

Wymagania edukacyjne z informatyki klasa 2 – zakres rozszerzony

Szczegółowe wymagania edukacyjne dla klasy 2A

Temat	Ocena dopuszczająca Uczeń:	Ocena dostateczna Uczeń:	Ocena dobra Uczeń:	Ocena bardzo dobra Uczeń:	Ocena celująca Uczeń:
Programowanie i rozwiązywanie problemów za pomocą komputera					
Przypomnij sobie, czyli wracamy do środowiska programistycznego	– definiuje środowisko programistyczne i jego najważniejsze elementy – na podstawie podręcznika definiuje proces kompilacji kodu programu	– korzystając z podręcznika lub zasobów internetowych, instaluje środowisko programistyczne, np. Eclipse wraz z pakietem MinGW	– samodzielnie instaluje środowisko programistyczne, np. Eclipse wraz z pakietem MinGW – korzystając z podręcznika lub zasobów internetowych, testuje poprawność działania środowiska programistycznego	– samodzielnie wykonuje czynności opisane dla oceny dobrej – omawia działanie debuggera i proces kompilacji z uwzględnieniem modułów związanych ze środowiskiem systemu operacyjnego	– płynnie posługuje się środowiskiem programistycznym i konfiguruje je samodzielnie
Przypomnij sobie, czyli podstawy języka programowania	– definiuje kod źródłowy – korzystając z podręcznika lub zasobów internetowych, nazywa i definiuje operatory w języku C++	– omawia znaczenie zmiennych w programie – korzystając z podręcznika lub zasobów internetowych, omawia działanie instrukcji warunkowej – korzystając z podręcznika lub zasobów internetowych, omawia działanie pętli – omawia znaczenie funkcji w programie komputerowym	– samodzielnie spełnia wymagania oceny dostatecznej – korzystając z podręcznika lub zasobów internetowych, podaje przykłady wykorzystania zmiennych, funkcji, instrukcji warunkowych i pętli	– samodzielnie spełnia wymagania oceny dobrej – wyjaśnia różnice pomiędzy typami danych w C++ oraz podaje przykłady ich stosowania	– wykorzystuje informacje z lekcji w prostych programach C++
Wskaźniki	– zna idee budowy zmiennych	– prawidłowo deklaruje zmienne wskaźnikowe;	– poprawnie definiuje wskaźniki różnych	– programuje z użyciem zmiennych	– samodzielnie planuje użycie zmiennych

	wskaźnikowych; – z pomocą nauczyciel a definiuje zmienne wskaźnikowe;	– ustawia wskaźniki na adresach zmiennych;	typów; – poprawnie interpretuje kody źródłowe z zastosowaniem wskaźników;	wskaźnikowych; – rozumie problem przekazywania wartości pomiędzy wskaźnikami i niebezpieczeństwo wskazywania na ten sam adres pamięci;	wskaźnikowych do rozwiązywania problemów informatycznych; – samodzielnie układa programy, stosując zmienne wskaźnikowe;
Wskaźniki i tablice	- zna zależność pomiędzy nazwą tablicy a wskaźnikiem; - wymienia poznane funkcje działające na tekstach;	- z pomocą nauczyciela interpretuje zapisy wskaźników odwołujące się to tablic; - zna różnice pomiędzy wskaźnikiem a nazwą tablicy; - określa działanie poszczególnych funkcji modułu <string.h>	- samodzielnie interpretuje zapisy wskaźników odwołujące się to tablic; - tworzy zapisy z użyciem wskaźników do tablic;	- sprawnie posługuje się wskaźnikami do tablic; - tworzy zmienne łańcuchowe i wykonuje na nich podstawowe operacje; - posługuje się funkcjami działającymi na tekstach;	- używa w kodach źródłowych wskaźników do tablic; - sprawnie posługuje się łańcuchami znaków;
Struktury – grupujemy dane w jedną całość Struktury - zmienne strukturalne	– określa funkcję jaką odgrywają struktury w programach komputerowych; – odróżnia od siebie typ strukturalny i zmienną typu strukturalnego; – z pomocą podręcznika analizuje przykłady typów strukturalnych;	– definiuje typ strukturalny i pojedynczą strukturę; – podaje przykłady typów strukturalnych; – definiuje zmienne strukturalne; – podaje przykłady zmiennych strukturalnych;	– omawia strukturę zmiennych strukturalnych; – deklaruje struktury składające się z różnych danych; – definiuje zmienne strukturalne; – podaje przykłady odwołania się do pola;	– deklaruje struktury składające się z różnych danych; – omawia i stosuje strukturę tablicową zmiennych strukturalnych; – uzasadnia stosowanie przez zmienną (* i &) w przypadku zmiany wartości zmiennej strukturalnej wewnątrz funkcji;	– samodzielnie planuje użycie struktur złożonych do łączenia w grupy danych, które mają być ze sobą powiązane, i wybiera dane na podstawie analizy algorytmu;
Operacje wejścia i wyjścia – strumienie i pliki Operacje wejścia i wyjścia – wykonujemy działania na plikach Operacje wejścia i wyjścia – wykonujemy działania na plikach tekstowych	– omawia poszczególne etapy działań na plikach; – wie, jaka jest różnica pomiędzy plikami tekstowymi a binarnymi w C++; – zna sens sekwencyjnego dostępu do elementów w	– deklaruje zmienne plikowe dla plików różnych typów; – wymienia i charakteryzuje etapy działań z plikami; - zna podstawowe funkcje do pracy z	– otwiera i tworzy pliki; – używa funkcji fprintf i fscanf, fwrite i fread, – programuje pobranie, modyfikacje i zapis danych do pliku; – prawidłowo zamyka pliki;	– posługuje się funkcjami: fputs, fputc, fgets i fgetc, fseek, ftell i foef; – świadomie stosuje dwa sposoby dostępu do pliku – sekwencyjny i swobodny; – prawidłowo dobiera i wykorzystuje funkcje	– samodzielnie planuje wykorzystanie odpowiednich funkcji działających na plikach do realizacji programów na podstawie analizy algorytmów;

