

**Przedmiot:** Informatyka rozszerzona  
**Rok szkolny:** 2016/2017  
**Klasa:** Klasa III technikum – zajęcia 2 godz. x 30 tyg. = 60 godz.

**Prowadzący:** mgr inż. Beata Maciewicz  
mgr Beata Piekuelko  
mgr Henryk Kuczmierczyk  
mgr Edmund Żółtowski

Proponowany podział materiału nauczania

<b>Lp.</b>	<b>Dział</b>	<b>Podział godzin dla zakresu rozszerzonego (180 godzin lekcyjnych)</b>
1.	Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki. Tendencje w rozwoju informatyki i jej zastosowań	4
2.	Bazy danych	50
3.	Multimedia	16
4.	Systemy operacyjne i sieci komputerowe	20
5.	Wprowadzenie do algorytmiki	15
6.	Algorytmy i ich zastosowanie	32
7.	Programowanie w języku C++	34
8.	Projekt programistyczny	9

Lp.	Temat lekcji	Liczba godzin do realizacji	Przewidywane osiągnięcia ucznia	Odwolanie do podręcznika	Odwolanie do podstawy programowej
<b>KLASA II</b>					
1,2	Zapoznanie z regulaminem pracowni, rozkładem materiału, PSO, przepisami BHP. (2 godziny dydaktyczne)				
<b>1. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach inf</b>					
3, 4,	Kierunki rozwoju informatyki i jej zastosowań.	2	Potrafi wymienić wiele zastosowań komputerów w różnych dziedzinach życia. Określa korzyści wynikające z rozwoju i wszechstronnego wykorzystania informatyki w różnych dziedzinach życia. Orientuje się w nowych technologiach i śledzi zmiany zachodzące w rozwoju informatyki. Dostrzega zalety e-learningu, korzysta z kursów zdalnego nauczania.	5.1 5.1.1 5.1.2	1.4 6.2 7.1 7.2 7.5
5, 6,	Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki.	2	Zna przepisy prawne dotyczące wykorzystywania, przetwarzania i publikowania informacji — prawo autorskie i licencje. Przestrzega zasad etyki w zakresie korzystania z utworów innych osób.	5.2 5.2.1 5.2.2 5.2.3 5.2.4	7.3 7.4
7,	Sprawdzian wiadomości	1		2.1 - 2.4.2	
<b>2. Bazy danych</b>					
Lp.	Temat lekcji	Liczba godzin do realizacji	Przewidywane osiągnięcia ucznia	Odwolanie do podręcznika	Odwolanie do podstawy programowej
8, 9, 10,	Podstawowe zasady tworzenia tabeli stanowiącej bazę danych.	3	Zna budowę bazy danych i pojęcia z nią związane. Zna i stosuje reguły, jakim podlega tabela stanowiąca źródło danych. Potrafi zgromadzić w pliku tekstowym dane będące bazą danych. Importuje dane umieszczone w pliku tekstowym do tabeli arkusza kalkulacyjnego. Zapisuje zgromadzone w tabeli dane w pliku tekstowym, stosując odpowiednie znaki separacji.	1.1 1.2 1.3	2.1 2.3 4.4
11, 12,	Wyszukiwanie informacji w tabeli przy użyciu autofiltru oraz filtru zaawansowanego.	2	Korzysta z autofiltru w celu wyselekcjonowania danych. Potrafi stosować złożone kryteria wyboru w filtrach zaawansowanych. Stosuje odpowiedni rodzaj filtru w celu wybrania potrzebnych w danej sytuacji informacji.	1.4.1 1.4.2 1.4.3 1.4.4	4.4
13	Tworzenie podsumowań danych — sumy pośrednie.	1	Potrafi zastosować funkcje standardowe arkusza do podsumowań danych. Zna pojęcie suma pośrednia. Wie, do czego używa się sum pośrednich, potrafi je stosować. Przy użyciu kreatora sum pośrednich potrafi dokonać podsumowania danych zawartych w tabeli. Potrafi dokonać korekty i zamiany kryteriów podsumowań w istniejącym zestawieniu sum pośrednich.	1.4.5	4.4
14, 15, 16,	Tworzenie podsumowań danych — tabele przestawne.	3	Rozumie pojęcie tabela przestawna. Wie, do czego służą tabele przestawne. Wskazuje, jakie dane można umieszczać w tabelach przestawnych. Potrafi wykonać zestawienie podsumowań danych przy użyciu kreatora tabel przestawnych. Potrafi dokonać zmiany w opcjach projektu istniejącej tabeli przestawnej. Przygotowuje statystyki oparte na raporcie tabeli przestawnej.	1.4.6 1.4.7	2.1 4.4
17, 18,	Graficzna prezentacja danych na wykresach.	2	Potrafi dobrać odpowiedni typ wykresu do prezentowanych na nim danych. Umie stworzyć różnego typu wykresy do danych zawartych w tabelach zwykłych i przestawnych. Potrafi ustalić odpowiednie opcje wykresu i formatować jego poszczególne elementy.	1.4.5 1.4.6	4.4

