

## Wymagania na oceny z przedmiotu Informatyka dla uczniów Liceum i Technikum w zakresie rozszerzonym

### A. Wokół komputera

**Uwaga:** Zgodnie z podstawą programową do informatyki dla szkół ponadpodstawowych: „**Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:**”

<b>Narzędzia i sposoby dbania o komputer</b>				
<b>dopuszczający</b>	<b>dostateczny</b>	<b>dobry</b>	<b>bardzo dobry</b>	<b>celujący</b>
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
potrafi sprawdzić zajętość dysku; usuwa zbędne pliki; odinstalowuje zbędne programy	zna możliwości wybranych narzędzi administracyjnych systemu Windows, m.in. Oczyszczanie dysku; dba, aby system operacyjny był na bieżąco aktualizowany	sprawdza ustawienia pamięci dyskowej komputera; wie, jak włączyć automatyczne zwalnianie miejsca na dysku twardym; wie, na czym polega proces defragmentacji dysku	wie, na czym polega działanie narzędzia Przywracanie systemu; wie, jak przyspieszyć pracę komputera, m.in. sprawdza aktualnie działające programy, wyłącza niepotrzebne programy; omawia, na czym powinna polegać konserwacja sprzętu komputerowego	potrafi zadbać o własny sprzęt komputerowy; samodzielnie zapoznaje się z możliwościami różnych narzędzi administracyjnych systemu Windows; odszukuje inne, niż omówione w temacie A1, możliwości dbania o sprzęt komputerowy; samodzielnie wykonuje niezbędne czynności konserwujące sprzęt komputerowy

## Planowanie rozbudowy i zakup nowego komputera

2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
zna i omawia parametry podstawowych elementów komputera; zna i przestrzega zasady bezpiecznej pracy z urządzeniami elektronicznymi, m.in. podczas demontażu obudowy komputera;	wie, co można zrobić, aby rozbudować komputer przenośny, m.in.: podłączenie modemu GSM, zewnętrznego dysku twardego, zwiększenie wielkości pamięci RAM; potrafi sprawdzić parametry komputera	wie, co można zrobić, aby rozbudować komputer stacjonarny, m.in.: wymiana kart rozszerzeń, zwiększenie wielkości pamięci RAM; omawia konfigurację oprogramowania dla komputera przeznaczonego do konkretnych zastosowań, np. do użytku domowego, do gier	potrafi zaprojektować konfigurację nowego komputera przeznaczonego do konkretnych zastosowań, np. do użytku domowego, do gier; potrafi zaprojektować konfigurację oprogramowania przed zakupem komputera przeznaczonego do konkretnych zastosowań, np. do użytku domowego, do gier; potrafi przygotować nośnik z dystrybucją Linuxa i uruchomić za jego pomocą komputer	samodzielnie projektuje rozbudowę komputera przenośnego i stacjonarnego; samodzielnie projektuje konfigurację oprogramowania przed zakupem komputera przeznaczonego do profesjonalnych zastosowań, np. dla grafika; dzieli się z innymi swoją wiedzą i doświadczeniami w/w zakresach; wyszukuje dodatkowe informacje; porównuje funkcjonalności wybranej dystrybucji systemu Linux i systemu Windows

## B. Wokół dokumentów komputerowych

**Uwaga:** Zgodnie z podstawą programową do informatyki dla szkół ponadpodstawowych: „*Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:*”