Operacje wejścia i wyjścia – wykonujemy działania na plikach	pliku;	plikami;		działające na plikach;	
Wykrywamy błędy (debugowanie)	<ul style="list-style-type: none"> – zna istotę debugowania i jego znaczenie w pracy programisty; – podaje przykłady, w których wskazane jest użycie debugera i uruchamiania krokowego programu; – wie, czym jest pułapka; 	<ul style="list-style-type: none"> – omawia różnicę pomiędzy opcjami debugowania (F7 i F8); – debuguje prosty program i wskazuje miejsca wyświetlania danych debugowania, np. wartości zmiennych; – świadomie stosuje debugowanie do wykrywania błędów w krótkim programie bez funkcji; 	<ul style="list-style-type: none"> – wykorzystuje debugowanie do analizy błędów w programie z wywołaniem funkcji z niewielką ilością zmiennych; – umie kontrolować wartości zmiennych w trakcie debugowania; – uruchamia program metoda krokową, kontrolując zawartości zmiennych; – umie uruchomić debugowanie od dowolnego miejsca programu; – debuguje lub pomija w tym procesie podprogramy; 	<ul style="list-style-type: none"> – uruchamia program metoda krokową, kontrolując zawartości zmiennych; – umie uruchomić debugowanie od dowolnego miejsca programu; – debuguje lub pomija w tym procesie podprogramy; – odnajduje błędy za pomocą debugera lub pracy krokowej, porównując wartości zmiennych z przewidywanymi przez algorytm; – umie kontrolować wartości zmiennych w trakcie debugowania; 	<ul style="list-style-type: none"> – biegle korzysta z debugera, analizując tylko wybrane fragmenty programu. – typuje miejsca w programie, w których należy szukać błędów, i kontroluje je debugerem;
W świecie wskaźników	<ul style="list-style-type: none"> – wie, jakie są mechanizmy odwoływania się do zmiennych dynamicznych za pomocą wskaźników; – podaje przykłady takich zastosowań typów wskaźnikowych; 	<ul style="list-style-type: none"> – wskazuje różnice pomiędzy zmiennymi wskaźnikowymi statycznymi a dynamicznymi; 	<ul style="list-style-type: none"> – posługuje się operatorami new i delete; – wie, jak działają powyższe operatory; – analizuje gotowe przykłady deklaracji zmiennych wskaźnikowych; – deklaruje zmienne wskaźnikowe; – umie odwołać się do zmiennej dynamicznej za 	<ul style="list-style-type: none"> – umie analizować błędy powstałe podczas tworzenia programu ze zmiennymi dynamicznymi; – umie analizować błędy powstałe podczas tworzenia programu ze zmiennymi dynamicznymi; – wie, czym jest spowodowana utrata 	<ul style="list-style-type: none"> – samodzielnie układa programy z zastosowaniem zmiennych dynamicznych; – podaje uzasadnienie użycia zmiennych dynamicznych we własnym rozwiązaniu problemu informatycznego;
Deklarujemy zmienne wskaźnikowe					
Tworzymy pierwsze zmienne dynamiczne					
Co jeszcze powinieneś wiedzieć o wskaźnikach?					

