREŚCI INTERNETOWYCH				
<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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 potencjalnych odbiorców, wykorzystując zasadę 5W, – występuje przed kamerą i mikrofonem, przekazuje treści 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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), – wykazuje się kreatywnością 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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, 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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),

<p>w sposób atrakcyjny dla odbiorców, utrzymuje ich uwagę,</p> <ul style="list-style-type: none"> – uczestniczy w pracach zespołu i rozumie, jakie zadania wykonują szczególnie członkowie, – dba o ochronę prawa autorskiego publikowanych treści. 	<p>w zakresie projektowania strony internetowej.</p>	<ul style="list-style-type: none"> – potrafi weryfikować legalność pozyskiwanych 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. 	<p>z zaawansowanych narzędzi,</p> <ul style="list-style-type: none"> – 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. 	<ul style="list-style-type: none"> – rozumie, na czym polega zaawansowane zarządzanie treścią w systemach CMS, – wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku, – potrafi wykonać zaawansowane zadania techniczne.
---	--	--	--	--

II. RELACJE BAZY DANYCH

<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – wyjaśnia podstawowe pojęcia związane z bazami danych: tabela, atrybut, rekord, pole, klucz główny, klucz obcy, relacja, – projektuje proste bazy danych, operując podstawowym zakresem poznanymi na lekcji narzędziami programu MS Access. – wymienia różne zastosowania baz danych. 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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. 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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. 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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. 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – stosuje różne narzędzia do tworzenia relacyjnych baz danych, – projektuje rozbudowane bazy danych, – tworzy maski wprowadzenia.
---	--	---	---	--

III. ALGORYTMIKA I PROGRAMOWANIE W JĘZYKU C++/PYTHON

<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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, 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – samodzielnie tworzy programy komputerowe w wybranym języku programowania do rozwiązywania zadań matematycznych i fizycznych – optymalnie wykorzystuje różne rodzaje pętli w stworzonych programach, – analizuje i poprawia błędy w kodach źródłowych programów 	<p><i>Uczeń:</i></p> <ul style="list-style-type: none"> – 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,
---	--	--	---	--

<p>w codziennym życiu,</p> <ul style="list-style-type: none"> - 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, - omawia na przykładzie działanie algorytmu Euklidesa. 	<ul style="list-style-type: none"> - 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, - 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. 	<p>podzielność jednej liczby przez drugą,</p> <ul style="list-style-type: none"> - 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, - 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. 	<p>napisanych przez inne osoby,</p> <ul style="list-style-type: none"> - 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 błędu składniowego, znajduje 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. 	<ul style="list-style-type: none"> - 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, - 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 najprostszyc zadań,
- nie wykazuje zainteresowania treściami prezentowanymi na lekcjach, nie rozwiązuje ćwiczeń, zadań domowych,
- otrzymuje cząstkowe oceny niedostateczne, których nie poprawia.