Tworzenie i edytowanie obrazów rastrowych				
2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
zna podstawowe możliwości programu do edycji obrazu rastrowego (programu GIMP); potrafi utworzyć prosty rysunek, stosując różne narzędzia malarskie i korekcyjne oraz wybrać odpowiedni tryb ich pracy	wykonuje operacje na obszarach selekcji, m.in.: wycinanie, kopiowanie, kadrowanie, obrysowywanie; stosuje poznane możliwości programu GIMP do edycji zdjęć	zna przeznaczenie i odpowiednio stosuje wybrane klawisze modyfikujące (i kombinacje klawiszy); potrafi wykonać proste modyfikacje barw: zmianę jasności, odwracanie kolorów, zmianę nasycenia, odcieni kolorów	zna pojęcia: <i>histogram</i> , <i>krzywa barw</i> ; tworzy własny pędzel; wykorzystuje tryb <b>Kolor</b> ; operuje na kanałach barw; używa krzywej jasności i funkcji histogramu; korzysta z narzędzia klonowania; w razie konieczności wyszukuje informacje w <b>Pomocy</b> ;	samodzielnie zapoznaje się z dodatkowymi możliwościami programu GIMP, w tym opcjami tworzenia pędzli; rozwiązuje trudniejsze zadanie, korzystając z narzędzia klonowania i innych; wyszukuje w Internecie informacje na temat tworzenia zdjęć panoramicznych; uczestniczy w konkursach dotyczących grafiki komputerowej

<b>Praca z warstwami obrazu i animacje</b>				
<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
poprawia jakość zdjęcia, stosując wybrane filtry; wie, na czym polega praca z warstwami	wykonuje przekształcenia obrazu (obroty, odbicia), tworząc obrazy w grafice rastrowej; stosuje filtry, m.in. poprawia ostrość obrazu, wykonuje proste fotomontaże, korzystając z warstw	stosuje filtry, m.in. wykonuje efekt zamglenia, usuwa szумы; wykonuje fotomontaże, korzystając z kilku warstw; wie, czym jest oderwane zaznaczenie i jak zakotwiczyć warstwę; przygotowuje animacje składające się z kilkunastu klatek, stosując pracę na warstwach; zapisuje animację w formacie GIF	samodzielnie tworzy rysunki, powiela warstwy i stosuje możliwości przekształceń, aby utworzyć kolejną klatkę animacji; wykorzystując dodatkowe możliwości programu GIMP, rozwiązuje różne zadania, edytując obrazy, tworząc fotomontaże korzystając z poznanych możliwości programu GIMP, projektuje dwuwymiarową wizualizację lub animację	wyszukuje w Internecie dodatkowe informacje na temat korzystania z filtrów w programie GIMP; wymyśla temat fotomontażu i przygotowuje animacje według własnych pomysłów, korzystając z różnych możliwości programu do tworzenia animacji; projektuje dwuwymiarową wizualizację lub animację, wykorzystując dodatkowe możliwości programu GIMP

### C. Wokół algorytmiki i programowania

**Uwaga:** Zgodnie z podstawą programową do informatyki dla szkół ponadpodstawowych: „**Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:**”

Prezentacja algorytmu liniowego w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna sposoby prezentacji algorytmów;</p> <p>testuje działanie algorytmu liniowego zapisanego w postaci listy kroków i przedstawionego w postaci schematu blokowego;</p> <p>potrafi narysować (odręcznie) schemat blokowy algorytmu liniowego;</p> <p>zapisuje prosty algorytm liniowy w wybranym języku programowania</p>	<p>zna pojęcie specyfikacji zadania i potrafi zapisać specyfikację zadania;</p> <p>zna i stosuje zasady tworzenia listy kroków;</p> <p>przedstawia algorytm liniowy w postaci listy kroków;</p> <p>zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym;</p> <p>podczas rysowania schematów blokowych potrafi wykorzystać <b>Kształty</b> z edytora tekstu;</p> <p>pisze programy</p>	<p>przedstawia dokładną specyfikację dowolnego zadania.</p> <p>analizuje poprawność budowy schematu blokowego;</p> <p>objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych;</p> <p>realizuje przykładowy algorytm liniowy w wybranym języku programowania na podstawie ich list kroków i schematów blokowych;</p> <p>testuje program dla różnych danych</p>	<p>potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych;</p> <p>potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego;</p> <p>analizuje działanie algorytmu dla przykładowych danych;</p> <p>potrafi wskazać i poprawić błędy w liście kroków czy w schemacie blokowym;</p> <p>Samodzielnie pisze program realizujący algorytm liniowy.</p>	<p>zapisuje specyfikację zadania i przedstawia w postaci listy kroków i schematu blokowego rozwiązanie trudniejszego zadania;</p> <p>w wybranym języku programowania (C++ lub Python) pisze trudniejsze programy realizujące algorytmy przedstawione w postaci list kroków i schematów blokowych;</p> <p>uczestniczy w konkursach/olimpiadach informatycznych</p>

	komputerowe realizujące dane algorytmy na podstawie ich list kroków i/lub schematów blokowych			
--	---	--	--	--