			<p>– pomocą zmiennej wskaźnikowej;</p>	<p>– dostępu do zmiennej dynamicznej; – zna znaczenie stałej nil; – określa zastosowania zmiennych wskaźnikowych takie jak odwołanie do zmiennych dynamicznych, użycie w instrukcji przypisania, użycie w relacji, jako wynik funkcji;</p>	
<p>Dynamiczne struktury danych</p>	<p>– umie podać przykład zastosowania dynamicznej struktury danych;</p>	<p>– zna znaczenie rekordu w tworzeniu struktur dynamicznych jako zmiennej dynamicznej; – wie, czym jest zmienna wskaźnikowa i umie opisać jej znaczenie w dynamicznej strukturze danych; – wie, czym jest pole wskaźnikowe w rekordzie;</p>	<p>– definiuje i deklaruje w programie struktury dynamiczne; – łączy zmienne dynamiczne na podstawie przykładów z podręcznika; – usuwa zmienną dynamiczną; – zmienia wartość zmiennej dynamicznej;</p>	<p>– samodzielnie łączy zmienne dynamiczne; – używa zmiennej wskaźnikowej do wypełniania pola rekordu dynamicznego; – wypełnia pola rekordu dynamicznego;</p>	<p>– samodzielnie układa programy z zastosowaniem struktur dynamicznych; – podaje uzasadnienie użycia struktur dynamicznych we własnym rozwiązaniu problemu informatycznego;</p>
<p>Dynamiczne struktury danych – zapis kodu</p>					
<p>Stos – ostatni wchodzi, pierwszy wychodzi LIFO</p>	<p>– umie opisać istotę stosu; – umie podać różnice pomiędzy LIFO a FIFO;</p>	<p>– zna budowę stosu i jego zastosowanie; – opisuje rolę wskaźników i zmiennych wskaźnikowych w tworzeniu i obsłudze kolejek;</p>	<p>– wie, jak zbudować stos dla zmiennych dynamicznych; – wie, jak przeglądać stos i jak usuwać zmienne dynamiczne; – wie, jak tworzyć w swoich programach kolejkę zmiennych; – wie, na czym polega wstawianie elementów do kolejki, jej przeglądanie i usuwanie elementów;</p>	<p>– samodzielnie buduje stos dla zmiennych dynamicznych; – zna funkcję zmiennej wskaźnikowej w adresowaniu wierzchołka stosu; – samodzielnie tworzy program przeglądający stos; – samodzielnie tworzy program usuwający zmienne dynamiczne;</p>	<p>– wie, jak zachowuje się komputer, gdy zostaje wywołane przerwanie, i wie, jaka w tym rola stosu procesora; – stosuje w swoich programach stos programowy; – identyfikuje w algorytmach i problemach informatycznych zagadnienia, w których</p>
<p>Kolejka – pierwszy wchodzi, pierwszy wychodzi (FIFO)</p>					

				– tworzy w swoich programach kolejkę zmiennych;	należy zastosować FIFO lub LIFO;
Lista jednokierunkowa	– wskazuje różnicę pomiędzy listą jedno- i dwukierunkową; – zna dokładnie różnice pomiędzy kolejką a stosem;	– wymienia cechy listy; – analizuje graficzne, podręcznikowe przedstawienie działań na listach; – wymienia niektóre problemy informatyczne, w których stosuje się listy;	– umie porządkować listy jednokierunkowe; – umie definiować w programie listy dwukierunkowe; – na podstawie podręcznika analizuje przykład programu z listą cykliczną; – definiuje strukturę dla listy dwukierunkowej; – wie, czym jest i w jakich przypadkach znajduje zastosowanie lista cykliczna i podaje zasadniczą cechę listy cyklicznej; – samodzielnie analizuje kod programu, w którym zastosowano listy;	– wstawia i usuwa elementy z listy jedno- i dwukierunkowej; – z powodzeniem stosuje w programach listy w tym także cykliczne;	– identyfikuje w algorytmach zagadnienia, które mogą być zrealizowane z wykorzystaniem list;
Lista dwukierunkowa, a nawet cykliczna					
Drzewo	– opisuje analogię struktury drzewiastej do struktury folderów w systemie operacyjnym;	– wymienia cechy drzewa; – analizuje graficzne, podręcznikowe przedstawienie struktury drzewiastej;	– wymienia niektóre problemy informatyczne, w których stosuje się drzewa; – interpretuje kod źródłowy zawierający implementacje struktur drzewiastych;	– modyfikuje kody źródłowe zawierający implementacje struktur drzewiastych;	– potrafi zastosować struktury drzewiaste do swoich potrzeb;
Czy litery to liczby, czyli kod ASCII i porównanie tekstów	– definiuje kod ASCII	– omawia znaczenie kodu ASCII – definiuje plagiat i odnosi tę definicję także do rzeczywistości	– wykorzystuje klawiaturę numeryczną do wprowadzania znaków za pomocą kodów ASCII	– interpretuje przepisy dotyczące plagiatów – wie, czym jest JSA i jakie ma znaczenie w zwalczaniu kopiowania	Nie przewiduje się oceny celującej w tym temacie

		szkolnej, np. do kopiowania prac domowych		prac naukowych	
Metoda naiwna, czyli szukamy wzorca w tekście	– na podstawie podręcznika opisuje istotę metody naiwnej	– na podstawie ilustracji z podręcznika omawia działanie algorytmu wyszukiwania wzorca w tekście	– samodzielnie omawia działanie algorytmu i metodę naiwną wyszukiwania wzorca w tekście – wizualizuje metodę naiwną na przykładzie krótkiego tekstu – na podstawie podręcznika testuje działanie algorytmu	– samodzielnie testuje działanie algorytmu	– samodzielnie tworzy lub modyfikuje algorytm wyszukiwania wzorca metodą naiwną
Realizacja algorytmu, czyli program szuka wzorca	– rozumie działanie algorytmu wyszukiwania wzorca metodą naiwną	– na podstawie podręcznika interpretuje i omawia działanie programu wyszukiwania wzorca	– samodzielnie omawia działanie programu z podręcznika i testuje go dla różnych danych	– omawia znaczenie pętli w programie z podręcznika	– samodzielnie układa program różny od programu z podręcznika
Przybliżona wartość, czyli komputer oblicza wartość pierwiastka kwadratowego	– rozumie, na czym polega proces szacowania wartości pierwiastka – wie, do czego służy funkcja pow	– na podstawie podręcznika lub z pomocą nauczyciela szacuje wartość pierwiastka na osi liczbowej. – biegle używa strumieni cin i cout oraz prawidłowo stosuje funkcję pow	– samodzielnie układa prosty program obliczający wartość pierwiastka, używając funkcji pow – program jest funkcjonalny i pozwala na wprowadzanie danych i wyświetlanie wyniku, – na podstawie podręcznika omawia znaczenie liczby kolejnych przybliżeń na wynik – rozumie działanie algorytmu obliczającego wartość pierwiastka	– samodzielnie układa algorytm obliczania pierwiastka kwadratowego – wykazuje na przykładach wpływ liczby przybliżeń na dokładność wyniku działania programu	– samodzielnie układa program obliczający wartość pierwiastka trzeciego stopnia bez użycia funkcji pow

