

**Wymagania na poszczególne  
oceny szkolne**

**Informatyka- poziom  
rozszerzony**

**LICEUM**

**OGÓLNOKSZTAŁCĄCE**

## Spis treści

Moduł A. Wokół komputera i sieci komputerowych .....	3
Kompresja danych .....	3
Sztuczna inteligencja .....	4
Moduł B. Wokół dokumentów komputerowych.....	5
Tworzenie i edytowanie obrazów rastrowych .....	5
Praca z warstwami obrazu i animacje .....	6
Moduł C. Wokół algorytmiki i programowania .....	7
Prezentacja algorytmu liniowego w wybranej notacji .....	7
Prezentacja algorytmu z warunkami w wybranej notacji .....	8
Prezentacja algorytmu iteracyjnego w wybranej notacji .....	9
Funkcje w wybranym języku programowania .....	10
Tablice w wybranym języku programowania .....	11
Stosowanie instrukcji iteracyjnych .....	12
Iteracyjna realizacja wybranych algorytmów, w tym algorytmów wyszukiwania .....	13
Elementy analizy algorytmów .....	14
Moduł D. Wokół Internetu i projektów.....	15
Praca w chmurze i zadania projektowe .....	15

## Moduł A. Wokół komputera i sieci komputerowych

Kompresja danych				
2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
rozumie, na czym polega kompresja danych i w jakim celu się ją stosuje; zna wzór na obliczenie współczynnika kompresji	wymienia i omawia rodzaje kompresji – bezstratną i stratną; potrafi obliczyć współczynnik kompresji; zna pojęcie <i>przepływności</i>	omawia algorytmy kompresji stratnej; omawia algorytmy kompresji bezstratnej (statyczne, słownikowe)	dokonyuje kompresji informacji, m.in. stosuje w przykładowych ćwiczeniach algorytmy kompresji bezstratnej; objaśnia różnice między kompresją stratną i bezstratną tekstów, obrazów, dźwięków i filmów	wykonuje samodzielnie zadania dla zainteresowanych

Sztuczna inteligencja				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie czym jest sztuczna inteligencja;</p> <p>potrafi wymienić obszary zastosowania sztucznej inteligencji (SI);</p>	<p>umie przeanalizować prosty system SI w arkuszu kalkulacyjnym;</p> <p>potrafi wytłumaczyć schemat sieci neuronowej;</p> <p>konstruuje zapytania w systemie ChatGPT (lub podobnym);</p> <p>tworzy proste grafiki w systemie tworzenia obrazu</p>	<p>konstruuje rozwinięte zapytania w systemie ChatGPT (lub podobnym);</p> <p>samodzielnie korzysta z systemów tworzenia obrazów</p>	<p>wskazuje zalety i wady z korzystania z systemów wykorzystujących SI;</p> <p>korzysta z różnych systemów tworzenia obrazów</p>	<p>wyszukuje w Internecie inne systemy wykorzystujące sztuczną inteligencję oraz ocenia ich przydatność w różnych dziedzinach;</p> <p>prowadzi dyskusje o zastosowaniu SI i potrafi prawidłowo uargumentować swoje zdanie</p>

## Moduł B. Wokół dokumentów komputerowych

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

Praca z warstwami obrazu i animacje				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
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 szumy;  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

## Moduł C. Wokół algorytmiki i programowania

Prezentacja algorytmu liniowego w wybranej notacji				
2	3	4	5	6
<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>	<b>Uczeń:</b>
<p>zna sposoby prezentacji algorytmów;</p> <p>testuje działanie algorytmu liniowego zapisanego w postaci listy kroków;</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>pisze programy komputerowe realizujące dane algorytmy na podstawie ich list kroków</p>	<p>przedstawia dokładną specyfikację dowolnego zadania;</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;</p> <p>testuje program dla różnych danych</p>	<p>potrafi przeprowadzić szczegółową analizę poprawności listy kroków;</p> <p>analizuje działanie algorytmu dla przykładowych danych;</p> <p>potrafi wskazać i poprawić błędy w liście kroków;</p> <p>samodzielnie pisze program realizujący algorytm liniowy</p>	<p>zapisuje specyfikację zadania i przedstawia w postaci listy kroków rozwiązanie trudniejszego zadania;</p> <p>w wybranym języku programowania (C++ lub Python) pisze trudniejsze programy realizujące algorytmy przedstawione w postaci list kroków;</p> <p>uczestniczy w konkursach/olimpiadach informatycznych</p>