## Prezentacja algorytmu z warunkami w wybranej notacji

2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
<p>określa sytuacje warunkowe;                      podaje przykłady zadań, w których występują sytuacje warunkowe;                      wie, jaka figura reprezentuje sytuacje warunkową;                      potrafi narysować (odręcznie) schemat blokowy nietrudnego algorytmu z warunkiem prostym</p>	<p>analizuje działanie algorytmu z warunkami zapisanego w postaci graficznej.                      zna i stosuje zasady tworzenia listy kroków algorytmu z warunkami;                      potrafi narysować schemat blokowy algorytmu z warunkami, korzystając z wybranego narzędzia;                      pisze program w wybranym języku programowania realizujący algorytm z warunkami prostymi; zna i stosuje instrukcję warunkową</p>	<p>przedstawia algorytm z warunkami w postaci listy kroków;                      analizuje listę kroków i schemat blokowy algorytmu z warunkami, testując go dla wybranych danych;                      potrafi zapisać warunek złożony;                      korzystając z przykładu, zapisuje w postaci programu algorytm z warunkami złożonymi;                      zna i omawia warunek istnienia trójkąta</p>	<p>Zapisuje algorytmy z pętlą zagnieżdżoną.                      testuje działanie algorytmu z warunkami zagnieżdżonymi zapisanego w postaci listy kroków na wybranych przykładach danych;                      buduje schemat blokowy algorytmu sprawdzania warunku trójkąta na podstawie listy kroków;                      zapisuje w postaci programu algorytm z warunkami zagnieżdżonymi</p>	<p>wymyśla samodzielnie problem z warunkami zagnieżdżonymi, zapisuje jego specyfikację, listę kroków, tworzy schemat blokowy i program w wybranym języku programowania korzystając z dodatkowych źródeł, znajduje inny, niż podany w podręczniku, sposób sprawdzenia, czy z danych trzech odcinków można zbudować trójkąt i zapisuje ten algorytm w postaci programu komputerowego;                      w wybranym języku programowania pisze trudniejsze programy realizujące algorytmy z warunkami (w tym zagnieżdżonymi) przedstawione w postaci list kroków i schematów</p>

				blokowych
--	--	--	--	-----------



Prezentacja algorytmu iteracyjnego w wybranej notacji				
2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
<p>analizuje listę kroków algorytmu iteracyjnego, testując go dla wybranych danych;</p> <p>analizuje schemat algorytmu iteracyjnego, testując go dla wybranych danych;</p>	<p>zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego; podaje przykłady algorytmów iteracyjnych;</p> <p>tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą; testuje rozwiązanie dla wybranych danych;</p> <p>zna i stosuje instrukcję iteracyjną <code>for</code> w wybranym języku programowania;</p> <p>zapisuje prosty algorytm iteracyjny w postaci programu w wybranym języku programowania</p>	<p>analizuje algorytmy, w których występują powtórzenia (iteracje);</p> <p>ocenia zgodność algorytmu ze specyfikacją;</p> <p>analizuje listę kroków i schemat blokowy algorytmu z pętlą zagnieżdżoną, testując go dla wybranych danych;</p> <p>zapisuje algorytm iteracyjny w postaci programu w wybranym języku programowania</p>	<p>zapisuje w postaci listy kroków algorytmów z warunkami i iteracyjnie;</p> <p>pisze listę kroków i tworzy schemat blokowy algorytmu z pętlą zagnieżdżoną;</p> <p>ocenia zgodność algorytmu ze specyfikacją problemu;</p> <p>wie, kiedy należy zastosować pętlę zagnieżdżoną;</p> <p>zapisuje w postaci programu wybrany algorytm z pętlą zagnieżdżoną;</p> <p>testuje program dla wybranych danych</p>	<p>przestrzega zasad zapisu algorytmów w zadanej postaci (notacji);</p> <p>stosuje listy kroków i schematy blokowe w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia;</p> <p>wskazuje podobieństwa i różnice dotyczące działania instrukcji warunkowych i iteracyjnej <code>for</code> w różnych językach programowania;</p> <p>zapisuje trudniejsze algorytmy iteracyjne w wybranym języku programowania; zapisuje programy w czytelnej postaci – stosuje wcięcia, komentarze;</p> <p>uczestniczy w konkursach/olimpiadach</p>