			kwadratowego i na podstawie tabeli z podręcznika testuje jego działanie		
Zbiór Cantora, czyli najprostsze tworzenie fraktali	– wie, czym jest fraktal i podaje przykłady, np. z natury, zachowujące się jak fraktale, np. liście	– przedstawia w formie graficznej istotę zbioru Cantora	– samodzielnie układa algorytm tworzący zbiór Cantora – testuje algorytm dla różnych stopni zbioru Cantora – za pomocą podręcznika testuje i rozumie działanie programu kreślącego graficzne przedstawienie zbioru Cantora	– samodzielnie układa i modyfikuje program z wykorzystaniem biblioteki turtle ilustrujący zbiór Cantora	– samodzielnie układa program ilustrujący zbiór Cantora w języku C++
Drzewo binarne, czyli hierarchiczna struktura danych	– na podstawie podręcznika kreśli przykładową strukturę drzewa binarnego	– na podstawie przykładów z podręcznika omawia zastosowanie binarnego drzewa poszukiwań – zna i omawia istotę regularnego drzewa binarnego	– tworzy BST (binarne drzewo poszukiwań) i omawia mechanizm szukania danego elementu, np. liczby – tłumaczy działanie programu wstawiającego element do kopca	– samodzielnie układa funkcję wstawiającą element do kopca – samodzielnie omawia metodę szukania wartości w BST – podejmuje próbę ułożenia algorytmu i programu wyszukującego element metodą drzewa poszukiwań	– samodzielnie układa algorytm i program wyszukujący element metodą drzewa poszukiwań
Trójkątny fraktal, czyli trójkąt Sierpińskiego	– wie, jak powstaje trójkąt i dywan Sierpińskiego i umie to zilustrować, np. na tablicy	– na podstawie przykładów z podręcznika omawia działanie algorytmu tworzącego fraktale Sierpińskiego	– tłumaczy istotę trójkąta i dywanu Sierpińskiego – omawia znaczenie liczby kroków i możliwość nieograniczonej ich liczby	– samodzielnie układa program z użyciem biblioteki turtle kreślący trójkąt lub dywan Sierpińskiego	– samodzielnie układa program kreślący fraktale Sierpińskiego z możliwością wybrania liczby kolejnych podziałów
Krzywa fraktalna, czyli płatek Kocha	– na przykładzie z podręcznika tłumaczy, jak powstaje krzywa Kocha	– tłumaczy, na czym polega efekt samopodobności w	– tłumaczy, na czym polega cecha płatka Kocha mówiąca tym, że	– samodzielnie układa program kreślący płatek Kocha, np. z użyciem	– samodzielnie modyfikuje program kreślący płatek Kocha,

