

Opis założonych osiągnięć ucznia – wymagania na poszczególne oceny szkolne „matrix” -programowanie i grafika komputerowa

Uczeń:	2	3	4	5	6
	Wie, co to jest algorytm. Określa dane do zadania oraz wyniki. Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym. Potrafi narysować (odręcznie) schemat blokowy algorytmu liniowego. Potrafi napisać prosty program, wyświetlający napis na ekranie monitora.	Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy. Zna pojęcie specyfikacji zadania. Zna wybrane sposoby prezentacji algorytmów. Przedstawia algorytm liniowy w postaci listy kroków. Podczas rysowania schematów blokowych potrafi wykorzystać Autokształty z edytora tekstu. Określa pojęcia program komputerowy, język programowania. Zapisuje prosty algorytm liniowy w wybranym języku programowania. Potrafi go skompilować i uruchomić.	Określa zależności między problemem, algorytmem a programem komputerowym. Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów, i podać przykłady. Przedstawia dokładną specyfikację dowolnego zadania. Analizuje poprawność budowy schematu blokowego. Wyjaśnia pojęcia: program wynikowy, kompilacja, translacja, interpretacja. Realizuje przykładowy algorytm liniowy w wybranym języku programowania. Wykonuje program i testuje go, podstawiając różne dane.	Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji). Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych. Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego. Analizuje działanie algorytmu dla przykładowych danych. Potrafi posłużyć się kompilatorem danego języka. Potrafi wskazać i poprawić błędy w programie.	Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji). Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia. Potrafi samodzielnie zapoznać się z kompilatorem wybranego języka programowania. Samodzielnie pisze program realizujący algorytm liniowy.

Zna klasyfikację języków programowania. Zna ogólną budowę programu i najważniejsze elementy języka programowania – słowa kluczowe, instrukcje, wyrażenia, zasady składni. Potrafi zrealizować prosty algorytm liniowy i z warunkami w języku wysokiego poziomu; potrafi skompilować i uruchomić program.	Zapisuje program w czytelnej postaci – stosuje wcięcia, komentarze. Rozumie znaczenie i działanie podstawowych instrukcji (m.in. iteracyjnych, warunkowych) wybranego języka programowania wysokiego poziomu. Wie, na czym polega programowanie strukturalne. Rozróżnia i poprawia błędy kompilacji i błędy wykonania. Potrafi zrealizować algorytmy iteracyjne w języku wysokiego poziomu.	Wymienia i omawia modele programowania. Potrafi prezentować złożone algorytmy (z podprogramami) w wybranym języku programowania. Zna i stosuje instrukcje wyboru. Prezentuje wybrane algorytmy iteracyjne w postaci programu komputerowego. Zna rekurencyjne realizacje prostych algorytmów. Rozumie i stosuje zasady programowania strukturalnego. Deklaruje procedury i funkcje bez parametrów. Wie, na czym polega różnica pomiędzy przekazywaniem parametrów przez zmienną i przez wartość w procedurach i funkcjach. Rozumie zasady postępowania przy rozwiązywaniu problemu metodą	Wie, jaka jest różnica między językiem wysokiego poziomu a językiem wewnętrznym; potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów. Deklaruje procedury i funkcje z parametrami. Wie, jakie znaczenie ma zasięg zmiennej. Definiuje funkcje rekurencyjne. Potrafi prezentować algorytmy rekurencyjne w postaci programu. Zapisuje w postaci programu wybrane algorytmy sortowania, algorytmy na tekstach, definiując odpowiednie procedury lub funkcje.	Ocenia efektywność działania programu. Wskazuje podobieństwa i różnice dotyczące tworzenia programów zapisanych w różnych językach programowania; wyjaśnia działanie poszczególnych instrukcji, sposób deklaracji zmiennych. Sprawnie definiuje i stosuje procedury i funkcje w programach. Sprawnie korzysta z dodatkowej, fachowej literatury. Rozwiązuje przykładowe zadania z matury i olimpiady informatyczne.
---	--	---	---	--

	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.	Potrafi zastosować łańcuchowy i tablicowy typ danych w zadaniach. Deklaruje typ tablicowy i łańcuchowy.	Rozumie, na czym polega dobór struktur danych do algorytmu. Wczytuje i wyprowadza elementy tablicy. Wprowadza dane tekstowe. Tworzy programy, dobierając odpowiednie struktury danych do programu.	Dobiera najlepszy algorytm i odpowiednie struktury danych do rozwiązania postawionego problemu.
--	---	---	---	--	--

