

# Technologia informacyjna - informatyka

## Wymagania na poszczególne oceny szkolne

### 1. Metody rozwiązywania problemów algorytmicznych

#### 1.1. Sposoby prezentacji algorytmów

2	3	4	5	6
<p>Wie, co to jest algorytm.</p> <p>Określa dane do zadania oraz wyniki.</p> <p>Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczenia w schemacie blokowym.</p> <p>Potrafi narysować (od ręcznie) schemat blokowy algorytmu liniowego.</p>	<p>Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy. Zna pojęcie specyfikacji zadania.</p> <p>Zna wybrane sposoby prezentacji algorytmów. Przedstawia algorytm w postaci listy kroków.</p> <p>Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Podczas rysowania schematów blokowych potrafi wykorzystać Autokształty z edytora tekstu.</p> <p>Korzysta (w stopniu podstawowym) z programu edukacyjnego do symulacji działania algorytmu skonstruowanego w postaci schematu blokowego.</p>	<p>Określa zależności między problemem, algorytmem a programem komputerowym.</p> <p>Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów. Przedstawia dokładną specyfikację dowolnego zadania.</p> <p>Zna znaczenie i działanie instrukcji symbolicznego języka programowania (pseudojęzyka).</p> <p>Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci.</p> <p>Potrafi skonstruować algorytm z warunkami zagnieżdżonymi i pętlą za pomocą programu edukacyjnego.</p>	<p>Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji), m.in. w pseudojęzyku.</p> <p>Zapisuje algorytmy z pętlą zagnieżdżoną.</p> <p>Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego.</p> <p>Analizuje działanie algorytmu dla przykładowych danych.</p> <p>Stosuje swobodnie oprogramowanie edukacyjne do graficznej prezentacji i analizy algorytmów.</p>	<p>Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji).</p> <p>Potrafi trafnie dobrać do algorytmu sposób prezentacji.</p> <p>Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia.</p> <p>Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych.</p> <p>Potrafi zaproponować własny pseudojęzyk (postać instrukcji i zasady składni).</p>

#### 1.2. Przegląd technik algorytmicznych i algorytmów klasycznych

2	3	4	5	6
<p>Określa sytuacje warunkowe. Podaje przykłady zadań, w których występują sytuacje warunkowe.</p> <p>Wie, na czym polega powtarzanie tych samych operacji.</p> <p>Potrafi omówić, na przykładzie, algorytm znajdowania najmniejszego z trzech elementów.</p>	<p>Potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzieniami).</p> <p>Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady.</p> <p>Wie, od czego zależy liczba powtórzeń.</p>	<p>Analizuje algorytmy, w których występują powtórzenia (iteracje). Zna sposoby zakończenia iteracji.</p> <p>Określa kroki iteracji.</p> <p>Potrafi zapisać w wybranej notacji np. algorytm sumowania <math>n</math> liczb, algorytm obliczania silni, znajdowania minimum w ciągu <math>n</math> liczb, algorytm rozwiązywania równania liniowego.</p>	<p>Zna metodę „dziel i zwyciężaj”, algorytm generowania liczb Fibonacciego, schemat Hornera. Omawia ich iteracyjną realizację i potrafi przedstawić jeden z nich w wybranej notacji.</p> <p>Zna inne algorytmy sortowania, np. kubełkowe, przez wstawianie.</p> <p>Zna przynajmniej jeden algorytm numeryczny, np. obliczanie wartości pierwiastka kwadratowego.</p>	<p>Rozumie dokładnie technikę rekurencji (znaczenie stosu).</p> <p>Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję.</p> <p>Zna trudniejsze algorytmy, np. algorytm trwałego małżeństwa, wieże Hanoi, problem ośmiu hetmanów.</p>

	<p>Potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy.</p> <p>Omawia i analizuje wybrane techniki sortowania w postaci gotowych schematów blokowych, skonstruowanych w programie edukacyjnym.</p>	<p>Zna iteracyjną postać algorytmu Euklidesa. Zna przynajmniej dwie techniki sortowania, np. bąbelkowe i przez wybór.</p> <p>Określa problemy, w których występuje rekurencja i podaje przykłady „zjawisk rekurencyjnych” – wziętych z życia i zadań szkolnych. Zna rekurencyjną realizację wybranego algorytmu, np. silni.</p>	<p>Wskazuje różnicę między rekurencją a iteracją.</p> <p>Zna rekurencyjną realizację wybranych algorytmów, np. silnię i algorytm Euklidesa.</p> <p>Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną.</p>	<p>Zna inne techniki sortowania, np. sortowanie przez scalanie ciągów i metodę szybką.</p> <p>Potrafi zapisać je w różnych notacjach (również w języku programowania wysokiego poziomu).</p> <p>Zna inne algorytmy numeryczne, np. wyznaczanie miejsca zerowego funkcji.</p> <p>Korzysta samodzielnie z dodatkowej literatury.</p>
--	---	---	---	--

