

PLAN WYNIKOWY

INFORMATYKA

KLASA II

Algorytmika i programowanie					
Lp.	Temat lekcji	Liczba godzin do realizacji	Przewidywane osiągnięcia ucznia	Odwołanie do podręcznika	Odwołanie do podstawy programowej
1.	Pojęcie algorytmu. Etapy rozwiązywania zadań za pomocą komputera.	1	Rozumie pojęcie <i>algorytm</i> . Potrafi podać przykładowe algorytmy związane z życiem codziennym, innymi nauczanyimi przedmiotami itp. Zna i stosuje etapy rozwiązywania zadań za pomocą komputera.	1.1 1.2	5.1 5.2 5.3 5.7
2.	Specyfikacja zadania. Sposoby reprezentowania algorytmów.	1	Poprawnie definiuje problem i formułuje jego specyfikację. Dokonuje analizy prostego i umiarkowanie złożonego zadania oraz opracowuje algorytm zgodny ze specyfikacją. Zna różne sposoby reprezentowania algorytmów, w tym opis słowny, listę kroków, schemat blokowy, drzewo algorytmu, program.	1.2 1.3 1.3.1 1.3.2 1.3.3 1.3.4	5.2 5.4 5.7 5.17
3.	Schemat blokowy — graficzna prezentacja algorytmu. Lista kroków — słowna prezentacja algorytmu.	2	Realizuje algorytmy, stosując różne sposoby ich reprezentowania, w tym schemat blokowy, listę kroków. Wyodrębnia elementy składowe algorytmu. Dobiera właściwy sposób rozwiązania i prezentacji algorytmu do konkretnego problemu.	1.3.1 1.3.2	5.2 5.4 5.25
4.	Języki programowania — pojęcie, klasyfikacja, przykłady.	1	Rozumie pojęcia: język programowania, translator, kompilator, interpreter. Zna klasyfikację języków programowania: imperatywne i deklaratywne, niskiego poziomu i wysokiego poziomu. Potrafi wymienić i sklasyfikować podstawowe języki programowania. Kompiluje i uruchamia przykładowe programy napisane w różnych językach programowania.	3.1	5.22
5.	Podstawowe zasady i metody programowania.	1	Zna i rozumie podstawowe zasady i metody programowania.	1.4 1.5 1.6	5.13
6.	Elementy zintegrowanego systemu programowania.	1	Korzysta z wybranego środowiska programistycznego (np. kompilatora), w którym zapisuje, kompiluje, uruchamia i testuje programy.	3 3.1	5.22
7.	Struktura programu.	1	Zna podstawową strukturę programu. Stosuje komentarze przy pisaniu programów.	3.2 3.2.1 3.2.8	5.21 5.23
8.	Operacje wejścia i wyjścia oraz ich zastosowanie w programach.	1	Korzysta w programach z podstawowych operacji wejścia i wyjścia.	3.2.2	5.21 5.23
9.	Zmienne i stałe — deklaracja i zastosowanie w programach.	1	Potrafi deklarować zmienne i wykorzystywać je w programach. Potrafi deklarować stałe i wykorzystywać je w programach.	3.2.3	5.21 5.23
10.	Wyrażenia arytmetyczne, relacje i operatory logiczne w programach.	1	Zna podstawowe wyrażenia arytmetyczne, relacje, operatory logiczne i funkcje matematyczne oraz stosuje je w programach. Zna priorytety relacji i działań charakterystyczne dla danego języka programowania oraz uwzględnia je podczas pisania programów.	3.2.4 3.2.5 3.2.6	5.21 5.23
11.	Generowanie liczb losowych.	1	Potrafi generować liczby losowe w wybranym języku programowania.	3.2.7	5.23
12.	Podstawowe konstrukcje algorytmiczne: instrukcja pusta, instrukcja przypisania, instrukcja złożona, instrukcje warunkowe, instrukcja wyboru, instrukcje	2	Zna i potrafi stosować podstawowe konstrukcje algorytmiczne, w tym instrukcję pustą, instrukcję przypisania, instrukcję złożoną, instrukcje warunkowe, instrukcję wyboru, instrukcje iteracyjne.	3.3.1 3.3.2 3.3.3 3.3.4 3.3.5	5.21 5.23