			Potrafi zaprezentować graficznie rozwiązania zadań i problemów z innych dziedzin nauczania i problemów z życia codziennego (równanie i układ równań, wyniki pomiarów fizycznych lub chemicznych, analiza rynku pracy).		
19, 20, 21, 22,	Rozwiązywanie przykładowych zadań maturalnych z wykorzystaniem arkusza kalkulacyjnego.	4	Świadomie wybiera właściwy sposób rozwiązania zadania. Korzysta z istniejącego oprogramowania w celu rozwiązania problemu. Wykorzystuje funkcje arkusza kalkulacyjnego i poznane metody wyszukiwania informacji do rozwiązywania problemu. Wykorzystuje zdobytą wiedzę i umiejętności do rozwiązywania prostych i umiarkowanie złożonych zadań z różnych dziedzin.	1.4.7	2.1 4.4
23, 24,	Podstawy relacyjnej bazy danych.	2	Zna i rozumie pojęcia: system zarządzania danymi, relacyjna baza danych. Zna zasady projektowania bazy danych. Zna właściwości rekordów i pól bazy danych i rozumie różnice pomiędzy nimi. Na przykładzie istniejącej relacyjnej bazy danych wymienia obiekty związane z takimi bazami (tabele, kwerendy, formularze, raporty, makrodefinicje).	2.1 2.2 2.2.1 2.2.2	2.1
25, 26, 27, 28, 29, 30,	Tabele w relacyjnej bazie danych: Projektowanie i tworzenie tabeli, Typy danych w tabelach, Podstawowe właściwości pól tabeli, Pole kluczowe tabeli, Definiowanie relacji między tabelami.	6	Potrafi zaprojektować układ tabel, unikając powtarzania danych. Potrafi zaimportować tabele z istniejącej bazy danych, arkusza kalkulacyjnego czy pliku tekstowego. Tworzy tabele przy użyciu kreatora tabel oraz w widoku projektu. Potrafi ustalić właściwości pól. Zna różne formaty danych. Zna pojęcia: klucz główny oraz klucz obcy. Dostrzega korzyści wynikające ze stosowania kluczy głównych. Łączy tabele odpowiednimi relacjami. Rozróżnia podstawowe typy relacji między tabelami, potrafi je nazwać. Projektuje tabele, uwzględniając różne typy danych. Projektuje relacyjne bazy danych z uwzględnieniem zjawisk redundancji. Zapewnia integralność danych.	2.3 2.3.1 2.3.2 2.3.3 2.3.4 2.3.5	2.1
31, 32,	Operacje na tabelach bazy danych.	2	Wykonuje podstawowe czynności edycyjne na obiektach bazy danych (kopiowanie, usuwanie, zmiana nazwy). Potrafi wstawiać obiekty OLE. Odróżnia osadzanie obiektu w tabeli od jego połączenia.	2.3.6	2.1 2.2
33, 34,	Zmiana sposobu prezentowania danych.	2	Potrafi zmieniać wygląd tabeli w widoku arkusza danych. Zmienia kolejność sortowania danych w tabeli. Potrafi porządkować tabele i przeglądać wybrane rekordy przez sortowanie jedno- i wieloparametrowe. Stosuje filtry do wyszukiwania informacji.	2.3.7	2.1 2.2
35, 36,	Import, eksport, załączanie tabeli.	2	Korzysta z danych przechowywanych w innych bazach programu Access oraz utworzonych w innych aplikacjach.	2.3.8	2.1
37, 38, 39, 40,	Wyszukiwanie informacji w relacyjnej bazie danych z użyciem kwerend wybierających.	4	Samodzielnie projektuje proste zapytania, korzystając z widoku projektu i kreatorów różnego typu kwerend. Stosuje odpowiednie kryteria i parametry do wyszukiwania danych. Definiuje złożone kryteria wyboru danych.	2.4 2.4.1	2.1 2.2 2.3
41, 42,	Definiowanie wyrażeń w kwerendach wybierających.	2	Konstruuje pola obliczeniowe, wprowadza korekty. Stosuje kwerendy parametryczne.	2.4.1	2.1 2.2 2.3
43, 44,	Wyszukiwanie informacji w relacyjnej bazie danych z użyciem kwerend funkcjonalnych.	2	Tworzy kwerendy funkcjonalne generujące tabele, aktualizujące dane, usuwające i dołączające dane. Tworzy zestawienia krzyżowe, odpowiednio definiując kryteria, nagłówki wierszy i kolumn.	2.4.2	2.1 2.2 2.3
45, 46,	Język zapytań SQL.	2	Wyszukuje informacje w bazach danych, stosując różne techniki (w tym konstruowanie rozbudowanych zapytań).	2.4.3	2.2
47, 48,	Wprowadzanie, wyszukiwanie, edycja i usuwanie danych bezpośrednio w tabelach i poprzez formularze.	2	Wie, jak wprowadzić dane bezpośrednio do tabeli lub poprzez formularz. Wie, jak korzystać z formularzy w celu wprowadzania, wyszukiwania i edycji danych. Potrafi drukować tabele i formularze. Tworzy i modyfikuje formularze oraz formanty za pomocą	2.5 2.5.1 2.5.2	2.1 2.2 2.3