### 1.3. Elementy analizy algorytmów

2	3	4	5	6
<p>Potrafi zanalizować przebieg algorytmu dla przykładowych danych i ocenić w ten sposób jego poprawność.</p>	<p>Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją. Określa liczbę prostych działań zawartych w algorytmie.</p>	<p>Rozumie, co to jest złożoność algorytmu i potrafi określić liczbę operacji wykonywanych na elementach zbioru w wybranym algorytmie sortowania.</p>	<p>Potrafi porównać złożoność różnych algorytmów tego samego zadania dla tych samych danych. Wie, kiedy algorytm jest uniwersalny.</p>	<p>Ocenia złożoność czasową i pamięciową algorytmu. Zna odpowiednie wzory.</p>

## 2. Realizacja algorytmów w wybranym języku programowania

### 2.1. Zasady programowania

2	3	4	5	6
<p>Zna klasyfikację języków programowania.</p> <p>Zna ogólną budowę programu i najważniejsze elementy języka – słowa kluczowe, instrukcje, wyrażenia, zasady składni.</p> <p>Potrafi zrealizować prosty algorytm liniowy w języku wysokiego poziomu; potrafi skompilować i uruchomić gotowy program.</p>	<p>Zapisuje program w czytelnej postaci – stosuje wcięcia, komentarze.</p> <p>Rozumie pojęcia: implementacja, kompilacja, uruchomienie, testowanie.</p> <p>Rozumie znaczenie i działanie podstawowych instrukcji wybranego języka programowania wysokiego poziomu.</p> <p>Rozróżnia i poprawia błędy kompilacji i błędy wykonania.</p> <p>Potrafi zrealizować algorytmy iteracyjne w języku wysokiego poziomu.</p>	<p>Potrafi prezentować złożone algorytmy (z podprogramami) w wybranym języku programowania. Zna rekurencyjne realizacje prostych algorytmów.</p> <p>Rozumie i stosuje zasady programowania strukturalnego.</p> <p>Wie, na czym polega różnica pomiędzy przekazywaniem parametrów przez zmienną i przez wartość w procedurach.</p> <p>Wie, jakie znaczenie ma zasięg działania zmiennej.</p> <p>Rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą.</p>	<p>Wie, jaka jest różnica między językiem wysokiego poziomu a językiem wewnętrznym; potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów.</p> <p>Potrafi realizować nawet bardzo złożone algorytmy, stosować procedury graficzne w realizacji skomplikowanych zadań – np. tworzyć własne animacje. Potrafi prezentować algorytmy rekurencyjne w postaci programu; potrafi zamienić rozwiązanie iteracyjne algorytmu na rekurencyjne. Zapisuje w postaci programu wybrane algorytmy sortowania.</p>	<p>Ocenia efektywność działania programu.</p> <p>Wie, na czym polega programowanie obiektowe i zdarzeniowe.</p> <p>Potrafi stosować techniki programowania dynamicznego lub programowania obiektowego.</p> <p>Zna i rozumie podobieństwa i różnice w strukturze programu zapisanego w różnych językach programowania – w deklaracji zmiennych i procedur, w składni i zasadach działania poszczególnych procedur.</p> <p>Sprawnie korzysta z dodatkowej, fachowej literatury.</p>

	Zna podstawowe zasady poprawnego programowania; testuje tworzone programy; wie, jak unikać problemów, takich jak np. zapętlenie się programu.	Zna zasady działania wybranych algorytmów sortowania.  Zna podstawowe procedury graficzne, potrafi narysować na ekranie wykres funkcji i podstawowe figury geometryczne.		
--	---	--	--	--

## 2.2. Dobór struktur danych do rozwiązywanego problemu

2	3	4	5	6
Wymienia przykłady prostych struktur danych. Potrafi zadeklarować zmienne typu liczbowego (całkowite, rzeczywiste) i stosować je w zadaniach.	Wie, czym jest zmienna w programie i co oznacza przypisanie jej konkretnej wartości. Rozróżnia struktury danych: proste i złożone. Podaje przykłady. Deklaruje typy złożone.	Potrafi zastosować łańcuchowy i tablicowy typ danych w zadaniach.	Rozumie, na czym polega dobór struktur danych do algorytmu. Potrafi zastosować rekordowy typ danych.	Zna dynamiczne struktury danych. Potrafi zastosować zmienne typu wskaźnikowego w zadaniach. Zna struktury listowe, np. stos, kolejkę, listę. Rozumie i potrafi zastosować typ obiektowy.