Prezentacja algorytmu z warunkami w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>określa sytuacje warunkowe; podaje przykłady zadań, w których występują sytuacje warunkowe; potrafi zapisać w liście kroków nietrudny algorytm z warunkiem prostym</p>	<p>analizuje działanie algorytmu z warunkami; zna i stosuje zasady tworzenia listy kroków algorytmu z warunkami; 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; zapisuje w postaci programu algorytm z warunkami zagnieżdżonymi</p>	<p>wymyśla samodzielnie problem z warunkami zagnieżdżonymi, zapisuje jego specyfikację, listę kroków 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</p>

Prezentacja algorytmu iteracyjnego w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>analizuje listę kroków algorytmu iteracyjnego, testując go dla wybranych danych</p>	<p>zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego;</p> <p>podaje przykłady algorytmów iteracyjnych;</p> <p>tworzy algorytm z warunkiem prostym i pętlą;</p> <p>testuje rozwiązanie dla wybranych danych;</p> <p>zna i stosuje instrukcję iteracyjną <b>for</b> 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 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 iteracyjne;</p> <p>pisze listę kroków 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 w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia;</p> <p>wskazuje podobieństwa i różnice dotyczące działania instrukcji warunkowych i iteracyjnej <b>for</b> w różnych językach programowania;</p> <p>zapisuje trudniejsze algorytmy iteracyjne w wybranym języku programowania;</p> <p>zapisuje programy w czytelnej postaci – stosuje wcięcia, komentarze;</p> <p>uczestniczy w konkursach/olimpiadach informatycznych</p>

Funkcje w wybranym języku programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<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;</p> <p>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 i 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 zmiennej i jakie ma znaczenie;</p> <p>deklaruje odpowiednio zmienne lokalne i globalne w programach</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 przesłanie zmiennych globalnych;</p> <p>wyjaśnia, na czym polega przesłanie parametrów funkcji;</p> <p>modyfikuje programy, stosuje zmienne globalne i lokalne, objaśnia otrzymane wyniki</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 przesłanie zmiennych;</p> <p>potrafi, na przykładzie programu utworzonego według własnego pomysłu, wyjaśnić przesłanie parametrów funkcji</p>

Tablice w wybranym języku programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<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, deklaruje tablicę i/lub listę, wczytuje i wyprowadza elementy tablicy i/lub listy, definiując odpowiednie funkcje w wybranym języku programowania</p>	<p>rozdziela struktury danych proste i złożone oraz podaje ich przykłady;</p> <p>wczytuje i wyprowadza elementy tablicy i/lub listy;</p> <p>definiuje odpowiednie funkcje;</p> <p>wie, jak odwołać się do elementu tablicy i/lub listy;</p> <p>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++;</p> <p>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;</p> <p>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
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>potrafi odróżnić algorytm liniowy od algorytmu iteracyjnego;</p> <p>omawia algorytm znajdowania elementu najmniejszego oraz analizuje listę kroków;</p> <p>analizuje i omawia gotową listę kroków algorytmu Euklidesa w jednej z wersji, testuje algorytm na podstawie listy</p>	<p>zna przykładowe algorytmy na liczbach naturalnych;</p> <p>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 Euklidesa w wersji z odejmowaniem;</p> <p>zna metodę „dziel i zwyciężaj”;</p> <p>zapisuje algorytm Euklidesa w wersji z odejmowaniem w postaci programu w wybranym języku programowania</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>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 tego algorytmu i pisze program realizujący algorytm obliczania wartości wielomianu według schematu Hornera;</p> <p>programując w/w algorytmu, definiuje odpowiednie funkcje, dobiera struktury danych, dba o stosowanie podstawowych zasad programowania</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, dobierając poprawne struktury danych;</p> <p>korzystając z dodatkowych źródeł, omawia przykłady zastosowań algorytmu Euklidesa i poznaje trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, algorytm znajdowania liczb bliźniaczych, potrafi zapisać je w języku programowania</p>

