

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

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

W związku z uszczupleniem przez MEN podstawy programowej, w rozkładzie materiału zmniejszyła się liczba godzin na realizację obowiązkowych zagadnień. Uzyskane w ten sposób dodatkowe godziny pozostają do dyspozycji nauczyciela w trakcie roku szkolnego. Zgodnie z założeniami MEN: *Ograniczony zakres treści nauczania – wymagań szczegółowych – da nauczycielom i uczniom więcej czasu na spokojniejszą i bardziej dogłębną realizację programów nauczania.*

Lp.	Temat	Liczba godzin	Zapisy podstawy programowej
Rozdział 1. Arkusz kalkulacyjny i bazy danych			
1	Jak pobierać dane do arkusza kalkulacyjnego	3	II.3c, II.4
2	Wyciągamy wiedzę z danych	4	II.3c, II.4
3	Korespondencja seryjna	2	II.2, II.3b, II.3c, II.3d
4	Relacyjne bazy danych	3	II.2, II.3c, II.3d
P1	Współdziałanie aplikacji – projekt zespołowy	3	II.3b, II.3c, II.3.d, II.4, IV.5
Rozdział 2. Algorytmika i programowanie w języku Python			
5	Od problemu do programu	4	I.1, I.3, II.1, II.2
6	Wyszukujemy i sumujemy	2	I.1, I.3, II.1, II.2
7	Binarny system liczbowy	3	I.2a, I.3, II.1
8	Czy ta liczba jest pierwsza?	3	I.2a, I.3, II.1
9	Algorytm Euklidesa i działania na ułamkach	3	I.2a, I.3, II.1
Suma godzin			30

Plan wynikowy – Informatyka na czasie, część 2 (Python)

Lp.	Temat	Liczba godzin	Osiągnięcia uczniów	
			Wymagania podstawowe. Uczeń:	Wymagania ponadpodstawowe. Uczeń:
1	Jak pobierać dane do arkusza kalkulacyjnego	3	<ul style="list-style-type: none"> – wyjaśnia, dlaczego warto stosować narzędzia wymiany danych – wymienia podstawowe zastosowania arkusza kalkulacyjnego – wyjaśnia podstawowe pojęcia związane z arkuszem kalkulacyjnym: skoroszyt, arkusz, adres komórki, formuła, funkcja, zakres adresów – pobiera dane z różnych źródeł i przetwarza je – modyfikuje dane w arkuszu – wykorzystuje adresy komórek w formułach obliczeniowych – wyjaśnia różnice między formułami i funkcjami – korzysta z wbudowanych funkcji arkusza kalkulacyjnego – stosuje różne sposoby zaznaczania zakresów komórek – kopiuje dane z komórek i wkleja je na różne sposoby, również między arkuszami – kopiuje formuły – stosuje funkcje: SUMA, ŚREDNIA, MAX, MIN, DŁ, JEŻELI – przedstawia dane w postaci wykresów 	<ul style="list-style-type: none"> – wyszukuje samodzielnie w internecie dane potrzebne do realizacji określonych zadań – importuje do arkusza dane z różnych źródeł, w tym ze stron WWW – buduje złożone formuły pozwalające wykonywać obliczenia, rozwiązujące określone problemy – poprawnie stosuje adresowanie względne, bezwzględne i mieszane – modyfikuje dane podczas ich importowania – wyszukuje w internecie informacje na temat nowych funkcji i stosuje je w zadaniach – dobiera typ wykresu do rodzaju danych – interpretuje otrzymane wyniki zgodnie z ustalonymi założeniami
2	Wyciągamy wiedzę z danych	4	<ul style="list-style-type: none"> – omawia różnicę między filtrowaniem i sortowaniem danych – filtruje i sortuje dane – tworzy tabele i stosuje w nich sortowanie i filtrowanie danych – stosuje formuły arkusza kalkulacyjnego do losowego generowania zbiorów danych – stosuje funkcję INDEKS do zwracania wartości określonych komórek – opisuje możliwości tabel przestawnych 	<ul style="list-style-type: none"> – generuje zestawy danych za pomocą narzędzi online – modyfikuje style tabel przestawnych – buduje tabele przestawne dla dużych zbiorów danych – tworzy fragmentatory i korzysta z osi czasu – interpretuje wyniki tabel i wykresów przestawnych – stosuje tabele przestawne do rozwiązywania złożonych zadań, w których wykorzystano duże zbiory danych

			<ul style="list-style-type: none"> – tworzy tabele przestawne – filtruje dane w tabeli przestawnej – aktualizuje tabelę przestawną po modyfikacji danych źródłowych – stosuje gotowe style tabel przestawnych – podsumowuje dane w tabeli przestawnej na różne sposoby – stosuje różne sposoby wyświetlania wartości w tabeli przestawnej – grupuje i rozgrupowuje daty w tabelach przestawnych – tworzy wykresy przestawne 	
3	Korespondencja seryjna	2	<ul style="list-style-type: none"> – tworzy dokument główny korespondencji seryjnej – umieszcza pola korespondencji seryjnej w tworzonych dokumentach – tworzy bazę adresatów – stosuje reguły warunkowe do personalizacji listów seryjnych – poprawnie scala dokumenty seryjne 	<ul style="list-style-type: none"> – zarządza danymi adresatów korespondencji seryjnej w arkuszu kalkulacyjnym – tworzy zestawy dokumentów seryjnych (listy, etykiety, koperty) – drukuje listy seryjne – wysyła korespondencję seryjną za pomocą poczty elektronicznej
4	Relacyjne bazy danych	3	<ul style="list-style-type: none"> – wyjaśnia podstawowe pojęcia związane z bazami danych: tabela, atrybut, rekord, pole, klucz główny, klucz obcy, relacja – wymienia różne zastosowania baz danych – projektuje nieduże bazy danych – zarządza danymi w bazie danych w programie MS Access – modyfikuje dane zawarte w bazie danych – tworzy tabele i definiuje relacje między nimi 	<ul style="list-style-type: none"> – tworzy kwerendy w programie MS Access
P1	Współdziałanie aplikacji – projekt zespołowy	3	<ul style="list-style-type: none"> – współpracuje w grupie, korzystając z narzędzi online – korzysta z programów graficznych podczas pracy nad zadaniem projektowym – stosuje funkcje arkusza kalkulacyjnego do przetwarzania danych – testuje rozwiązania wypracowane w grupie – korzysta z zasobów internetowych, wyszukując potrzebne 	<ul style="list-style-type: none"> – prezentuje efekty pracy grupowej na forum klasy – przyjmuje rolę lidera odpowiedzialnego za zespół i projekt

