

## Wymagania edukacyjne:

### warsztaty programowania - klasa IIA, liceum ogólnokształcące (3-letnie)

Dział	Wymagania edukacyjne (ocena celująca) Uczeń:	Wymagania edukacyjne (ocena bardzo dobra) Uczeń:	Wymagania edukacyjne (ocena dobra) Uczeń:	Wymagania edukacyjne (ocena dostateczna) Uczeń:	Wymagania edukacyjne (ocena dopuszczająca) Uczeń:
Algorytmika i programowanie - podstawy	<p>Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji).</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 kompilatorem wybranego języka programowania.</p> <p>Samodzielnie pisze program realizujący algorytm liniowy.</p>	<p>Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji).</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 posłużyć się kompilatorem danego języka.</p> <p>Potrafi wskazać i poprawić błędy w programie.</p>	<p>Określa zależności między problemem, algorytmem a programem komputerowym.</p> <p>Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów, i podać przykłady.</p> <p>Przedstawia dokładną specyfikację dowolnego zadania.</p> <p>Analizuje poprawność budowy schematu blokowego.</p> <p>Wyjaśnia pojęcia: program wynikowy, kompilacja, translacja, interpretacja.</p> <p>Realizuje przykładowy algorytm liniowy w wybranym języku programowania.</p> <p>Wykonuje program i testuje go, podstawiając różne dane.</p>	<p>Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy.</p> <p>Zna pojęcie specyfikacji zadania.</p> <p>Zna wybrane sposoby prezentacji algorytmów.</p> <p>Przedstawia algorytm liniowy w postaci listy kroków.</p> <p>Podczas rysowania schematów blokowych potrafi wykorzystać Autokształty z edytora tekstu.</p> <p>Określa pojęcia: program komputerowy, język programowania.</p> <p>Zapisuje prosty algorytm liniowy w wybranym języku programowania. Potrafi go skompilować i uruchomić.</p>	<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 umieszczania w schemacie blokowym.</p> <p>Potrafi narysować (odręcznie) schemat blokowy algorytmu liniowego.</p> <p>Potrafi napisać prosty program, wyświetlający napis na ekranie monitora.</p>
Algorytmy warunkowe i iteracyjne	<p>Sprawnie posługuje się poleceniami warunkowymi.</p> <p>Stosuje iterację bezwarunkową do rozwiązywania problemów algorytmicznych np. warunku istnienia trójkąta.</p> <p>Korzysta samodzielnie z dodatkowej literatury fachowej.</p>	<p>Zapisuje algorytmy z pętlą zagnieżdżoną.</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>Analizuje algorytmy, w których występują powtórzenia (iteracje).</p> <p>Ocenia zgodność algorytmu ze specyfikacją.</p> <p>Zna sposoby zakończenia iteracji. 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>Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci.</p> <p>Zna iteracyjną postać algorytmu Euklidesa.</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>Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Testuje rozwiązanie dla wybranych danych.</p>	<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 konkretnym przykładzie algorytm znajdowania najmniejszego z trzech elementów.</p>