	iteracyjne.			3.3.6	
13.	Umiejętność korzystania w implementacjach z podstawowych konstrukcji algorytmicznych.	2	Stosuje w programach podstawowe konstrukcje algorytmiczne.	3.3	5.12 5.21 5.23
14.	Proste typy danych: typy całkowite, typy rzeczywiste, typ znakowy i typ logiczny.	1	Rozumie pojęcie proste typy danych. Zna proste typy danych, w tym: typy całkowite, typy rzeczywiste, typ znakowy, typ logiczny.	3.4	5.21 5.23
15.	Operacje na prostych typach danych oraz ich wykorzystanie w programach.	1	Wykonuje operacje na prostych typach danych. Korzysta w programach z prostych typów danych. Potrafi w sytuacji problemowej wybrać właściwy typ danych dla wykorzystywanych zmiennych.	3.4	5.21 5.23
16.	Algorytmy liniowe i z warunkami.	1	Rozumie pojęcia: algorytm liniowy, algorytm z warunkami. Identyfikuje metodę programowania liniowego i z warunkami wykorzystaną w przykładowych algorytmach.	1.4.1 1.4.2	5.5
17.	Realizacja algorytmów liniowych i warunkowych.	1	Stosuje algorytmy liniowe i algorytmy z warunkami podczas rozwiązywania problemów. Przedstawia algorytmy liniowe i warunkowe w postaci listy kroków, schematu blokowego i programu. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	1.4.1 1.4.2	4.4 5.5 5.12 5.21
18.	Algorytmy badające własności geometryczne.	1	Realizuje algorytmy: sprawdzanie warunku trójkąta, określanie prostokątności i równoległości prostych, obliczanie długości odcinka, obliczanie odległości punktu od prostej, obliczanie odległości punktów na płaszczyźnie, badanie położenia punktu względem prostej, badanie przynależności punktu do odcinka. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	2.1	5.11.f 5.21
19.	Rozwiązywanie równania liniowego i kwadratowego.	2	Realizuje algorytmy: rozwiązywanie równania liniowego, rozwiązywanie równania kwadratowego, stabilny algorytm rozwiązujący równanie kwadratowe. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	1.3.1 1.3.2 1.3.3 1.3.4 1.4.3	4.4 5.5 5.12 5.21
20.	Iteracja jako metoda programowania.	1	Rozumie pojęcia: iteracja, wzór iteracyjny. Potrafi definiować iterację. Stosuje metodę iteracji przy realizacji algorytmów. Identyfikuje wykorzystaną metodę iteracji w przykładowych algorytmach.	1.5	5.5 5.23
21.	Realizacja algorytmów iteracyjnych — wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych.	3	Realizuje algorytmy iteracyjne: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, proste i złożone algorytmy wykonywane na tekstach (łańcuchach) i tablicach (np. znajdowanie podciągów o określonych własnościach) itp. Przedstawia algorytmy iteracyjne w postaci listy kroków, schematu blokowego i programu. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	1.5	4.4 5.11.a 5.12 5.21 5.23
22.	Złożoność obliczeniowa algorytmów.	1	Rozumie pojęcia: złożoność obliczeniowa, złożoność czasowa, złożoność pamięciowa. Potrafi wyznaczać, oceniać i uzasadniać złożoność czasową algorytmów. Potrafi wyznaczać, oceniać i uzasadniać złożoność pamięciową algorytmów.	1.10.1	5.18 5.19
23.	Analiza złożoności obliczeniowej przykładowych algorytmów.	1	Analizuje złożoność czasową wybranych algorytmów. Porównuje złożoność czasową algorytmów rozwiązujących ten sam problem. Dobiera możliwie najszybszy algorytm rozwiązujący postawiony problem. Analizuje złożoność pamięciową wybranych algorytmów. Porównuje złożoność pamięciową algorytmów rozwiązujących ten sam problem. Dobiera odpowiednie struktury danych umożliwiające	1.10.1 2	5.4 5.12 5.14 5.18 5.19

			rozwiązanie postawionego problemu.		
24.	Własności algorytmów — efektywność, poprawność, skończoność i optymalność.	1	Rozumie pojęcia: własności algorytmów, efektywność algorytmu, poprawność algorytmu, skończoność algorytmu, optymalność algorytmu. Analizuje własności wybranych algorytmów. Rozwiązując problem, wybiera algorytm najkorzystniejszy ze względu na jego własności. Oceni i uzasadnia efektywność, poprawność, skończoność i optymalność wybranych algorytmów.	1.10.1 1.10.2 1.10.3	5.4 5.6 5.16 5.20 5.26
25.	Ogólna charakterystyka i cel strukturalizacji programu.	1	Rozumie podstawowe zasady i cel strukturalizacji programu. Potrafi wydzielać fragmenty programu i definiować funkcje.	3.5.1	5.12 5.15 5.23