Określa sytuacje warunkowe. Podaje przykłady zadań, w których występują sytuacje warunkowe. Wie, na czym polega powtarzanie tych samych operacji. Potrafi omówić na konkretnym przykładzie algorytm znajdowania najmniejszego z trzech elementów.	Potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzieniami). Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady. Wie, od czego zależy liczba powtórzeń. Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Testuje rozwiązanie dla wybranych danych. Określa problemy, w których występuje rekurencja i podaje przykłady „zjawisk rekurencyjnych” – wziętych z życia i zadań szkolnych. Potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy. Omawia wybrane algorytmy sortowania. Omawia wybrany	Analizuje algorytmy, w których występują powtórzenia (iteracje). Ocenia zgodność algorytmu ze specyfikacją. Zna sposoby zakończenia iteracji. Określa kroki iteracji. Potrafi zapisać w wybranej notacji np. algorytm sumowania n liczb, algorytm obliczania silni, znajdowania minimum w ciągu n liczb, algorytm rozwiązywania równania liniowego. Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci. Zna iteracyjną postać algorytmu Euklidesa. Zna rekurencyjną realizację wybranego algorytmu, np. silni. Zna przynajmniej dwie techniki sortowania (np. bąbelkowe, przez wybór) i zapisuje wybrany algorytm w postaci programu	Zapisuje algorytmy z pętlą zagnieżdżoną. Zna metodę „dziel i zwyciężaj”, algorytm generowania liczb Fibonacciego, schemat Hornera. Omawia ich iteracyjną realizację i potrafi przedstawić jeden z nich w wybranej notacji. Zna inne algorytmy sortowania, np. pozycyjne, przez wstawianie. Wskazuje różnicę między rekurencją a iteracją. Zna rekurencyjną realizację wybranych algorytmów, np. obliczania silni i algorytm Euklidesa. Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną. Zapisuje wybrany algorytm na tekstach (np. tworzenie anagramów, zliczanie znaków w tekście, sprawdzanie, czy dany ciąg	Rozumie dokładnie technikę rekurencji (znaczenie stosu). Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję. Zna trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, szukanie wzorca w tekście. Potrafi zapisać je w różnych notacjach (również w języku programowania wysokiego poziomu). Korzysta samodzielnie z dodatkowej literatury fachowej.
---	---	---	--	---

	Wymienia własności algorytmów. Potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego poprawność.	Zna i omawia własności algorytmów. Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją. Określa liczbę prostych działań zawartych w algorytmie.	Rozumie, co to jest złożoność czasowa algorytmu i potrafi określić liczbę operacji wykonywanych na elementach zbioru w wybranym algorytmie sortowania. Rozróżnia złożoność czasową i pamięciową.	Wie, jak ocenić złożoność pamięciową algorytmu. Potrafi porównać złożoność obliczeniową różnych algorytmów tego samego zadania dla tych samych danych. Wie, kiedy algorytm jest efektywny.	Określa złożoność czasową i pamięciową wybranych algorytmów. Zna odpowiednie wzory. Określa efektywność algorytmów.
--	---	--	--	---	--

Zna urządzenia multimedialne, wymienia przykładowe nazwy, określa ogólnie przeznaczenie urządzeń multimedialnych. Posługuje się drukarką i skanerem. Po zeskanowaniu zapisuje obraz w pliku w formacie domyślnym. Wymienia programy do tworzenia i obróbki grafiki. Posługuje się jednym z nich w celu tworzenia własnych rysunków. Tworzy i edytuje obrazy w wybranym programie graficznym, korzystając z podstawowych narzędzi do edycji obrazu. Wie, czym różni się grafika rastrowa od wektorowej.	Korzysta z różnych urządzeń multimedialnych, zna ich działanie, podaje ich przeznaczenie. Definiuje pojęcie komputer multimedialny. Zna sposoby reprezentacji obrazu i dźwięku w komputerze. Zna modele barw. Zapisuje plik graficzny w różnych formatach. Zna zastosowanie poszczególnych formatów, ich zalety i wady. Wymienia formaty zapisu dźwięku. Zna podstawowe możliwości wybranych programów do edycji obrazu rastrowego i wektorowego. Potrafi wybrać fragmenty obrazu i wykonać na nich różne operacje, np. selekcje, przekształcenia (obroty, odbicia). Potrafi, korzystając z gotowego pliku video, wykonać proste operacje, tj. podzielić film	Posługuje się sprawnie wybranymi urządzeniami multimedialnymi. Zna różne możliwości komputera w zakresie edycji obrazu, dźwięku, animacji i wideo. Omawia model barw. Zna pojęcia: RGB i CMYK. Omawia formaty plików dźwiękowych. Potrafi stosować różne narzędzia malarskie i korekcyjne oraz wybrać odpowiedni tryb ich pracy. Wie, na czym polega praca z warstwami. Wykonuje rysunki, korzystając z warstw. Potrafi tworzyć przykładowe fotomontaże. Tworzy obraz w grafice wektorowej, rysuje figury, ścieżki. Wykonuje przekształcenia obrazu (obroty, odbicia), tworząc obrazy w grafice rastrowej i wektorowej.	Zna i stosuje w praktyce zaawansowaną obróbkę grafiki rastrowej i wektorowej. 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. Stosuje filtry. Wykonuje ćwiczenia z maskami (wybiera fragmenty obrazu). Potrafi retuszować obraz. Wie, czym są krzywe Béziera i rysuje je. Opracowuje samodzielnie krótki film.	Samodzielnie zapoznaje się z programami do obróbki grafiki rastrowej i wektorowej. Tworzy obrazy, wykorzystując różne możliwości programów. Korzystając z Pomocy i innych źródeł, poznaje możliwości programów graficznych. Przygotowuje grafikę do własnej strony internetowej lub prezentacji multimedialnej.
--	---	---	--	--