## 3. Zasady działania komputera i sieci komputerowych

### 3.1. System komputerowy

2	3	4	5	6
Określa następujące pojęcia: bit, bajt. Zna pojęcie systemu pozycyjnego.  Wymienia części składowe zestawu komputerowego, podaje ich parametry i przeznaczenie.  Rozróżnia rodzaje pamięci komputera, określa ich własności i przeznaczenie.  Wie, co to jest system operacyjny, i korzysta z jego podstawowych funkcji.  Wykonuje podstawowe operacje na plikach i folderach.	Wie, co to jest system binarny, i potrafi dokonać zamiany liczby z systemu dziesiętnego na binarny i odwrotnie.  Potrafi sklasyfikować środki (urządzenia) i narzędzia (oprogramowanie) technologii informacyjnej.  Wie, jak działa komputer. Wyjaśnia rolę procesora. Rozumie organizację pamięci komputerowych.  Potrafi omówić funkcje systemu operacyjnego.  Zna zasady ochrony plików. Potrafi nadać podstawowe atrybuty plikom, jak też wyszukiwać poszczególne pliki.	Potrafi wykonać działania arytmetyczne na liczbach binarnych (dodawanie i odejmowanie).  Zna system szesnastkowy i potrafi wykonać konwersję liczb binarnych na liczby w systemie szesnastkowym i odwrotnie.  Analizuje model logiczny komputera. Wie, co to jest kod ASCII. Potrafi wymienić rodzaje aktualnie używanych komputerów.  Zna metody wyszukiwania plików.	Potrafi omówić dokładnie działanie procesora. Potrafi wykonać dowolną konwersję pomiędzy systemem dziesiętnym, dwójkowym i szesnastkowym.  Zna sposób zapisu liczby całkowitej i rzeczywistej (zmiennoprzecinkowej).  Umie wymienić przynajmniej dwa systemy operacyjne i podać ich najważniejsze funkcje.  Zna zaawansowane metody wyszukiwania i odzyskiwania plików.  Zna przynajmniej jeden algorytm szyfrowania danych. Potrafi zaszyfrować i odszyfrować prosty tekst.	Zna operacje logiczne na liczbach binarnych i przesunięcia bitowe. Potrafi zapisać w języku programowania wysokiego poziomu algorytm konwersji liczb z dowolnego systemu pozycyjnego na inny. Wykonuje sprawnie operacje na liczbach zapisanych w różnych systemach pozycyjnych.  Potrafi odzyskać utracony plik, stosując zaawansowane metody.  Potrafi omówić różne systemy operacyjne, wskazać ich najważniejsze funkcje.  Samodzielnie wyszukuje informacje na temat kompresji i szyfrowania danych. Zna kilka sposobów szyfrowania informacji. Potrafi zapisać algorytm szyfrowania w postaci programu. Zna działanie algorytmu kompresji.

### 3.2. Sieci komputerowe, w tym Internet

2	3	4	5	6
<p>Zna pojęcie sieci komputerowej, potrafi wymienić jej rodzaje.</p> <p>Zna pojęcie logowania.</p> <p>Potrafi wymienić kilka cech pracy w sieci, odróżniających ją od pracy na autonomicznym komputerze.</p> <p>Zna kilka sposobów połączenia z Internetem.</p>	<p>Wymienia korzyści płynące z korzystania z sieci.</p> <p>Zna podstawowe klasy i topologie sieciowe.</p> <p>Potrafi wymienić urządzenia i elementy sieciowe oraz omówić ich ogólne przeznaczenie.</p> <p>Zna cechy systemu działającego w szkolnej pracowni.</p> <p>Orientuje się – w zakresie podstawowym – w działaniu Internetu.</p>	<p>Zna znaczenie protokołu w sieciach (w tym TCP/IP). Wie, na czym polega wymiana informacji w sieci.</p> <p>Zna zasady pracy w sieci, m.in. zasady udostępniania zasobów.</p> <p>Potrafi omówić zagrożenia płynące z sieci.</p> <p>Charakteryzuje różne połączenia z Internetem; potrafi omówić przesyłanie pakietów danych w Internecie.</p>	<p>Zna schemat działania sieci komputerowych.</p> <p>Potrafi wymienić zalety i wady różnych topologii sieci. Charakteryzuje topologie gwiazdy, magistrali i pierścienia.</p> <p>Zna podstawowe cechy systemu Linux.</p> <p>Umie z pomocą nauczyciela zrealizować małą sieć komputerową – skonfigurować jej składniki, udostępnić pliki, dyski, drukarki, dodać użytkowników.</p>	<p>Omawia szczegółowo model warstwowy sieci.</p> <p>Omawia różne systemy sieciowe. Dokońuje ich analizy porównawczej.</p> <p>Charakteryzuje system Linux.</p> <p>Potrafi samodzielnie zbudować małą sieć domową.</p>