<p style="text-align: center;"><b>Procedury i funkcje w językach programowania - zastosowanie</b></p>	<p>Sprawnie definiuje i stosuje procedury i funkcje w programach.</p> <p>Samodzielnie dokonuje wyboru oraz stosuje odpowiednią strukturę algorytmiczną podczas rozwiązywania zadań.</p> <p>Sprawnie korzysta z dodatkowej, fachowej literatury.</p> <p>Rozwiązuje przykładowe zadania z matury.</p>	<p>Deklaruje procedury i funkcje z parametrami.</p> <p>Wie, jakie znaczenie ma zasięg zmiennej.</p> <p>Definiuje funkcje rekurencyjne. Potrafi prezentować algorytmy rekurencyjne w postaci programu.</p> <p>Zapisuje w postaci programu wybrane algorytmy sortowania, algorytmy na tekstach, definiując odpowiednie procedury lub funkcje.</p>	<p>Wymienia i omawia modele programowania.</p> <p>Potrafi prezentować złożone algorytmy (z podprogramami) w języku programowania C++.</p> <p>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 i funkcjach.</p>	<p>Potrafi poprawnie zadeklarować dane potrzebne do rozwiązania zadania.</p> <p>Wie, co określa typ danych i format danych.</p> <p>Potrafi odpowiednio sformatować wyprowadzane wyniki.</p> <p>Omawia etapy programowania w języku C++.</p> <p>Zna kryteria, jakie powinien spełniać poprawny program.</p> <p>Deklaruje procedury i funkcje bez parametrów.</p>	<p>Zna ogólną budowę programu i najważniejsze elementy języka programowania – słowa kluczowe, instrukcje, wyrażenia, zasady składni.</p> <p>Zna zastosowanie procedury i funkcji.</p> <p>Zna podstawowe typy zmiennych.</p> <p>Potrafi zdefiniować prostą procedurę i funkcję.</p>
<p style="text-align: center;"><b>Stosowanie tablic i łańcuchów w programowaniu</b></p>	<p>Prawidłowo dobiera struktury danych do algorytmu oraz obsługuje wartości w nich przechowywane.</p>	<p>Sprawnie posługuje się zmiennymi tablicowymi oraz łańcuchami w algorytmach.</p> <p>Potrafi dobrać strukturę danych do algorytmu.</p> <p>Prawidłowo wykonuje działania na elementach tablic i łańcuchów.</p>	<p>Sprawnie wykonuje działania na zmiennych tablicowych jednowymiarowych.</p> <p>Deklaruje tablice dwuwymiarowe. Wprowadza i wyprowadza wartości tablic dwuwymiarowych, wykonuje podstawowe działania na ich elementach.</p>	<p>Deklaruje i prawidłowo używa w algorytmach tablice jednowymiarowe.</p> <p>Wprowadza i wyprowadza elementy tablicy jednowymiarowej.</p> <p>Używa w działaniach elementów tablicy jednoelementowej.</p>	<p>Wie jak zadeklarować zmienną tablicową.</p> <p>Potrafi zinterpretować algorytm zawierający zmienne tablicowe.</p>

Iteracja warunkowa, realizacja wybranych algorytmów	<p>Zna i sprawnie posługuje się poleceniami iteracji warunkowej.</p> <p>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).</p> <p>Korzysta samodzielnie z dodatkowej literatury fachowej.</p>	<p>Zna inne algorytmy sortowania, np. pozycyjne, przez wstawianie.</p> <p>Zapisuje wybrany algorytm na tekstach (np. tworzenie anagramów, zliczanie znaków w tekście, sprawdzanie, czy dany ciąg jest palindromem) w postaci programu komputerowego.</p> <p>Zapisuje algorytm wydawania reszty metodą zachłanną w postaci programu komputerowego.</p>	<p>Analizuje algorytmy, w których występują powtórzenia (iteracje).</p> <p>Ocena zgodność algorytmu ze specyfikacją.</p> <p>Zna sposoby zakończenia iteracji. 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>Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci.</p> <p>Zna iteracyjną postać algorytmu Euklidesa.</p> <p>Zna rekurencyjną realizację wybranego algorytmu, np. silni.</p> <p>Zna przynajmniej dwie techniki sortowania (np. bąbelkowe, przez wybór) i zapisuje wybrany algorytm w postaci programu komputerowego.</p> <p>Omawia wybrane algorytmy na tekstach.</p> <p>Potrafi wyjaśnić, na czym polega wydawanie reszty metodą zachłanną i napisać listę kroków tego algorytmu.</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>Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Testuje rozwiązanie dla wybranych danych.</p> <p>Określa problemy, w których występuje rekurencja i podaje przykłady „zjawisk rekurencyjnych” – wziętych z życia i zadań szkolnych.</p> <p>Potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy.</p> <p>Omawia wybrane algorytmy sortowania.</p> <p>Omawia wybrany algorytm na tekstach (np. tworzenie anagramów).</p>	<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 konkretnym przykładzie algorytm znajdowania najmniejszego z trzech elementów.</p>
Rekurencja w algorytmach	<p>Rozumie dokładnie technikę rekurencji (znaczenie stosu).</p> <p>Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję.</p> <p>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).</p>	<p>Wskazuje różnicę między rekurencją a iteracją.</p> <p>Zna rekurencyjną realizację wybranych algorytmów, np. obliczania silni i algorytm Euklidesa.</p> <p>Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną.</p>	<p>Rozumie działanie struktury rekurencyjnej, dokładnie wyjaśnia jej działanie.</p> <p>Definiuje proste algorytmy rekurencyjne.</p>	<p>Analizuje proste algorytmy rekurencyjne.</p> <p>Potrafi wyjaśnić zasadę działania rekurencji na wybranych przykładach.</p>	<p>Wie na czym polega struktura rekurencji w algorytmach.</p> <p>Rozpoznaje proste algorytmy rekurencyjne.</p>