				informatycznych
--	--	--	--	-----------------

<b>Funkcje w wybranym języku programowania</b>				
<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
<p>Wie, czym są podprogramy;</p> <p>W wybranym języku programowania definiuje prostą funkcję niezwracającą wartości bez parametrów.</p> <p>Wie, jak wywołać funkcję niezwracającą wartości bez parametrów w programie.</p> <p>Wywołuje taką funkcję w programie.</p>	<p>Wie, na czym polega programowanie strukturalne;</p> <p>Pisze programy, stosując funkcje.</p> <p>Definiuje funkcje niezwracające wartości bez parametrów i z parametrami. Stosuje te funkcje w programach.</p> <p>Wyjaśnia pojęcia zmienna lokalna i zmienna globalna.</p>	<p>Wymienia modele programowania;</p> <p>Rozumie i stosuje zasady programowania strukturalnego.</p> <p>Wyjaśnia, czym są parametry funkcji.</p> <p>Wyjaśnia, czym jest funkcja zwracająca wartość. Porównuje do funkcji niezwracającej wartości. Wskazuje różnice.</p> <p>Wie, na czym polega wywołanie funkcji z parametrami i wywołuje taką funkcję w programach.</p> <p>Definiuje funkcje zwracające wartości bez parametrów i z parametrami oraz stosuje je w programach.</p> <p>Wie, co to jest zasięg</p>	<p>Omawia modele programowania;</p> <p>Rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą.</p> <p>Wyjaśnia, czym różni się programowanie zstępujące od wstępującego.</p> <p>Definiuje funkcje niezwracające wartości i zwracające wartość bez parametrów i z parametrami oraz stosuje je w programach.</p> <p>Wyjaśnia, na czym polega przesłanianie zmiennych globalnych.</p> <p>Wyjaśnia, na czym polega przesłanianie parametrów funkcji.</p> <p>Modyfikuje programy, stosuje zmienne globalne i lokalne, objaśnia</p>	<p>Porównuje modele programowania, wskazując różnice.</p> <p>Sprawnie definiuje i stosuje funkcje niezwracające wartości i zwracające wartość bez parametrów i z parametrami w programach.</p> <p>Rozwiązuje przykładowe zadania z matury i olimpiady informatycznej.</p> <p>Potrafi, na przykładzie programu utworzonego według własnego pomysłu, wyjaśnić różnice w stosowaniu zmiennych lokalnych i globalnych, omówić zasięg zmiennych i przesłanianie zmiennych.</p> <p>Potrafi, na przykładzie programu utworzonego według własnego pomysłu, wyjaśnić przesłanianie</p>

		<p>zmiennej i jakie ma znaczenie.</p> <p>Deklaruje odpowiednio zmienne lokalne i globalne w programach.</p>	<p>otrzymane wyniki.</p>	<p>parametrów funkcji.</p>
--	--	---	--------------------------	----------------------------