			kreatora i w widoku projektu. Dodaje etykiety, formanty, nagłówki lub stopkę do formularza, zmienia jego szatę graficzną.		
49, 50,	Przygotowanie zestawień wybranych danych w raportach.	2	Tworzy proste raporty przy użyciu narzędzia Autoraport. Wie, jak korzystać z kreatora raportów do generowania dowolnych rodzajów raportów. Potrafi grupować informacje w raporcie. Wie, kiedy korzystać z podsumowania statystycznego. Przygotowuje raport do druku.	2.6	2.1 2.3
51,	Makropolecenia.	1	Definiuje makropolecenia składające się z sekwencji kilku akcji.	2.7	2.3
52, 53,	Ochrona bazy danych.	2	Ma świadomość tego, że do baz danych zawierających określone informacje, np. dane osobowe, dostęp powinni mieć tylko uprawnieni użytkownicy. Zna i potrafi stosować podstawowe mechanizmy chroniące plik bazy danych przed jego otwarciem i ewentualną przypadkową lub celową modyfikacją. Rozumie konieczność wykonywania kopii bezpieczeństwa (na przykład podczas wprowadzania i testowania kwerend funkcjonalnych). Zna sposób i rozumie celowość kodowania bazy danych.	2.8	2.5
54, 55, 56,	Rozwiązywanie przykładowych zadań maturalnych z wykorzystaniem relacyjnej bazy danych.	3	Tworzy rozwiązania w wybranym środowisku użytkowym. Świadomie wybiera właściwy sposób rozwiązania zadania. Testuje rozwiązania. Wykorzystuje zdobytą wiedzę i umiejętności do rozwiązywania prostych i umiarkowanie złożonych zadań z różnych dziedzin. Odwołując się do zbioru danych, analizuje problem, którego rozwiązanie wymaga zaprojektowania i utworzenia relacyjnej bazy danych. Projektuje strukturę bazy danych (tabele i relacje między nimi) z uwzględnieniem specyfiki informacji zawartych w bazie. Tworzy zaprojektowaną bazę danych. Projektuje strukturę bazy danych (tabele i relacje między nimi) z uwzględnieniem specyfiki informacji zawartych w bazie. Tworzy zaprojektowaną bazę danych.	2.9	2.1 2.2 2.3
57, 58, 59, 60,	Podsumowanie wiadomości.	4			