### 3.3. Tendencje w rozwoju informatyki i jej zastosowań

2	3	4	5	6
<p>Potrafi omówić historię komputerów.</p> <p>Umie wskazać ogólny kierunek zmian w technologiach komputerowych.</p> <p>Zna podstawowe zasady netykiety.</p>	<p>Potrafi określić nowoczesne trendy w zastosowaniu urządzeń komputerowych.</p> <p>Jest w stanie omówić prawne i społeczne aspekty zastosowania informatyki.</p>	<p>Potrafi wskazać nowości w zakresie usług internetowych oraz odszukać informacje na temat najnowszych pomysłów na komputery.</p>	<p>Przygotowuje analizę porównawczą, pokazującą na przestrzeni wielu lat rozwój informatyki, w tym sieci komputerowych, oraz multimediów.</p>	<p>Wskazuje tendencje w rozwoju informatyki i jej zastosowania, dostrzegając przeobrażenia w tej dziedzinie w kraju i na świecie.</p>

## 4. Przetwarzanie danych w bazach danych

### 4.1. Projektowanie relacyjnej bazy danych

2	3	4	5	6
<p>Podaje obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, instytucji naukowych, społecznych i gospodarczych.</p> <p>Podaje przykłady programów do tworzenia baz danych.</p> <p>Potrafi wykonać podstawowe operacje na bazie danych przygotowanej w jednej tabeli (wprowadzanie, redagowanie, sortowanie, wyszukiwanie, prezentacja).</p> <p>Potrafi uporządkować bazę rosnąco lub malejąco według jednego lub kilku pól.</p>	<p>Rozumie metody przetwarzania danych na przykładzie gotowej bazy danych. Określa podstawowe pojęcia (rekord, pole, typ pola).</p> <p>Samodzielnie tworzy w jednej tabeli bazę danych, składającą się z kilku pól różnych typów. Projektuje przykładowy formularz i raport.</p> <p>Potrafi wykonywać operacje przetwarzania danych w bazie składającej się z kilku rekordów.</p> <p>Zna zasady przygotowania korespondencji seryjnej.</p>	<p>Projektuje relacyjną bazę danych składającą się z dwóch tabel połączonych relacją (na zadany temat). Projektuje formularz i raport według wskazówek nauczyciela.</p> <p>Zna zasady definiowania kluczy podstawowych.</p> <p>Drukuje wybrane rekordy, formularze i raporty.</p> <p>Łączy informacje z bazy danych z dokumentami innych programów, np. edytora tekstu czy arkusza kalkulacyjnego.</p>	<p>Potrafi wytłumaczyć pojęcie relacji.</p> <p>Projektuje relacyjną bazę danych składającą się z trzech lub większej liczby tabel. Samodzielnie ustala zawartość bazy (rodzaj informacji).</p> <p>Zna kilka rodzajów formularzy i raportów, w tym raporty w postaci wykresów.</p> <p>Umie zaprojektować samodzielnie wygląd formularza i raportu.</p> <p>Zna pojęcie indeksu. Odróżnia sortowanie od indeksowania.</p> <p>Potrafi w tworzonej bazie ustalić klucze indeksu.</p>	<p>Zna dokładnie wybrany program do projektowania baz danych.</p> <p>Potrafi samodzielnie zaprojektować bazę danych, korzystając z wybranego narzędzia (programu). Projekt bazy opiera na rzeczywistych informacjach, aby można było wykorzystać ją w praktyce, np. w szkole czy w domu.</p> <p>Sprawnie korzysta z dodatkowej, fachowej literatury.</p>