			<p>informacje</p> <ul style="list-style-type: none"> – stosuje zasady netykiety i korzysta z niej w komunikacji zdalnej 	
5	Od problemu do programu	4	<ul style="list-style-type: none"> – wyjaśnia pojęcie algorytmu – podaje przykłady algorytmów niekomputerowych – wymienia cechy poprawnego algorytmu – wyjaśnia na przykładzie pojęcie specyfikacji problemu – tworzy algorytm wyznaczania pierwiastka kwadratowego – zapisuje algorytm Herona w postaci listy kroków – wyjaśnia pojęcia związane z algorytmiką i programowaniem: schemat blokowy, lista kroków, kod źródłowy, kod wynikowy, kompilator, interpreter, słowa kluczowe, funkcje, plik wykonywalny – zapisuje algorytm w postaci kodu źródłowego – kompiluje zapisany kod źródłowy – znajduje i poprawia błędy w kodzie źródłowym programu – wyjaśnia pojęcie zmiennej i typu zmiennej – wymienia zasady tworzenia kodu źródłowego w wybranym języku programowania – stosuje podstawowe konstrukcje wybranego języka programowania: instrukcje wejścia i wyjścia, operatory arytmetyczne i logiczne oraz instrukcję warunkową – tworzy program sprawdzający warunek trójkąta 	<ul style="list-style-type: none"> – 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
6	Wyszukujemy i sumujemy	2	<ul style="list-style-type: none"> – wyjaśnia, na czym polega iteracyjne rozwiązywanie problemu – stosuje w swoich programach podstawowe rodzaje pętli: while i for – opisuje zasady użycia pętli w programach – analizuje w tabeli działanie algorytmu krok po kroku – opisuje, jak komputer porównuje dwie wartości – tworzy program wyszukiwania największej liczby z danego ciągu liczb 	<ul style="list-style-type: none"> – stosuje różne rodzaje pętli, dostosowując wybór do rozwiązywanego problemu – stosuje w kodzie operatory inkrementacji i dekrementacji – używa w algorytmach rozwiązania z wartownikiem, czyli specjalną wartością wskazującą koniec ciągu wczytywanych do programu wartości – analizuje kod źródłowy i poprawia ewentualne błędy

7	Binarny system liczbowy	3	<ul style="list-style-type: none"> – zapisuje liczby w binarnym systemie liczbowym oraz w systemie szesnastkowym – wyjaśnia pojęcia związane z systemami liczbowymi: system pozycyjny, podstawa systemu liczbowego – omawia algorytm konwersji liczb między systemami dziesiętnym i binarnym – zapisuje algorytm konwersji między systemami liczbowymi w postaci programu komputerowego 	<ul style="list-style-type: none"> – stosuje operację dzielenia całkowitego w rozwiązywaniu problemów – tworzy algorytmy konwersji między różnymi systemami liczbowymi – programuje algorytmy konwersji między różnymi systemami liczbowymi, stosując strukturę tablicy lub listy
8	Czy ta liczba jest pierwsza?	3	<ul style="list-style-type: none"> – definiuje liczby złożone i liczby pierwsze – podaje przykłady użycia liczb pierwszych – omawia algorytmy sprawdzające podzielność liczb – bada podzielność wybranych liczb, programując poznane algorytmy w wybranym języku – grupuje instrukcje w funkcje i wyjaśnia cel stosowania funkcji 	<ul style="list-style-type: none"> – tworzy samodzielnie programy dla poznanych algorytmów – wyjaśnia pojęcia liczb doskonałych, bliźniaczych, zaprzyjaźnionych – pisze programy wykorzystujące poznane rodzaje liczb pierwszych – wyjaśnia praktyczne znaczenie liczb pierwszych w informatyce
9	Algorytm Euklidesa i działania na ułamkach	1	<ul style="list-style-type: none"> – wyjaśnia pojęcia: NWD, NWW – podaje przykłady zastosowania algorytmu Euklidesa – zapisuje algorytm Euklidesa w postaci listy kroków – tworzy program pozwalający na dodawanie ułamków – stosuje odpowiednie konstrukcje wybranego języka programowania do implementacji omawianych zagadnień (w tym: funkcję, która nie zwraca wartości) 	<ul style="list-style-type: none"> – tworzy programy realizujące działania na ułamkach – opisuje algorytm Euklidesa i tworzy realizujący go program w wybranym języku programowania – opisuje różnicę w sprawności dwóch wersji algorytmu Euklidesa: z odejmowaniem i z dzieleniem – poznaje inne zastosowania algorytmu Euklidesa, wykorzystując informacje zawarte w internecie lub innych źródłach