26.	Zmienne globalne i lokalne oraz ich wykorzystanie w funkcjach.	1	Rozumie różnice między zmiennymi lokalnymi i globalnymi oraz świadomie korzysta z wybranego typu zmiennych w programach.	3.5.2	5.15 5.23
27.	Sposoby przekazywania parametrów w funkcjach. Przeładowanie funkcji.	1	Rozumie różnice między parametrami formalnymi i aktualnymi. Zna sposoby przekazywania parametrów w funkcjach i świadomie stosuje je podczas pisania programów. Zna typy przeładowania funkcji i stosuje je w programach.	3.5.3 3.5.4	5.15 5.23
28.	Zastosowanie w implementacjach funkcji z parametrami.	1	Stosuje w programach funkcje z parametrami. Potrafi dokonać wyboru właściwego sposobu przekazywania parametrów w określonej sytuacji.	3.5.3	5.15 5.23
29.	Rekurencja jako metoda programowania.	1	Rozumie pojęcia: rekurencja, wzór rekurencyjny, zależność rekurencyjna. Potrafi podać przykłady zastosowania rekurencji. Potrafi definiować zależności rekurencyjne oraz odpowiednie funkcje rekurencyjne. Realizuje algorytmy rekurencyjne: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wyznaczanie elementów prostych rekurencyjnych ciągów liczbowych. Identyfikuje wykorzystaną metodę rekurencji w przykładowych algorytmach.	1.6 1.6.1 1.6.2	5.5 5.9 5.11.a 5.21 5.23
30.	Realizacja algorytmów rekurencyjnych — obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, zamiana iteracji na rekurencję, wieże Hanoi.	2	Stosuje metodę rekurencji przy realizacji algorytmów. Przedstawia algorytmy rekurencyjne w postaci listy kroków i programu. Realizuje algorytmy rekurencyjne: wyznaczanie elementów rekurencyjnych ciągów liczbowych, wieże Hanoi itp. Potrafi dokonywać zamiany metody rekurencyjnej wykorzystanej w algorytmie na iteracyjną. Analizuje wartości zmiennych w przykładowych algorytmach rekurencyjnych.	1.6 1.6.3	5.9 5.11.a 5.12 5.21 5.23
31.	Wyznaczanie NWD (algorytm Euklidesa) i NWW dwóch liczb naturalnych.	1	Za pomocą algorytmu Euklidesa wyznacza największy wspólny dzielnik dwóch liczb naturalnych. Wyznacza najmniejszą wspólną wielokrotność dwóch liczb naturalnych, wykorzystując algorytm obliczający największy wspólny dzielnik. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	2.2 2.2.1 2.2.2	4.4 5.4 5.9 5.11.a 5.21
32.	Strukturalne typy danych — typ tablicowy.	1	Rozumie pojęcia: strukturalne typy danych, abstrakcyjne typy danych. Potrafi definiować typ tablicowy, w tym tablice jedno- i wielowymiarowe. Realizuje programy z wykorzystaniem typu tablicowego.	3.6.1	5.12 5.14 5.21

33.	Realizacja algorytmów z wykorzystaniem tablicy.	2	Realizuje algorytmy z wykorzystaniem tablicy w postaci listy kroków, schematu blokowego i programu.	3.6.1	5.12 5.14 5.21 5.23
34.	Badanie, czy liczba jest pierwsza, rozkład liczby na czynniki pierwsze, generowanie liczb pierwszych (sito Eratostenesa).	1	Rozumie pojęcie liczba pierwsza. Realizuje algorytm sprawdzający, czy dana liczba jest pierwsza. Dokonuje rozkładu liczby na czynniki pierwsze. Realizuje algorytm generujący liczby pierwsze — sito Eratostenesa. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	2.4.1 2.4.2	4.4 5.11.a 5.21
35.	Wyznaczanie wartości wielomianu schematem Hornera, szybkie podnoszenie do potęgi z wykorzystaniem schematu Hornera.	1	Wyznacza wartość wielomianu schematem Hornera. Realizuje algorytm szybkiego podnoszenia do potęgi z wykorzystaniem schematu Hornera.	2.3.4	5.4 5.11.c 5.21
36.	Pozycyjne systemy liczbowe — system binarny, ósmkowy i szesnastkowy.	1	Rozumie pojęcia: system liczbowy, pozycyjny system liczbowy. Zna definicje systemu dwójkowego (binarnego), ósemkowego (ósmkowego) i szesnastkowego (heksadecymalnego).	2.3.1	5.11.a
37.	Zamiana liczb z dowolnego systemu na system dziesiętny z zastosowaniem schematu Hornera.	1	Realizuje algorytm zamiany liczby z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera.	2.3.5	5.11.a 5.21