		<p>płatku Kocha – omawia algorytm powstawania płatka Kocha</p>	<p>brzeg płatka ma nieskończoną długość, a pole wartość skończoną – na podstawie podręcznika omawia działanie programu używającego biblioteki turtle kreślącego płatek Kocha</p>	<p>biblioteki turtle, i testuje jego działanie</p>	<p>np. dodaje możliwość wprowadzania liczby poziomów</p>
Wykorzystanie algorytmów w rozwiązywaniu problemów i programowaniu					
<p>Znajdowanie elementów w zbiorze, czyli wyszukiwanie liniowe</p>	<p>– omawia istotę metody naiwnej w zastosowaniu do wyszukiwania elementów w zbiorze</p>	<p>– na podstawie podręcznika omawia istotę metody wyszukiwania liniowego</p>	<p>– omawia algorytm w postaci pseudokodu wyszukiwania liniowego – tworzy algorytm dla metody naiwnej</p>	<p>– samodzielnie tworzy algorytm w postaci pseudokodu wyszukiwania liniowego</p>	<p>– układa program dla metody liniowej</p>
<p>Szukanie połówek, czyli wyszukiwanie elementów poprzez połowienie</p>	<p>– na podstawie podręcznika omawia istotę metody wyszukiwania przez połowienie i podaje przykłady</p>	<p>– omawia istotę sortowania i jej znaczenie w metodzie szukania elementów metodą binarną (połowienie)</p>	<p>– omawia algorytm w postaci pseudokodu wyszukiwania binarnego</p>	<p>– samodzielnie tworzy algorytm w postaci pseudokodu wyszukiwania binarnego</p>	<p>– układa program dla metody wyszukiwania binarnego</p>
<p>Ustawianie kart, czyli sortowanie przez wstawianie</p>	<p>– na podstawie podręcznika lub innych wiarygodnych źródeł omawia istotę sortowania „przez wstawianie”</p>	<p>– na podstawie podręcznika lub innych wiarygodnych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu „przez wstawianie” – analizuje przykład sprawdzający poprawność działania algorytmu</p>	<p>– samodzielnie na przykładzie omawia istotę metody sortowania „przez wstawianie” – omawia działanie przykładowego algorytmu opartego na metodzie sortowania „przez wstawianie” – sprawdza działanie algorytmu na przykładach</p>	<p>– samodzielnie układa algorytm sortowania „przez wstawianie” w postaci listy kroków i schematu blokowego – samodzielnie weryfikuje poprawność działania programu na przykładach</p>	<p>– samodzielnie układa program sortujący metoda „przez wstawianie” w innym języku niż C++, np. Java</p>
<p>Reguła fałsi, czyli przybliżone rozwiązywanie równań</p>	<p>– omawia zagadnienie określoności i ciągłości funkcji</p>	<p>– na podstawie podręcznika omawia istotę metody fałsi w kontekście szukania</p>	<p>– samodzielnie na przykładzie omawia istotę metody fałsi do obliczania miejsca</p>	<p>– samodzielnie układa algorytm znajdowania miejsca zerowego funkcji metoda fałsi</p>	<p>– układa program znajdowania miejsca zerowego funkcji metoda fałsi</p>

		miejsca zerowego funkcji – omawia znaczenie twierdzenie Talesa w obliczaniu miejsca zerowego funkcji	zerowego funkcji – omawia działanie przykładowego algorytmu opartego ona metodzie falsi – sprawdza działanie algorytmu na przykładach		
Geometria obliczeniowa, czyli sprawdzanie przynależności punktu do wielokąta wypukłego	– omawia różnice między figurami wypukłymi a wklęsłymi	– omawia na przykładzie pojęcie przynależności punktu do figury wypukłej i wklęsłej	– na podstawie podręcznika wyjaśnia działanie algorytmu badania przynależności punktu do wielokąta wypukłego i wklęsłego	– samodzielnie układa i omawia algorytm badania przynależności punktu do wielokąta wypukłego i wklęsłego	– układa program badający przynależność punktu do wielokąta wypukłego i wklęsłego
Algorytm wyszukiwujący, czyli liniowe przeszukiwanie ciągu w poszukiwaniu żadanego elementu z wykorzystaniem wartownika	– omawia zagadnienie i specyfikę algorytmu naiwnego – na podstawie podręcznika omawia pojęcie przestrzeni poszukiwań	– na podstawie podręcznika omawia istotę wyszukiwania liniowego z wartownikiem	– na podstawie podręcznika wyjaśnia działanie algorytmu wyszukiwania liniowego z wartownikiem	– samodzielnie układa i omawia algorytm wyszukiwania liniowego z wartownikiem	– układa program wyszukiwania liniowego z wartownikiem
Podejście zachłanne, czyli problem plecakowy	– wyjaśnia istotę metody zachłannej – na podstawie podręcznika wyjaśnia, na czym polega problem plecakowy	– samodzielnie omawia zastosowanie metody zachłannej w kontekście problemu plecakowego	– na podstawie podręcznika wyjaśnia działanie algorytmu rozwiązującego problem plecakowy	– samodzielnie układa algorytm rozwiązujący problem plecakowy metodą zachłanną	– układa program rozwiązujący problem plecakowy metodą zachłanną
Z miasta A do miasta B, czyli szukanie najkrótszej drogi metodą zachłanną	– omawia istotę metody zachłannej i podejścia iteracyjnego w rozwiązywaniu problemów – wyjaśnia, na czym polega problem znalezienia najkrótszej drogi	– na podstawie podręcznika wyjaśnia podstawy i założenia algorytmu Dijkstry	– samodzielnie na przykładzie wyjaśnia działanie algorytmu Dijkstry	– samodzielnie układa algorytm Dijkstry i wyjaśnia jego działanie	– układa program rozwiązujący problem poszukiwania najkrótszej drogi pomiędzy punktami
Pakowanie plecaka, czyli programowanie dynamiczne	– przypomina założenia problemu pakowania	– wyjaśnia działanie metody zstępującej i	– samodzielnie, na przykładzie wyjaśnia	– samodzielnie układa algorytm rozwiązujący	– układa program rozwiązujący problem