### KLASA III

1,2	Zapoznanie z regulaminem pracowni, rozkładem materiału, PSO, przepisami BHP. (2 godziny dydaktyczne)				
<b>3. Systemy operacyjne i sieci komputerowe</b>					
Lp.	Temat lekcji	Liczba godzin do realizacji	Przewidywane osiągnięcia ucznia	Odwołanie do podręcznika	Odwołanie do podstawy programowej
3	Rola, funkcje i zasady pracy sprzętu komputerowego.	1	Opisuje budowę logiczną współczesnego komputera. Używając poprawnej polszczyzny i terminologii informatycznej opisuje funkcjonowanie komputera i rolę jego części składowych. Rozumie rolę, jaką pełni BIOS. Zna proces uruchamiania komputera.	3.1 3.2	1.2 1.4
4	Budowa i rola systemu operacyjnego.	1	Zna budowę systemu operacyjnego. Rozumie i opisuje rolę systemu operacyjnego w komputerze. Zna najpopularniejsze systemy plików.	3.3 3.3.1 3.3.2	1.2
5, 6,	Maszyna wirtualna z systemem operacyjnym DR-DOS i Linux.	2	Potrafi zainstalować oprogramowanie maszyny wirtualnej. Rozróżnia podstawowe systemy operacyjne. Wykonuje podstawowe polecenia na plikach i katalogach w różnych systemach operacyjnych.	3.3.3	1.2 1.4
7	Rodzaje sieci komputerowych.	1	Potrafi podzielić sieci ze względu na zasięg i topologię. Przedstawia budowę i funkcjonowanie komputerowej sieci lokalnej i globalnej. Zna specyfikacje sieci komputerowych i podstawowe media sieciowe. Zna urządzenia sieciowe i sposoby transmisji.	3.4 3.4.1	1.3
8	Model OSI.	1	Zna funkcje poszczególnych warstw modelu OSI.	3.4.2	1.3

			Wyjaśnia, na czym polega przepływ informacji pomiędzy warstwami.		
9, 10,	Zestaw protokołów TCP/IP.	2	Rozumie, na czym polega podział na warstwy zestawu protokołów TCP/IP, i porównuje go z modelem OSI. Potrafi określić, czy dany komputer może się komunikować z innymi komputerami w sieci. Omawia sposób przesyłania danych między protokołami TCP/IP. Zna i stosuje pojęcie pakiet danych. Szczegółowo analizuje warstwę sieciową, używając pojęć: datagram IP, segment, pakiet, ramka, protokół IP, klasy adresów IP, adresacja IP, adresy zastrzeżone, rodzaje adresowania, maska, podsieci.	3.4.3 3.4.4	1.3
11, 12,	Projektowanie sieci komputerowej z wykorzystaniem adresacji bezklasowej.	2	Potrafi zaprojektować sieć z uwzględnieniem zasad bezpieczeństwa (separacji podsieci). Potrafi przydzielić odpowiednie pule adresów IP do wymaganych segmentów podsieci. Zna zasady administrowania siecią komputerową. Odczytuje i analizuje informacje o ustawieniach sieciowych danego komputera i jego lokalizacji w sieci.	3.4.4	1.3 1.4
13	Protokoły sieciowe.	1	Potrafi wyjaśnić, czym są protokół i zestaw protokołów sieciowych. Potrafi wymienić nazwy podstawowych protokołów sieciowych i opisać ich własności.	3.4.5	1.3
14	Obsługa i konfiguracja sieci w systemie Windows.	1	Potrafi skonfigurować komputer z systemem operacyjnym Windows, aby uzyskać dostęp do internetu.	3.4.6	1.2
15	Ogólne zasady administrowania siecią komputerową w architekturze klient-serwer.	1	Potrafi wymienić serwery działające w architekturze klient-serwer. Zna zadania stawiane przed administratorem sieci.	3.4.7	1.3
16	Bezpieczeństwo informacji w sieciach.	1	Rozumie konieczność stosowania hasła w celu uwierzytelniania użytkownika oraz ochrony danych przed dostępem niepowołanych osób. Tworzy złożone hasła, chroni je i często zmienia. Zna zagrożenia związane z pracą komputera w sieci: niszczące programy, najczęstsze ataki, sniffer, podszywanie i naśladownictwo. Potrafi je scharakteryzować (opisać) oraz podjąć działania prewencyjne. Zna różne sposoby zabezpieczeń przed zagrożeniami, takie jak: szyfrowanie, certyfikaty cyfrowe, zabezpieczenia systemów operacyjnych, zaporę sieciową, filtrowanie pakietów. Rozumie konieczność stosowania programów antywirusowych i systematycznej aktualizacji bazy wirusów. Zna podstawy kodowania danych.	3.4.8	1.4 7.3 7.4
17, 18,	Tworzenie i publikowanie własnych materiałów w sieci.	2	Tworzy dokumenty dostępne w sieci. Zna strukturę oraz podstawowe znaczniki dokumentu hipertekstowego. Potrafi stworzyć i zmodyfikować witrynę WWW z wykorzystaniem tekstu, tabel, dźwięku i odpowiednich formatów grafiki i animacji. Zamieszcza własną witrynę WWW w sieci i administruje nią. Dostrzega korzyści związane ze stosowaniem arkusza stylów.	3.5.1 3.5.2 3.5.3 3.5.4 3.5.5 3.5.6	1.4 3.1
19, 20,	Czym jest PHP?	2	Zna podstawy języka PHP i wykorzystuje go na stronach HTML. Stosuje formularze i tworzy proste aplikacje bazodanowe.	3.5.8	1.4 3.1
21, 22,	Instalowanie systemów zarządzania treścią Drupal i Gallery 2 oraz administrowanie nimi.	2	Potrafi administrować systemem zarządzania treścią (instaluje i konfiguruje moduły, nadaje użytkownikom uprawnienia do zasobów, dba o bezpieczeństwo zasobów). Planuje współpracę i zespołowo wykonuje projekt witryny internetowej. Publikuje i udostępnia własne materiały w sieci.	3.5.7	1.4 2.4 3.1 3.2 6.1
<b>4. Multimedia i grafika komputerowa</b>					
Lp.	Temat lekcji	Liczba godzin do realizacji	Przewidywane osiągnięcia ucznia	Odwwołanie do podręcznika	Odwwołanie do podstawy programowej
23, 24,	Grafika rastrowa — własności obrazu.	2	Rozróżnia pojęcia: rozdzielczość, wymiary, rozmiar i wielkość obrazu. Wykonuje zaawansowane czynności edycyjne z wykorzystaniem wielu warstw obrazu. Stosuje transformacje.	4.1.1 4.1.2 4.1.5 4.1.6	1.1 4.2