## Tablice w wybranym języku programowania

2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
<p>Analizuje i omawia gotowe przykłady programów, w których są użyte tablice lub listy.</p> <p>Potrafi wyjaśnić, dlaczego do rozwiązania niektórych zadań należy użyć tablic (list).</p>	<p>Zna pojęcia: <i>tablica</i>, <i>lista</i>, <i>zmienna indeksowana</i>.</p> <p>Na bazie przykładów z podręcznika, deklaruje tablicę i/lub listę, wczytuje i wyprowadza elementy tablicy i/lub listy, definiując odpowiednie funkcje w wybranym języku programowania.</p>	<p>Rozróżnia struktury danych: proste i złożone. Podaje przykłady.</p> <p>Wczytuje i wyprowadza elementy tablicy i/lub listy. Definiuje odpowiednie funkcje.</p> <p>Wie, jak odwołać się do elementu tablicy i/lub listy. Potrafi zastosować tablicę (listę) w zadaniach oraz modyfikować program, znaleźć błędy i je poprawić.</p>	<p>Rozumie, na czym polega dobór struktur danych do algorytmu i tworzy programy, dobierając odpowiednie struktury danych do programu.</p> <p>Deklaruje tablicę dwuwymiarową w języku C++. Definiuje odpowiednie funkcje.</p> <p>Definiuje różne listy (w tym dwuwymiarowe) w języku Python.</p> <p>Tworzy programy, dobierając odpowiednie struktury danych do programu.</p>	<p>Omawia podobieństwa i różnice w definiowaniu tablic w języku C++ i list w języku Python</p> <p>Stosuje w programach listy jednowymiarowe i dwuwymiarowe, odpowiednio dobierając określoną strukturę danych (tu: rodzaj listy) do algorytmu.</p> <p>Rozwiązuje przykładowe zadania z konkursów i olimpiady informatycznej, w których należy zastosować tablice i/lub listy i funkcje.</p> <p>Dobiera najlepszy algorytm i odpowiednie struktury danych do rozwiązania postawionego problemu.</p>

Stosowanie instrukcji iteracyjnych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>Analizuje i omawia działanie gotowych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli <b>for</b>.</p> <p>Analizuje i omawia działanie gotowych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli <b>while</b>.</p>	<p>Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady.</p> <p>Zna postać i działanie instrukcji iteracyjnej <b>while</b> w językach C++ i/lub Python i stosuje ją w tworzonych programach komputerowych.</p>	<p>Zna sposoby zakończenia iteracji. Określa kroki iteracji.</p> <p>Stosuje instrukcję <b>while</b> w programach komputerowych.</p> <p>W języku C++ stosuje instrukcję <b>do ... while</b> w programach komputerowych.</p> <p>Zna różne sposoby określania liczby iteracji w języku Python, np. poprzez powtarzanie poleceń zawartych wewnątrz pętli dla konkretnych wartości lub poprzez podanie liczby powtórzeń z zastosowaniem funkcji <b>range ()</b>.</p>	<p>Rozumie różnicę pomiędzy instrukcją <b>for</b> a instrukcją <b>while</b> w wybranym języku programowania.</p> <p>Modyfikuje programy, zamieniając pętlę <b>for</b> na <b>while</b> i odwrotnie. Ocenia program po zmianie (styl, czytelność).</p> <p>Pisze programy, stosując poznane instrukcje iteracyjne.</p> <p>Dobiera odpowiednią technikę algorytmiczną i odpowiednie struktury danych do rozwiązywanego problemu.</p>	<p>Potrafi samodzielnie zastosować odpowiedni rodzaj instrukcji pętli w tworzonym programie.</p> <p>Potrafi samodzielnie dobrać odpowiednią instrukcję <b>while</b> lub <b>do... while</b>.</p> <p>Omawia podobieństwa i różnice w działaniu wszystkich omówionych instrukcji pętli w języku C++.</p> <p>Sprawnie pisze trudniejsze programy, stosując poznane instrukcje iteracyjne.</p> <p>Wymyśla samodzielnie problem iteracyjny, formułuje zadanie, pisze jego specyfikację i listę kroków oraz zapisuje w wybranym języku programowania.</p>