	<p>plecaka</p> <ul style="list-style-type: none"> – na podstawie podręcznika wyjaśnia założenia programowania dynamicznego – opisuje zastosowanie zmiennych tablicowych 	<p>wstępującej</p> <ul style="list-style-type: none"> – objaśnia zastosowanie metodą programowania dynamicznego w rozwiązywaniu problemu plecakowego 	<p>działanie algorytmu rozwiązującego problem plecakowy z zastosowaniem programowania dynamicznego</p>	<p>problem plecakowy z zastosowaniem programowania dynamicznego</p>	<p>plecakowy z zastosowaniem programowania dynamicznego</p>
<p>Metoda haszowania, czyli wyszukiwanie wzorca w tekście</p>	<ul style="list-style-type: none"> – wyjaśnia i definiuje pojęcie haszowania – omawia znaczenie wyszukiwania wzorca w tekście 	<ul style="list-style-type: none"> – na podstawie podręcznika opisuje metodę haszowania w kontekście wyszukiwania wzorca w tekście – omawia znaczenie zmiennych tablicowych w programowaniu metodą haszowania 	<ul style="list-style-type: none"> – omawia istotę tablicy haszującej na podstawie funkcji przedstawionej w podręczniku – na podstawie podręcznika omawia zastosowanie metody haszowania do wyszukiwania wzorca w tekście 	<ul style="list-style-type: none"> – samodzielnie omawia istotę tablicy haszującej i przedstawia przykładową implementację programową – omawia zastosowanie metody haszowania do wyszukiwania wzorca w tekście 	<ul style="list-style-type: none"> – układa program rozwiązujący problem wyszukiwania wzorca w tekście z zastosowaniem metody haszowania
<p>Metoda Monte Carlo, czyli symulacja ruchów Browna</p>	<ul style="list-style-type: none"> – wie, czym jest modelowanie matematyczne – omawia istotę metody Monte Carlo 	<ul style="list-style-type: none"> – na podstawie podręcznika omawia przykład stosowania metody Monte Carlo w wyznaczaniu wartości liczby pi – na podstawie podręcznika omawia zjawisko ruchów Browna 	<ul style="list-style-type: none"> – omawia algorytm i program przedstawiony w podręczniku wyznaczający wartość liczby pi metodą Monte Carlo – omawia przedstawiony w podręczniku problem symulacji ruchów Browna 	<ul style="list-style-type: none"> – samodzielnie omawia metodę Monte Carlo w kontekście zastosowania do wyznaczania wartości liczby pi – samodzielnie omawia zastosowanie metody Monte Carlo do symulacji ruchów Browna 	<ul style="list-style-type: none"> – układa program wizualizujący symulacje ruchów Browna
<p>Nie wszystko jest takie oczywiste, czyli jak działa internet</p>	<ul style="list-style-type: none"> – wie, jaką rolę odgrywają protokoły w sieciach komputerowych – opisuje rolę adresów w sieciach lokalnych i internecie 	<ul style="list-style-type: none"> – na podstawie podręcznika umie wyświetlić parametry połączenia sieciowego za pomocą polecenia <code>tracert</code> uruchomionego w Wierszu poleceń – na podstawie podręcznika uruchamia i stosuje program do 	<ul style="list-style-type: none"> – wyjaśnia różnice pomiędzy adresem IP a adresem symbolicznym – na podstawie opisu z podręcznika sprawnie posługuje się programami do śledzenia połączeń w sieci – tłumaczy rolę DNS-ów w globalnej sieci 	<ul style="list-style-type: none"> – samodzielnie korzysta z programów do śledzenia połączeń i znajdowania właściciela domen – tłumaczy zadania protokołu DHCP – interpretuje wyniki działania programów śledzących połączenia oraz polecenia systemowego 	<ul style="list-style-type: none"> – biegle posługuje się różnymi programami do diagnozowania i testowania działania sieci komputerowych

		śledzenia połączeń z serwerem wybranej strony WWW	– tłumaczy rolę adresów IP – wie, jakie instytucje są odpowiedzialne za przydzielanie adresów IP w sieci globalnej – wie, czym jest domena	tracert	
Zabezpieczenia firewall	- wie czym jest firewall	- wie jak działa zaporę sieciową - omawia typy zapór sieciowych	- omawia serwery NAT - zna wady zapór sieciowych	- wyjaśnia, na czym polega tworzenie przez zaporę sieciową strefy ograniczonego zaufania nazywanej również strefą zdemilitaryzowaną DMZ.	- samodzielnie konfiguruje zapory sieciowe
Funkcje urządzeń i protokoły	- wie co to są urządzenia sieciowe	- omawia poszczególne urządzenia sieciowe - zna protokoły modeli warstwowych sieci	- omawia zastosowanie poszczególnych urządzeń sieciowych	- omawia protokoły warstw aplikacji - omawia protokoły warstwy transportowej - omawia protokoły warstwy sieciowej - omawia protokoły wirtualnych sieci prywatnych: PPP, PPTP i L2TP	- omawia trasę po której pakiety są przesyłane do konkretnego serwera www - omawia za co odpowiadają protokoły ICMP, ARP
Kto tam, czyli identyfikujemy komputery w sieci	– na podstawie podręcznika konfiguruje sieć w komputerze z systemem Windows, korzystając z automatycznych opcji konfiguracyjnych	– na podstawie podręcznika umie sprawdzić poleceniem ping poprawność połączenia sieciowego	– na podstawie podręcznika omawia informacje wyświetlone za pomocą poleceń ipconfig oraz ping – na podstawie podręcznika omawia rolę poszczególnych protokołów sieciowych	– samodzielnie posługuje się poleceniami ipconfig oraz ping w celu sprawdzenia parametrów połączeń sieciowych – samodzielnie opisuje poszczególne dane odczytane za pomocą ipconfig oraz ping – samodzielnie korzysta z automatycznych narzędzi konfiguracji sieci – samodzielnie omawia	– samodzielnie konfiguruje połączenie sieciowe z pominięciem nastaw automatycznych