	Tworzy prezentację składającą się z kilku slajdów. Wstawia teksty i obrazy, stosuje animacje. Zna ogólne zasady tworzenia prezentacji. Potrafi dobrać tło, atrybuty czcionek, odpowiednio rozmieścić tekst i grafikę na slajdzie.	Rozróżnia sposoby przygotowania prezentacji wspomagającej wystąpienie prelegenta oraz prezentacji typu kiosk. Wie, na czym polega dostosowanie treści i formy do rodzaju prezentacji. Zna i stosuje poprawne zasady tworzenia prezentacji wspomagającej wystąpienie prelegenta. Potrafi zaprojektować prezentację wspomagającą własne wystąpienie. Posługuje się szablonem projektu. Wyszukuje i gromadzi gotowe materiały (teksty i obrazy, dźwięk). Komponuje układ slajdów i ich animację. Stosuje zasady prezentowania pokazu slajdów. Zna zasady przygotowania prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się.	Zna i stosuje metody projektowania różnych rodzajów prezentacji. Potrafi zaprojektować prezentację wspomagającą własne wystąpienie. Wybiera temat, przygotowuje scenariusz, wyszukuje oraz tworzy materiały (teksty i obrazy, dźwięk). Komponuje układ slajdów i ich animacje. Posługuje się widokiem sortowania slajdów. Stosuje zasady referowania konkretnego tematu wspomagane o prezentacją. Zna i stosuje zasady przygotowania prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się. Dodaje efekty multimedialne : animacje, grafikę, dźwięki, podkład muzyczny. Ustawia i testuje chronometraż. Stosuje hiperłącza.	Na gotowym, poprawnie wykonanym przykładzie przedstawia zasady tworzenia prezentacji multimedialnych. Wyjaśnia, na czym polega dostosowanie treści i formy do rodzaju prezentacji. Wyjaśnia różnice w zasadach projektowania prezentacji wspomagającej wystąpienie prelegenta, prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się. Potrafi organizować prezentację w widoku konspektu. Wykorzystuje możliwości tworzenia schematu organizacyjnego oraz możliwości tworzenia wykresów. Nagrywa narrację. Poprawnie ustawia i testuje chronometraż, stosuje hiperłącza. Uczestniczy w przygotowaniu w formie projektów grupowych:	Przygotowuje profesjonalnie prezentacje dowolnego typu. Potrafi, korzystając z prezentacji wspomagającej wystąpienie prelegenta, przeprowadzić profesjonalny pokaz. Dodaje do prezentacji materiały ze skanera, aparatu cyfrowego i kamery cyfrowej. Publikuje prezentację w Internecie. Dopasowuje parametry konwersji do formatu HTML. Przygotowuje materiały ułatwiające opracowanie prezentacji, np. wydruk miniaturek slajdów wraz z notatkami. Nagrywa narrację i dodaje ją do prezentacji.
--	--	--	--	---	--

	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 zlecone zadania szczegółowe. Planuje temat projektu.	Wie, co to jest system informatyczny . Potrafi omówić zakres prac na każdym etapie. Realizuje projekt na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania. Uczestniczy czynnie w analizie systemu informacyjnego, przygotowuje dokumentację. Przygotowuje założenia w postaci dokumentów edytora tekstu, korzystając z szablonów. Współpracuje przy projektowaniu tabel, formularzy i raportów.	Realizuje projekt zgodnie z zamierzoną organizacją pracy zespołowej i wytyczonymi wcześniej etapami projektowania. Wykonuje trudniejsze prace związane z projektowaniem systemu. Przeprowadza analizę systemu informacyjnego. Projektuje złożone kwerendy, formularze, raporty. Uczestniczy we wdrażaniu systemu informatycznego.	Potrafi pełnić funkcję koordynatora projektu. Przydziela zadania szczegółowe, dba o ich prawidłowe wykonanie, nadzoruje pracę innych, dba o dobrą atmosferę w grupie.
--	--	---	--	--	---