			Zna wady grafiki rastrowej. Potrafi przechwytywać obraz ze skanera i aparatu fotograficznego.		
25, 26,	Reprezentacja obrazu w komputerze.	2	Rozumie sposób zapisu barwy i jej własności. Zna modele barw. Zna sposoby reprezentowania obrazów bitmapowych w komputerze. Rozróżnia formaty plików bitmapowych i zna ich przeznaczenie.	4.1.3 4.1.4 4.1.7 4.1.8	1.1 4.1 4.2
27, 28,	Algorytmy kompresji stratnej i bezstratnej.	2	Zna pojęcia: kompresja stratna i kompresja bezstratna. Rozumie różnice między kompresją stratną i bezstratną. Podaje przykłady. Potrafi wymienić i omówić algorytmy kompresji bezstratnej. Realizuje przykładowe algorytmy kompresji bezstratnej, w tym metodę kodowania długości ciągów, metodę słownikową oraz opartą na strategii zachłannej — metodę kodowania znaków. Rozróżnia kodowanie prefiksowe od nieprefiksowego.	4.1.6	4.2 5.11.e
29, 30,	Grafika wektorowa.	2	Rozumie sposób zapisu obrazu wektorowego. Wykonuje podstawowe operacje edycyjne w edytorze grafiki wektorowej. Rozróżnia sposoby i formy reprezentowania informacji w postaci obrazu, stosując kryterium użyteczności i przeznaczenia. Wymienia wady i zalety grafiki wektorowej i bitmapowej.	4.1.9 4.1.10	1.1 4.2
31, 32, 33, 34,	Grafika trójwymiarowa — 3D.	4	Zna techniki tworzenia grafiki 3D. Rozumie geometrię trójwymiarową. Wie, z czego składa się trójwymiarowa scena. Zna sposoby reprezentowania w komputerze obrazów bitmapowych 3D. Potrafi umieścić na scenie podstawowe obiekty, określić ich kolor, wielkość i położenie.	4.1.11	4.2
35, 36,	Dźwięk.	2	Zna sposoby reprezentowania dźwięku w komputerze. Zna sposób konwersji sygnału analogowego na cyfrowy. Wykorzystuje różne techniki pozyskiwania, selekcji, przetwarzania i interpretacji oraz przechowywania dźwięku.	4.2 4.2.1 4.2.2 4.2.3	1.1 4.3
37, 38,	Wideo.	2	Wie, jak powstaje ruchomy obraz. Rozumie pojęcie kontener multimedialny. Wie, jak działają kodeki. Potrafi zmontować film i go opublikować.	4.3 4.3.1 4.3.2	1.1 4.3
<b>5. Wprowadzenie do algorytmiki</b>					
39	Pojęcie algorytmu. Etapy rozwiązywania zadań za pomocą komputera	1	Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; stosuje podejście algorytmiczne do rozwiązywania problemu; formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowa nie rozwiązania.	1.1, 1.2	5.1, 5.2, 5.3, 5.7
40	Sposoby reprezentowania algorytmów: lista kroków, schemat blokowy, drzewo algorytmu.	1	Uczeń dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji.	1.3, 1.3.1, 1.3.2, 1.3.3	5.4
41	Algorytmy liniowe. Program w języku programowania wysokiego poziomu.	1	Uczeń dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji i posługuje się podstawowymi technikami algorytmicznymi.	1.3.4, 1.4.1	5.4, 5.5
42	Algorytmy z warunkami na przykładzie rozwiązywania równania kwadratowego.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy numeryczne, np.: obliczanie wartości pierwiastka kwadratowego.	1.4.2, 1.4.3	5.11.c.1
43	Pojęcie iteracji z określoną i nieokreśloną liczbą powtórzeń.	1	Uczeń posługuje się podstawowymi technikami algorytmicznymi.	1.5	5.5
44	Iteracja – ćwiczenia.	1	Uczeń posługuje się podstawowymi technikami algorytmicznymi.	1.5	5.5
45	Iteracja – ćwiczenia zaawansowane.	1	Uczeń posługuje się podstawowymi technikami algorytmicznymi.	1.5	5.5
46	Pojęcie rekurencji na przykładzie obliczania silni liczby naturalnej.	1	Uczeń posługuje się podstawowymi technikami algorytmicznymi i stosuje rekurencję w prostych sytuacjach problemowych.	1.6, 1.6.1	5.5, 5.9
47	Rekurencja: wyznaczenie wyrazów ciągu Fibonacciego. Ćwiczenia.	1	Uczeń stosuje rekurencję w prostych sytuacjach problemowych. Opisuje podstawowe algorytmy i stosuje: sprawdzanie, czy liczba jest liczbą pierwszą, doskonałą, iteracyjne i rekurencyjne obliczanie wartości liczb Fibonacciego.	1.6.2	5.9, 5.11.a.2, 5.11.a.5