## 4.2. Wyszukiwanie informacji z użyciem języka zapytań

2	3	4	5	6
Wyszukuje informacje w bazie, korzystając wyłącznie z gotowych kwerend i narzędzi wbudowanych do programu.	Tworzy samodzielnie kwerendy (proste i złożone), korzystając z wbudowanych do programu narzędzi.	Zna podstawowe konstrukcje języka zapytań. Wie, co to jest język SQL. Potrafi przeanalizować przykład zapytania utworzonego w języku SQL. Z pomocą nauczyciela potrafi zapisać prostą kwerendę, korzystając z języka zapytań.	Zna zasady wyszukiwania informacji w bazie z wykorzystaniem języka zapytań. Potrafi zapisać złożone kwerendy, korzystając z wybranej instrukcji, np. SELECT; stosuje jej główne klauzule.	Opierając się na profesjonalnej literaturze, potrafi samodzielnie zapisywać złożone kwerendy z wykorzystaniem języka zapytań.

## 4.3. Realizacja projektu programistycznego, w tym przygotowanie dokumentów i raportów

2	3	4	5	6
Uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje. Bierze udział w testowaniu projektu.	Zna wszystkie etapy projektowania systemów informatycznych. Uczestniczy czynnie w poszczególnych etapach projektu, wykonując zlecane zadania szczegółowe.	Wie, co to jest system informatyczny. Potrafi omówić zakres prac na każdym etapie. Uczestniczy czynnie w analizie systemu informatycznego, przygotowuje dokumentację. Pracuje przy projektowaniu tabel, formularzy i raportów.	Wykonuje trudniejsze prace związane z projektowaniem bazy. Projektuje złożone kwerendy, formularze, raporty. Uczestniczy we wdrażaniu systemu informatycznego.	Potrafi wystąpić w roli koordynatora projektu. Przydziela zadania szczegółowe, dba o ich prawidłowe wykonanie, nadzoruje pracę innych, dba o dobrą atmosferę w grupie.

# 5. Wśród multimedialnych

## 5.1. Przetwarzanie informacji w różnych postaciach, m.in. graficznej i dźwiękowej

2	3	4	5	6
Wymienia programy do tworzenia i obróbki grafiki. Posługuje się jednym z nich w celu tworzenia własnych rysunków. Potrafi na kilka sposobów umieścić grafikę w dokumencie tekstowym, np. oblać rysunek tekstem. Wykonuje podstawowe operacje na rysunku, np. skalowanie, kadrowanie. Zna urządzenia multimedialne, wymienia przykładowe nazwy, określa ich ogólne przeznaczenie.	Zna sposoby reprezentacji obrazu i dźwięku w komputerze. Zna możliwości kilku wybranych programów do edycji obrazu i do tworzenia animacji. Zapisuje plik graficzny w różnych formatach (zna zastosowanie poszczególnych formatów, ich zalety i wady). Potrafi wybrać proste fragmenty obrazu i wykonać na nich różne operacje. Korzysta z różnych urządzeń multimedialnych, zna ich działanie, podaje ich przeznaczenie. Określa pojęcie: komputer multimedialny.	Rozróżnia grafikę wektorową i rastrową. Zna różne możliwości komputera w zakresie edycji obrazu, dźwięku, animacji i wideo. Zna pojęcia: RGB i CMYK. Potrafi stosować różne narzędzia malarskie i korekcyjne oraz wybrać odpowiedni tryb ich pracy. Orientuje się, co to jest rozdzielczość. Ustala rozdzielczość dla skanowanych i edytowanych obrazów. Posługuje się sprawnie wybranymi urządzeniami multimedialnymi.	Rozumie twórczy charakter tworzenia grafiki. Zna i stosuje w praktyce zaawansowaną obróbkę grafiki rastrowej. Przy użyciu odpowiednich narzędzi potrafi zaznaczyć fragmenty obrazu nawet o skomplikowanym kształcie. Potrafi zapisywać pliki multimedialne w różnych formatach, ze szczególnym uwzględnieniem formatów internetowych. Zna pojęcia: filtr, histogram, krzywa barw. Potrafi zdefiniować barwy i wykonać na nich operacje. Przekształca obraz – geometrycznie i z zastosowaniem filtrów.	Korzysta z profesjonalnej literatury dotyczącej przetwarzania multimedialnych. Potrafi samodzielnie odkrywać możliwości programów komputerowych w zakresie montażu wideo. Zna metody przechwytywania danych wideo. Zna zasady montażu filmu. Potrafi wykonać taki montaż. Zna sposoby udostępniania filmu innym osobom oraz metody konwersji między różnymi formatami.