	Zna i stosuje podstawowe sposoby wprowadzania danych i wyprowadzania wyników w wybranym języku programowania.	Potrafi poprawnie zadeklarować dane potrzebne do rozwiązania zadania. Wie, co określa typ danych i format danych. Potrafi odpowiednio sformatować wyprowadzane wyniki.	Tworzy własne typy danych (typ okrojony, typ wyliczeniowy) i stałe. Zna procedury i funkcje umożliwiające generowanie danych losowych. Stosuje w programach generowanie losowych danych.	Poprawnie formatuje wyprowadzane wyniki. Omawia stos jako przykład struktury danych. Wykonuje operacje na stosie. Implementuje stos z wykorzystaniem tablicy. Zna i omawia zastosowanie stosu (odwrotną notację polską – ONP).	Dobiera sposób wprowadzania danych do rozwiązywanego zadania. Potrafi sprawdzić poprawność danych wprowadzanych do programu. Definiuje własne typy danych, potrzebne do rozwiązania danego zadania. Korzystając z możliwości generowania danych losowych, tworzy własne gry komputerowe.
--	--	---	---	---	--

	Zna i omawia strukturę typów danych w wybranym języku programowania. Wie, jakiego typu dane zapisuje się w rekordzie. Wie, czym charakteryzują się dane typu tablicowego.	Tworzy prosty program, w którym deklaruje dane typu rekordowego (strukturalnego). Wczytuje dane do rekordu. Potrafi omówić ogólnie zasady przetwarzania plików w wybranym języku programowania.	Potrafi zastosować rekordowy typ danych do przetwarzania danych różnego rodzaju. Zna zasady przetwarzania plików w wybranym języku programowania (w językach programowania). Stosuje odpowiednie procedury i funkcje. Deklaruje zmienne typu plikowego. Korzystając z przykładów, odczytuje dane z pliku i zapisuje dane do pliku.	Wykonuje operacje na plikach w wybranym języku programowania (w językach programowania). Przetwarza pliki tekstowe. Tworzy własne programy, w których wykorzystuje przetwarzanie plików. Zna dynamiczne struktury danych i typ wskaźnikowy danych. Deklaruje zmienne typu wskaźnikowego. Stosuje w programach zmienne wskaźnikowe. Tworzy zmienne dynamiczne. Zna wybrane struktury dynamiczne. Analizuje gotowe programy, w których zastosowano listę jednokierunkową i binarne drzewo poszukiwań; uruchamia je i testuje dla wybranych danych.	Stosuje w programach wybrane struktury dynamiczne. Tworzy listę jednokierunkową. Tworzy i przegląda binarne drzewo poszukiwań. Pisze program wyszukujący wartości w binarnym drzewie uporządkowanym. Korzystając z dodatkowej literatury, zapoznaje się z innymi strukturami dynamicznymi, np. z listą dwukierunkową. Tworzy programy z zastosowaniem struktur dynamicznych. Rozwiązuje zadania z matury i olimpiady informatyczne i bierze w niej udział.
--	--	--	--	--	--

	Wymienia modele programowania. Omawia model programowania strukturalnego. Wie, jakie są korzyści z definiowania procedur i funkcji. Potrafi zdefiniować procedury i funkcje w wybranym języku programowania. Wie, co to jest moduł.	Omawia szczegółowo modele programowania. Zna zasady programowania modularnego. Analizując przykładowe programy, tworzy własne moduły.	Wie, na czym polega programowanie obiektowe. Zna podstawowe pojęcia programowania obiektowego: klasa, obiekt, pola, metody.	Rozumie i potrafi zastosować typ obiektowy. Definiuje klasy. Deklaruje pola prywatne. Omawia cechy programowania obiektowego: dziedziczenie i polimorfizm. Wie, czym są metody wirtualne. Poznaje zastosowanie tych cech, analizując gotowe programy. Modyfikuje programy według wskazówek nauczyciela.	Tworzy własne programy, stosując poznane zasady programowania modularnego i obiektowego. Korzysta z fachowej literatury. Wie, na czym polega programowanie zdarzeniowe. Rozwiązuje zadania z olimpiady informatycznej i bierze w niej udział.
--	---	--	---	--	---