48	Rekurencja na przykładzie algorytmu wieży Hanoi.	1	Uczeń stosuje rekurencję w prostych sytuacjach problemowych.	1.6.3	5.9
49	Metoda „dziel i zwyciężaj”.	1	Uczeń posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów.	1.7, 1.7.1	5.8
50	Programowanie zachłanne. Minimalizacja łączenia par.	1	Uczeń stosuje podejście zachłanne w rozwiązywaniu problemów. Opisuje podstawowe algorytmy i stosuje algorytmy na liczbach całkowitych, np.: wydawanie reszty metodą zachłanną.	1.8, 1.8.1	5.10, 5.11.a.6
51	Kryptografia i kryptoanaliza. Metody szyfrowania	1	Opisuje podstawowe algorytmy i stosuje algorytmy kompresji i szyfrowania, np.: kody znaków o zmiennej długości, np. alfabet Morse’a, kod Huffmana, szyfr Cezara, szyfr przestawieniowy, szyfr z kluczem jawnym (RSA), wykorzystanie algorytmów szyfrowania, np. w podpisie elektronicznym.	1.9	5.11.e
52	Własności algorytmów: złożoność obliczeniowa i efektywność, poprawność i skończoność, optymalność algorytmów	1	Uczeń ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania. Projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów.	1.10, 1.10.1, 1.10.2, 1.10.3	5.6, 5.12 – 5.20
53	Sprawdzian wiadomości.	1		1.1-1.1.10.3	
54, 55, 56, 57, 58, 59, 60,	Rozwiązywanie zadań maturalnych	7			