				rolę poszczególnych protokołów sieciowych	
Kupujemy świadomie, czyli poznajemy parametry urządzeń peryferyjnych	<ul style="list-style-type: none"> – umie oszacować koszty wydruku dla danego typu lub modelu drukarki – rozpoznaje i nazywa wejścia sygnałowe w monitorach – wie, do czego służy skaner 	<ul style="list-style-type: none"> – wymienia parametry drukarek – na podstawie podręcznika określa wpływ poszczególnych parametrów drukarek na jakość druku – na podstawie podręcznika omawia parametry monitorów – na podstawie podręcznika omawia parametry skanerów 	<ul style="list-style-type: none"> – na podstawie podręcznika omawia cechy i parametry poszczególnych typów drukarek i ich wpływ na wybór dokonywany ze względu na zastosowanie – określa parametry monitorów oraz wpływ formatu obrazu na zastosowanie na różnych stanowiskach 	<ul style="list-style-type: none"> – samodzielnie analizuje parametry urządzeń peryferyjnych i ocenia ich przydatność do konkretnego zastosowania – samodzielnie wyjaśnia zalety i wady różnych rodzajów ekranów monitorów 	<ul style="list-style-type: none"> – potrafi na podstawie danych katalogowych trafnie dobrać urządzenie peryferyjne, biorąc pod uwagę wymagania użytkownika
Projektuję rozbudowę i zakup nowego zestawu	<ul style="list-style-type: none"> - wie czym jest komputer stacjonarny - wie czym jest komputer przenośny 	<ul style="list-style-type: none"> - zna różnice między komputerem stacjonarnym i przenośnym 	<ul style="list-style-type: none"> - potrafi przeprowadzić analizę potrzeb przed zakupem nowego komputera 	<ul style="list-style-type: none"> - samodzielnie wyjaśnia wady i zalety komputera przenośnego i komputera stacjonarnego, omawiając jego poszczególne komponenty 	<ul style="list-style-type: none"> - potrafi na podstawie danych dostępnych w sieci www wybrać urządzenia, które polepszą funkcjonowanie komputera przenośnego lub stacjonarnego
Host i serwer, czyli o działaniu sieci	<ul style="list-style-type: none"> – na podstawie podręcznika i z pomocą nauczyciela omawia podstawowe pojęcia związane z budową sieci komputerowych (np. <i>host</i>, <i>serwer</i>, <i>klient</i>, <i>LAN</i>, <i>WAN</i>, <i>WiFi</i>); –wie, czym są media transmisyjne; – wymienia nazwy najczęściej spotykanych topologii sieci lokalnej i na 	<ul style="list-style-type: none"> – samodzielnie omawia znaczenie podstawowych pojęć związanych z budową i działaniem sieci komputerowych; – na podstawie podręcznika omawia cechy poszczególnych mediów transmisyjnych; – samodzielnie wymienia nazwy najczęściej spotykanych topologii sieci lokalnej i wymienia 	<ul style="list-style-type: none"> – samodzielnie omawia znaczenie wszystkich pojęć związanych z budową i działaniem sieci komputerowych; – oblicza maksymalną ilość danych przesyłanych w sieci na podstawie danych o czasie i przepustowości łącza; – z niewielką pomocą nauczyciela wymienia 	<ul style="list-style-type: none"> – wybiera odpowiednią topologię i medium dla różnych sieci lokalnych; – identyfikuje topologię istniejącej sieci (np. w pracowni); – identyfikuje rodzaje mediów transmisyjnych; – prawidłowo stosuje poznane pojęcia dotyczące sieci lokalnych. 	<ul style="list-style-type: none"> – omawia niewymienione w podstawie programowej zagadnienia związane z tematem.