Elementy analizy algorytmów				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wymienia własności algorytmów;  potrafi przeanalizować przebieg prostego algorytmu zapisanego w postaci listy kroków dla przykładowych danych i ocenić w ten sposób jego poprawność	zna i omawia własności algorytmów;  wie, kiedy algorytm jest poprawny;  potrafi przeanalizować przebieg algorytmu (np. obliczania silni) zapisanego w postaci listy kroków dla przykładowych danych i ocenić w ten sposób jego poprawność;  analizuje program wybranego algorytmu (np. obliczania silni) i ocenia jego poprawność	wie, jak sprawdzić, czy algorytm jest skończony;  potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją;  określa liczbę prostych działań zawartych w algorytmie;  określa liczbę prostych działań zawartych w algorytmie;  potrafi poprawić program, który jest niepoprawny	wie, kiedy algorytm jest skończony;  potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy kroków dla przykładowych danych i ocenić w ten sposób jego skończoność;  analizuje program realizujący wybrany algorytm i ocenia jego skończoność;  modyfikuje program, aby działał poprawnie;  sprawdza poznane własności algorytmów, rozwiązując zadania, m.in. uzasadnia skończoność algorytmu znajdowania największego wspólnego dzielnika (NWD) dwóch liczb naturalnych;  oblicza liczbę operacji porównania w algorytmie wyboru minimum z tablicy zawierającej $n$ losowo uporządkowanych liczb	potrafi samodzielnie ocenić poprawność i skończoność wybranych algorytmów;  potrafi samodzielnie wymyśleć zadanie (problem), napisać specyfikację zadania, listę kroków i program realizujący to zadanie;  korzysta samodzielnie z dodatkowej literatury fachowej;  poznane dodatkowe możliwości wybranego języka programowania (np. standardowe funkcje) i stosuje w programach;  rozwiązuje przykładowe zadania z olimpiady informatycznej

## Moduł D. Wokół Internetu i projektów

Praca w chmurze i zadania projektowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna etapy pracy nad projektem i bierze udział w pracy grupowej jako członek zespołu;</p> <p>uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje</p>	<p>planuje temat projektu;</p> <p>bierze aktywny udział w pracy grupowej jako członek zespołu, gromadząc i selekcjonując materiały do projektu;</p> <p>bierze udział w testowaniu projektu;</p> <p>uczestniczy w przygotowaniu dokumentacji projektu, korzystając z edytora tekstu, ewentualnie z wybranego szablonu</p>	<p>korzystając z chmury, potrafi udostępnić pliki, linki do folderu, umożliwić współdzielenie danego folderu;</p> <p>realizuje projekt na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania, przygotowuje dokumentację projektu;</p> <p>współpracuje w grupie, wykonując projekt na temat projektowania zakupu nowego zestawu komputerowego oraz oprogramowania dla ucznia szkoły ponadpodstawowej oraz wpływu trendów w historycznym rozwoju pojęć i metod informatyki oraz technologii na możliwości rozwoju grafiki komputerowej</p>	<p>korzysta z oprogramowania dostępnego w chmurze, m.in. tworzy dokumenty w edytorze tekstu i umieszcza w chmurze, współdzieli je z innymi współużytkownikami;</p> <p>przygotowuje wybrane zadanie szczegółowe zgodnie z etapami przygotowania projektu;</p> <p>gromadzi materiały i inne pomoce, opracowuje dokumentację projektu, wykorzystując m.in. możliwości pracy w chmurze;</p> <p>udostępnia pliki, linki do folderu;</p> <p>wyszukuje informacje na e-platformach do e-nauczania;</p> <p>prezentuje wykonane zadanie projektowe i inicjuje dyskusje</p>	<p>sprawnie posługuje się środowiskiem przeznaczonym do współpracy i realizacji projektów zespołowych, w tym środowiskiem w chmurze;</p> <p>potrafi pełnić funkcję koordynatora grupy;</p> <p>koordynuje wykonywanie zadań szczegółowych na poszczególnych etapach;</p> <p>zarządza folderami (współdzieleniem) i pracą nad dokumentami, w tym dokumentacją projektu;</p> <p>ustala sposób prezentacji projektu i wyznacza osobę (osoby) do prezentacji;</p> <p>w miarę możliwości współtworzy zasoby udostępniane na platformach do e-nauczania;</p> <p>przygotowuje projekt na wybrany przez siebie temat</p>