## KLASA IV

1, 2,	Zapoznanie z regulaminem pracowni, rozkładem materiału, PSO, przepisami BHP. (2 godziny dydaktyczne)
-------	--

### 6. Algorytmy i ich zastosowanie

Lp.	Temat lekcji	Liczba godzin do realizacji	Przewidywane osiągnięcia ucznia	Odwołanie do podręcznika	Odwołanie do podstawy programowej
3	Algorytmy badające własności geometryczne.	1	Uczeń opisuje podstawowe algorytmy i stosuje: algorytmy badające własności geometryczne, np.: sprawdzanie warunku trójkąta, badanie położenia punktów względem prostej, badanie przynależności punktu do odcinka, przecinanie się odcinków, przynależność punktu do obszaru.	2.1	5.11.f
4	Wyznaczanie największego wspólnego dzielnika za pomocą algorytmu Euklidesa	1	Uczeń opisuje podstawowe algorytmy i stosuje: rozkładanie liczby na czynniki pierwsze, iteracyjna i rekurencyjna realizacja algorytmu Euklidesa.	2.2.1	5.11.a.3, 5.11.a.4
5	Obliczanie najmniejszej wspólnej wielokrotności. Ćwiczenia	1	Uczeń opisuje podstawowe algorytmy i stosuje: rozkładanie liczby na czynniki pierwsze.	2.2.2	5.11.a.3
6	Systemy liczbowe. Systemy liczbowe. Konwersje pozycyjnych systemów liczbowych.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy na liczbach całkowitych, np.: reprezentacja liczb w dowolnym systemie pozycyjnym, w tym w dwójkowym i szesnastkowym.	2.3.1, 2.3.2	5.11.a.1
7	Operacje arytmetyczne wykonywane na różnych systemach liczbowych.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy na liczbach całkowitych, np.: reprezentacja liczb w dowolnym systemie pozycyjnym, w tym w dwójkowym i szesnastkowym.	2.3.3	5.11.a.1
8	Konwersja i operacje arytmetyczne wykonywane na różnych systemach liczbowych – ćwiczenia.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy na liczbach całkowitych, np.: reprezentacja liczb w dowolnym systemie pozycyjnym, w tym w dwójkowym i szesnastkowym.	2.3.2, 2.3.3	5.11.a.1
9	Wyznaczanie wartości wielomianu i zamiana liczb z dowolnego pozycyjnego systemu	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy numeryczne, np.: obliczanie wartości wielomianu za pomocą schematu Hornera, zastosowania schematu Hornera,	2.3.4, 2.3.5	5.11.c.2, 5.11.c.3