## Iteracyjna realizacja wybranych algorytmów, w tym algorytmów wyszukiwania

2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
<p>Potrafi odróżnić algorytm liniowy od algorytmu iteracyjnego.</p> <p>Omawia algorytm znajdowania elementu najmniejszego. Analizuje listę kroków i rysuje schemat blokowy na jej podstawie.</p> <p>Analizuje i omawia gotową listę kroków i schemat blokowy algorytmu Euklidesa w jednej z wersji. Testuje algorytm na podstawie listy lub schematu.</p>	<p>Zna przykładowe algorytmy na liczbach naturalnych; Wie, na czym polega metoda wyszukiwania liniowego i przez połowienie.</p> <p>Zapisuje algorytm znajdowania najmniejszego (największego) elementu w postaci programu.</p> <p>Zna iteracyjną postać algorytmu Euklidesa w wersji z odejmowaniem.</p> <p>Potrafi napisać listy kroków algorytmu i narysować schemat blokowy Euklidesa w wersji z odejmowaniem.</p> <p>Zna metodę „dziel i zwyciężaj”.</p> <p>Zapisuje algorytm Euklidesa w wersji z odejmowaniem w postaci</p>	<p>Potrafi omówić algorytm naiwny i optymalny jednoczesnego znajdowania największego i najmniejszego elementu w zbiorze.</p> <p>Zna iteracyjną postać algorytmu Euklidesa z resztą z dzielenia. Pisze listę kroków tego algorytmu.</p> <p>Potrafi narysować schemat blokowy algorytmu Euklidesa w wersji z resztą z dzielenia.</p> <p>Zapisuje algorytm Euklidesa w wersji z resztą z dzielenia w postaci programu w wybranym języku programowania.</p> <p>Wyjaśnia na przykładzie różnicę między wersją algorytmu Euklidesa z odejmowaniem a wersją z resztą z dzielenia.</p>	<p>Określa liczbę porównań w algorytmie naiwnym i optymalnym znajdowania największego i najmniejszego elementu w zbiorze. Porównuje otrzymane wyniki.</p> <p>Pisze listę kroków algorytmu jednoczesnego znajdowania minimum i maksimum z wykorzystaniem metody dziel i zwyciężaj. Zapisuje ten algorytm w postaci programu w języku C++ i/lub Python.</p> <p>Omawia zastosowanie schematu Hornera do obliczania wartości wielomianu; pisze listę kroków, rysuje schemat blokowy tego algorytmu i pisze program realizujący algorytm obliczania wartości wielomianu</p>	<p>Podaje przykłady problemów, w których można zastosować wyszukiwanie liniowe lub przez połowienie.</p> <p>Pisze trudniejsze programy komputerowe, w których wykorzystuje poznane algorytmy.</p> <p>Korzystając z dodatkowych źródeł, wyszukuje informacje o zastosowaniu metody „dziel i zwyciężaj” oraz pisze program według własnego pomysłu pokazujący zastosowanie tej metody.</p> <p>Samodzielnie zapoznaje się ze schematem Hornera i zapisuje go w postaci programu, dopierając poprawne struktury danych.</p> <p>Korzystając z dodatkowych źródeł, omawia przykłady</p>



	programu w wybranym języku programowania.		według schematu Hornera. Programując w/w algorytmy, definiuje odpowiednie funkcje, dobiera struktury danych. Dbą o stosowanie podstawowych zasad programowania.	zastosowań algorytmu Euklidesa i poznaje trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, algorytm znajdowania liczb bliźniaczych. Potrafi zapisać je w języku programowania.
--	---	--	---	--