38.	Zamiana liczb z systemu dziesiętnego na dowolny inny system liczbowy.	1	Realizuje algorytm zamiany liczby z systemu dziesiętnego na dowolny inny pozycyjny system liczbowy. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmu.	2.3.2	4.4 5.11.a 5.21
39.	Wykonywanie podstawowych działań arytmetycznych w różnych systemach liczbowych.	1	Realizuje algorytmy wykonujące podstawowe operacje arytmetyczne w systemach liczbowych: dwójkowym, ósemkowym i szesnastkowym. Realizuje algorytmy wykonujące podstawowe działania arytmetyczne w różnych pozycyjnych systemach liczbowych.	2.3.3	5.11.a 5.21
40.	Reprezentacja danych liczbowych w komputerze i błędy w obliczeniach.	1	Zna reprezentacje danych liczbowych w komputerze, w tym reprezentację binarną liczb całkowitych i niecałkowitych, stałopozycyjną reprezentację liczb, zmiennopozycyjną reprezentację liczb, pojęcie mantysy i cechy. Wyjaśnia źródła błędów w obliczeniach komputerowych, w tym błąd względny i bezwzględny.	2.3.6 2.3.7	1.1 5.27
41.	Programowanie liniowe — definicja i analiza przykładów.	1	Rozumie pojęcie programowanie liniowe. Rozpoznaje metodę programowania liniowego w poznanych algorytmach. Stosuje programowanie liniowe przy konstruowaniu algorytmów.	2.5	5.5 5.12 5.21
42.	Liniowe przeszukiwanie ciągu liczbowego.	1	Wykorzystuje metody liniowe przy przeszukiwaniu ciągu liczbowego. Realizuje algorytm przeszukiwania liniowego tablicy jednowymiarowej. Realizuje algorytm przeszukiwania liniowego tablicy jednowymiarowej z wartownikiem. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	2.5.1 2.5.2	4.4 5.11.b 5.21
43.	Znajdowanie minimalnego lub maksymalnego elementu w ciągu liczbowym.	1	Realizuje algorytm znajdowania maksymalnego elementu w ciągu liczbowym. Realizuje algorytm znajdowania minimalnego elementu w ciągu liczbowym. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	2.6	4.4 5.11.b 5.21
44.	Znajdowanie lidera w zbiorze.	1	Zna pojęcie lidera w zbiorze. Realizuje algorytm znajdowania lidera w zbiorze.	2.7	5.11.b 5.21
45.	Monotoniczność ciągu	1	Rozumie pojęcie monotoniczność ciągu liczbowego.	2.8	4.4

	liczbowego.		Realizuje algorytm sprawdzający, czy ciąg liczbowy jest rosnący, czy malejący. Realizuje algorytm sprawdzający, czy ciąg liczbowy jest nierosnący lub niemalejący. Realizuje algorytm sprawdzający, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.		5.11.a 5.21
46.	Liniowe sortowanie ciągu liczbowego.	2	Zna liniowe metody sortowania ciągu liczbowego. Realizuje algorytm porządkowania przez wybór. Realizuje algorytm porządkowania przez wstawianie. Realizuje algorytm porządkowania bąbelkowego. Realizuje algorytm sortowania przez zliczanie.	2.9 2.9.1 2.9.2	5.4 5.11.b 5.21
47.	Metoda „dziel i zwyciężaj” — definicja i analiza przykładów.	1	Rozumie pojęcie metoda „dziel i zwyciężaj”. Potrafi definiować i stosować metodę „dziel i zwyciężaj” w odpowiednich sytuacjach. Identyfikuje metodę „dziel i zwyciężaj” w przykładowych algorytmach.	1.7	5.5 5.8
48.	Zastosowanie metody „dziel i zwyciężaj” w algorytmach.	3	Wykorzystuje metodę „dziel i zwyciężaj” przy realizacji algorytmów. Realizuje algorytm przeszukiwania binarnego uporządkowanego ciągu liczbowego. Realizuje algorytm jednoczesnego znajdowania minimalnego i maksymalnego elementu. Realizuje algorytm sortowania przez scalanie. Realizuje algorytm sortowania szybkiego. Realizuje algorytm znajdowania przybliżonej wartości miejsca zerowego funkcji ciągłej — metoda połowienia przedziałów.	1.7.1 2.10.1 2.10.2 2.10.3 2.11.3	5.4 5.8 5.11.b 5.11.c 5.12 5.21
49.	Obliczenia przybliżone. Obliczanie wartości pierwiastka kwadratowego z liczby dodatniej (algorytm Newtona-Raphsona).	1	Rozumie pojęcia: obliczenia przybliżone, metody numeryczne. Realizuje algorytm obliczający wartość pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona). Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	2.11 2.11.1	4.4 5.5 5.21