	podstawie ilustracji (np. z podręcznika) wymienia ich podstawowe cechy.	ich podstawowe cechy; – wymienia jednostki przesyłu informacji i omawia zależności pomiędzy nimi; – wie, że do różnych zastosowań można używać różnych rodzajów kabla sieciowego.	nazwy i cechy podstawowych kabli dla sieci lokalnych CAT 3, 5 i 5e, 6 i 7; – wymienia podstawowe dla transmisji danych cechy światłowodu.		
Zespół, czyli realizujemy projekty w chmurze	– omawia budowę przykładowej chmury, wymieniając jej podstawowe składniki.	– przedstawia i omawia przykładowy schemat struktury chmury informatycznej; – wymienia najczęściej spotykane w chmurach narzędzia (m.in. dysk, edytor, arkusz, kalendarz); – wie, że chmurę można wykorzystać do pracy zespołowej.	– omawia przeznaczenie poszczególnych elementów chmury użytej na zajęciach; – porównuje działanie programów z chmury z ich odpowiednikami z aplikacji komputerowych; – posługuje się chmurą w stopniu wystarczającym do wykonywania podstawowych czynności, takich jak edycja dokumentu, kopiowanie plików itp.; – wie, które aplikacje można wykorzystać do organizacji pracy zespołu.	– biegle posługuje się programami i dyskami sieciowymi; – samodzielnie przygotowuje chmurę do pracy zespołu, m.in. konfiguruje kalendarz, udostępnia foldery i pliki.	– samodzielnie planuje i organizuje pracę zespołu w chmurze; – posługuje się kilkoma chmurami (np. Google i OneDrive).
Tutorial, czyli jak tworzyć pomoce i instrukcje obsługi	– wie czym są style i szablony; – zmienia styl w trakcie edycji dokumentu.	– prawidłowo dobiera style do treści zawartej w dokumencie; – wie, że można samodzielnie definiować	– dobiera styl i szablon zgodnie z planowaną zawartością i tematem dokumentu; – na podstawie podręcz-	– samodzielnie definiuje szablon i styl dokumentu; – projektuje szablon zgodnie z treścią przyszłego dokumentu;	– projektuje szablony dla różnych edytorów (np. LibreOffice Draw) i ich używa.

		style i szablony; – uruchamia edycję, wykorzystując szablon oferowany przez edytor.	nika definiuje szablon i styl dokumentu; – odnajduje w sieci (np. w chmurze) szablony dla różnych dokumentów.	– modyfikuje i projektuje nowe style.	
Rozbudowana struktura, czyli korzystamy z konspektu w edytorze tekstu	– rozumie i omawia pojęcie konspektu; – podaje, czym jest akapit i jaką pełni funkcję w edycji i formatowaniu tekstu.	– na podstawie podręcz- nika tworzy konspekt dokumentu; – na podstawie podręcz- nika dzieli dokument na sekcje i kolumny.	– samodzielnie tworzy konspekt dokumentu; – samodzielnie dzieli dokument na sekcje i kolumny; – tworzy spis treści na podstawie konspektu.	– uzasadnia stosowanie podziału dokumentu tekstowego na sekcje lub kolumny; – stosuje różny podział na niektórych stronach (sekcjach) dokumentu.	– stosuje podziały i sekcje w różnych edytorach tekstu.
Broszura, czyli jak projektować duże dokumenty	– wstawia do dokumentów SmartArt i Kształty.	– uzasadnia wybór danego kształtu lub ilustracji SmartArt; – na podstawie podręcz- nika tworzy spisy ilustracji i tabel.	– na podstawie podręcz- nika lub tutoriali zmienia domyślne opcje edytora; – na podstawie podręcz- nika lub tutoriali wyko- nuje automatyczne spisy treści, tabel i ilustracji; – na podstawie podręcz- nika lub tutoriali aktuali- zuje spisy po zmianach.	– samodzielnie zmienia domyślne opcje edytora; – samodzielnie wykonuje automatyczne spisy treści, tabel i ilustracji; – samodzielnie aktualizuje spisy po zmianach; – decyduje, które z opcji domyślnych zmienić i uzasadnia swój wybór.	– spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice).
Recenzja, czyli proponujemy poprawki w tekście	– odczytuje i interpretuje dokumenty, w których zastosowano tryb recenzji; – odczytuje notatki i zaznaczenia w Adobe Acrobat Reader DC.	– na podstawie podręcz- nika używa opcji Recenzja w Word i wstawiania komentarzy w dokumencie PDF.	– na podstawie podręcz- nika lub tutoriali używa opcji Recenzja do wpisywania komentarzy w Word oraz opcji wstawiania komentarzy i zaznaczenia fragmentów w dokumencie PDF; – na podstawie podręcz- nika lub tutoriali uruchamia opcję śledzenia zmian.	– samodzielnie używa opcji Recenzja do wpisywania komentarzy w Word oraz opcji wstawiania komentarzy i zaznaczenia fragmentów w dokumencie PDF; – samodzielnie uruchamia opcję śledzenia zmian; – świadomie używa opcji Recenzja do proponowa- nia zmian w dokumencie; – analizuje wynik działa- nia opcji porównywania dokumentów.	– spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice); – przedstawia dokument z innego przedmiotu lub projektu, w którym użył opcji Recenzja dla dokumentu Word lub Notatka dla PDF.

<p>OLE, czyli łączymy dokumenty w dokumentacji pracy zespołu</p>	<p>– na podstawie podręcznika wyjaśnia, czym jest i do czego służy technika OLE.</p>	<p>– wyjaśnia, jakie obiekty mogą być wstawiane za pomocą techniki OLE; – na podstawie podręcznika lub opcji Pomoc umieszcza w dokumencie obiekty OLE; – na podstawie podręcznika wyjaśnia, czym są OLE2 i ActiveX; – wyjaśnia, dlaczego podczas używania ActiveX należy zachować szczególną ostrożność.</p>	<p>– na podstawie podręcznika i opcji Pomocy przetwarza obiekty OLE w edytorach tekstu itp.</p>	<p>– samodzielnie przetwarza obiekty OLE w edytorach tekstu itp.</p>	<p>– spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice).</p>
--	--	---	---	--	--