## Elementy analizy algorytmów

2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
<p>Wymienia własności algorytmów.</p> <p>Potrafi przeanalizować przebieg prostego algorytmu zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego poprawność.</p>	<p>Zna i omawia własności algorytmów.</p> <p>Wie, kiedy algorytm jest poprawny.</p> <p>Potrafi przeanalizować przebieg algorytmu (np. obliczania silni) zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego poprawność.</p> <p>Analizuje program wybranego algorytmu (np. obliczania silni) i ocenia jego poprawność.</p>	<p>Wie, jak sprawdzić, czy algorytm jest skończony.</p> <p>Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją.</p> <p>Określa liczbę prostych działań zawartych w algorytmie.</p> <p>Określa liczbę prostych działań zawartych w algorytmie.</p> <p>Potrafi poprawić program, który jest niepoprawny.</p>	<p>Wie, kiedy algorytm jest skończony.</p> <p>Potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego skończoność.</p> <p>Analizuje program realizujący wybrany algorytm i ocenia jego skończoność. Modyfikuje program, aby działał poprawnie.</p> <p>Sprawdza poznane własności algorytmów, rozwiązując zadania, m.in. uzasadnia skończoność algorytmu znajdowania największego wspólnego dzielnika (NWD) dwóch liczb naturalnych.</p>	<p>Potrafi samodzielnie ocenić poprawność i skończoność wybranych algorytmów.</p> <p>Potrafi samodzielnie wymyśleć zadanie (problem), napisać specyfikację zadania, listę kroków i program realizujący to zadanie.</p> <p>Korzysta samodzielnie z dodatkowej literatury fachowej.</p> <p>Poznane dodatkowe możliwości wybranego języka programowania (np. standardowe funkcje) i stosuje w programach.</p> <p>Rozwiązuje przykładowe zadania z olimpiady informatycznej.</p>

			Oblicz liczbę operacji porównania w algorytmie wyboru minimum z tablicy zawierającej $n$ losowo uporządkowanych liczb.	
--	--	--	--	--

## D. Wokół Internetu i projektów

**Uwaga:** Zgodnie z podstawą programową do informatyki dla szkół ponadpodstawowych: „*Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:*”

Praca w chmurze i zadania projektowe				
2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
Zna etapy pracy nad projektem i bierze udział w pracy grupowej jako członek zespołu. Uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje.	Planuje temat projektu. Bierze aktywny udział w pracy grupowej jako członek zespołu, gromadząc i selekcjonując materiały do projektu. Bierze udział w testowaniu projektu. Uczestniczy w przygotowaniu dokumentacji projektu, korzystając z edytora tekstu; ewentualnie z wybranego szablonów.	Korzystając z chmury, potrafi udostępnić pliki, linki do folderu; umożliwić współdzielenie danego folderu. Realizuje projekt na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania; przygotowuje dokumentację projektu. Współpracuje w grupie, wykonując projekt na temat projektowania zakupu nowego zestawu komputerowego oraz oprogramowania dla ucznia szkoły ponadpodstawowej; oraz wpływu trendów w	Korzysta z oprogramowania dostępnego w chmurze, m.in. tworzy dokumenty w edytorze tekstu i umieszcza w chmurze. Współdzieli je z innymi współużytkownikami. Przygotowuje wybrane zadanie szczegółowe zgodnie z etapami przygotowania projektu. Gromadzi materiały i inne pomoce, opracowuje dokumentację projektu, wykorzystując m.in. możliwości pracy w chmurze. Udostępnia pliki, linki do folderu.	Sprawnie posługuje się środowiskiem przeznaczonym do współpracy i realizacji projektów zespołowych, w tym środowiskiem w chmurze. Potrafi pełnić funkcję koordynatora grupy. Koordynuje wykonywanie zadań szczegółowych na poszczególnych etapach. Zarządza folderami (współdzieleniem) i pracą nad dokumentami, w tym dokumentacją projektu. Ustala sposób prezentacji projektu i wyznacza osobę (osoby) do prezentacji. W miarę możliwości

		historycznym rozwoju pojęć i metod informatyki oraz technologii na możliwości rozwoju grafiki komputerowej.	Wyszukuje informacje na e-platformach do e-nauczania. Prezentuje wykonane zadanie projektowe wykorzystaniem projektora innych przygotowanych pomocy lub materiałów. Inicjuje dyskusje.	współtworzy zasoby udostępniane na platformach do e-nauczania. Przygotowuje projekt na wybrany przez siebie temat.
--	--	---	--	---