65.	Realizacja przykładowych	5	Wykorzystuje różne metody algorytmiczne i poznane	część 1	5.1
50.	Obliczanie planu przeszukiwania w grafie przy pomocy algorytmu.	1	Realizuje algorytm przeszukiwania w grafie przy pomocy algorytmu.	podręcznika 2.11.2	5.12.c 5.21 5.5
51.	Dynamiczne struktury danych — listy, stosy, kolejki, drzewa.	2	Potrafi definiować dynamiczne struktury danych, w tym listy, stosy, kolejki, drzewa binarne. Pisze programy z wykorzystaniem dynamicznych struktur danych. Konstruuje binarne drzewo poszukiwań oraz wykonuje operacje dodawania i sortowania elementów drzewa.	3.7 3.7.1 3.7.2 3.7.3 3.7.4	5.12 5.12 5.24 5.24 5.25
66.	Projektowanie i realizacja	4	Rozumie pojęcie programowania.	4.1	5.28
52.	— definicja i analiza przykładów.	1	Potrafi określić kolejność wykonania instrukcji w programie.	4.2 1.8.1	5.3 5.10
53.	Algorytm zachłanny dla problemu plecakowego, algorytm wydawania reszty.	1	Realizuje odpowiednio informację, nie bierze do realizacji projektów z różnych dziedzin. Realizuje indywidualny lub zespołowo projekt programowy, w tym określa etapy pracy i dzieli zadanie na moduły, wykonuje określone części zadania, łączy poszczególne moduły w całość, sporządza dokumentację projektu.	2.12.1 2.12.2	5.10 5.11.a 5.12 5.21
54.	Strukturalne typy danych — typ łańcuchowy.	1	Zna predefiniowane operacje na tekście i korzysta z nich. Realizuje programy z wykorzystaniem typu łańcuchowego.	3.6.2	5.14 5.21 5.23
55.	Realizacja algorytmów na tekstach.	2	Realizuje algorytmy wykonywane na tekstach w postaci listy kroków, schematu blokowego i programu, na przykład sprawdzanie, czy tekst jest palindromem, sortowanie tekstu, sprawdzanie, czy teksty są anagramami, wyszukiwanie wzorca w tekście.	2.13 2.13.1 2.13.2 2.13.3 2.13.4 3.6.2	5.11.d 5.12 5.14 5.21
56.	Wyznaczanie wartości wyrażenia zapisanego w odwrotnej notacji polskiej (ONP).	1	Zna zasady zapisu wyrażenia podanego w postaci ONP. Wykorzystując stos, realizuje algorytm obliczający wartość wyrażenia zapisanego w postaci ONP.	2.13.5	5.11.d 5.12 5.14 5.21
57.	Kryptografia i kryptoanaliza. Metody szyfrowania.	1	Rozumie pojęcia: kryptografia, kryptoanaliza, algorytmy kryptograficzne. Potrafi wymienić podstawowe metody szyfrowania.	1.9	2.5 5.11.e
58.	Realizacja wybranych algorytmów kryptograficznych.	2	Realizuje przykładowe algorytmy kryptograficzne, w tym algorytmy symetryczne, asymetryczne itp. Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów.	2.14.1 2.14.2	4.4 5.11.e 5.12 5.14 5.21
59.	Strukturalne typy danych — struktury.	1	Potrafi definiować typ strukturalny.	3.6.3	5.12 5.14 5.23
60.	Realizacja algorytmów z zastosowaniem struktur.	1	Realizuje programy z wykorzystaniem typu strukturalnego.	3.6.3	5.2 5.12 5.14 5.21
61.	Tablice struktur i ich zastosowanie w implementacjach.	1	Potrafi definiować tablice struktur. Realizuje programy z wykorzystaniem tablic struktur.	3.6.3	5.2 5.12 5.14 5.21 5.23
62.	Plikowe operacje wejścia-wyjścia.	2	Zna i potrafi definiować różne typy plików, w tym tekstowe, binarne. Zna i realizuje etapy przetwarzania plików. Zna podstawowe operacje na plikach i korzysta z nich.	3.8	5.12 5.14
63.	Realizacja programów z zastosowaniem typu plikowego.	2	Realizuje programy z wykorzystaniem typu plikowego, m.in. importuje dane z plików zewnętrznych i eksportuje wyniki do plików zewnętrznych.	3.8	5.12 5.14 5.21
64.	Tworzenie prostych baz danych z zastosowaniem operacji na plikach.	2	Potrafi skonstruować prostą bazę danych z zastosowaniem operacji na plikach.	3.8	5.12 5.14 5.21