	liczbowego na dziesiętny za pomocą schematów Hornera.		reprezentację liczb w różnych systemach liczbowych, szybkie podnoszenie do potęgi.		
10	Reprezentacja danych liczbowych w komputerze.	1	Uczeń opisuje podstawowe algorytmy i stosuje: algorytmy na liczbach całkowitych, np.: reprezentacja liczb w dowolnym systemie pozycyjnym, w tym w dwójkowym i szesnastkowym oraz algorytmy numeryczne np.: zastosowania schematu Hornera: reprezentacja liczb w różnych systemach liczbowych, szybkie podnoszenie do potęgi.	2.3.6	5.11.a.1, 5.11.c.3
11	Błędy w obliczeniach	1	Uczeń potrafi wyjaśnić źródło błędów w obliczeniach komputerowych (błąd względny, błąd bezwzględny).	2.3.7	5.27
12	Sito Eratostenesa – generowanie i badanie liczb pierwszych.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy na liczbach całkowitych, np.: sprawdzanie, czy liczba jest liczbą pierwszą, doskonałą.	2.4.1, 2.4.2	5.11.a.2
13	Sprawdzian wiadomości	1		2.1 - 2.4.2	
14	Liniowe przeszukiwanie ciągu liczbowego.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (sortowania), np.: algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kulekowy.	2.5.1, 2.5.2	5.11.b.2
15	Znajdowanie minimalnego i maksymalnego elementu w zbiorze. Lider zbioru.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (sortowania), np.: jednoczesne znajdowanie największego i najmniejszego elementu w zbiorze: algorytm naiwny i optymalny.	2.6, 2.7	5.11.b.1
16	Sprawdzanie monotoniczności ciągu liczbowego.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (sortowania), np.: algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kulekowy.	2.8	5.11.b.2
17	Metody sortowania przez porównanie: porządkowanie bąbelkowe, przez wybór oraz przez wstawianie.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (sortowania), np.: algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kulekowy.	2.9.1	5.11.b.2
18	Sortowanie w czasie liniowym – porządkowanie przez zliczanie i porządkowanie kulekowe.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (sortowania), np.: algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kulekowy.	2.9.2	5.11.b.2
19	Jednoczesne znajdowanie minimalnego i maksymalnego elementu z zastosowaniem metody „dziel i zwyciężaj”	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (sortowania), np.: jednoczesne znajdowanie największego i najmniejszego elementu w zbiorze: algorytm naiwny i optymalny. Posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów.	2.10.1	5.11.b.1, 5.8
20	Sortowanie przez scalanie.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (sortowania), np.: algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kulekowy.	2.10.2	5.11.b.2
21	Sortowanie szybkie.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy wyszukiwania i porządkowania (sortowania), np.: algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kulekowy.	2.10.3	5.11.b.2
22	Metody numeryczne i obliczenia przybliżone – obliczanie wartości pierwiastka kwadratowego z liczby nieujemnej metodą Newtona-Raphsona.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy numeryczne, np.: obliczanie wartości pierwiastka kwadratowego.	2.11.1	5.11.c.1
23	Obliczanie pola obszaru ograniczonego wykresem funkcji.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy numeryczne, np.: obliczanie pola obszarów zamkniętych.	2.11.2	5.11.c.5
24	Znajdowanie przybliżonej wartości miejsca zerowego funkcji – metoda połowienia przedziałów.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy numeryczne, np.: wyznaczanie miejsc zerowych funkcji metodą połowienia.	2.11.3	5.11.c.4
25	Zastosowanie programowania zachłannego – problem plecakowy. Programowanie dynamiczne.	1	Uczeń stosuje podejście zachłanne w rozwiązywaniu problemów. Opisuje podstawowe algorytmy i stosuje: algorytmy na liczbach całkowitych, np.: wydawanie reszty metodą zachłanną.	2.12.1	5.10, 5.11.a.6
26	Algorytm wydawania reszty. Ćwiczenia.	1	Uczeń stosuje podejście zachłanne w rozwiązywaniu problemów. Opisuje podstawowe algorytmy i stosuje: algorytmy na liczbach całkowitych, np.: wydawanie reszty	2.12.2	5.10, 5.11.a.6



			metodą zachłanną.		
27	Sprawdzian wiadomości.	1		2.5.1-2.12.2	
28	Algorytmy na tekstach: palindromy i sortowanie tekstu.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy na tekstach, np.: sprawdzanie, czy dany ciąg znaków tworzy palindrom, porządkowanie alfabetyczne.	2.13.1, 2.13.2	5.11.d
29	Algorytmy na tekstach: anagramy	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy na tekstach, np.: sprawdzanie, czy dany ciąg znaków tworzy anagram.	2.13.3	5.11.d
30	Wyszukiwanie wzorca w tekście.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy na tekstach, np.: wyszukiwanie wzorca w tekście.	2.13.4	5.11.d.3
31	Wyznaczanie wartości wyrażenia zapisanego w odwrotnej notacji polskiej ONP	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy na tekstach, np.: obliczanie wartości wyrażenia podanego w postaci odwrotnej notacji polskiej.	2.13.5	5.11.d.4
32	Szyfrowanie symetryczne i asymetryczne – ćwiczenia.	1	Uczeń opisuje podstawowe algorytmy i stosuje algorytmy kompresji i szyfrowania, np.: kody znaków o zmiennej długości, np. alfabet Morse'a, kod Huffmana, szyfr Cezara, szyfr przedstawieniowy, szyfr z kluczem jawnym (RSA), wykorzystanie algorytmów szyfrowania, np. w podpisie elektronicznym.	2.14.1, 2.14.2	5.11.e
33	Sprawdzian wiadomości.	1		2.13.1-2.14.2	
<b>7. Programowanie w języku C++</b>					
Lp.	Temat lekcji	Liczba godzin do realizacji	Przewidywane osiągnięcia ucznia	Odwołanie do podręcznika	Odwołanie do podstawy programowej
34	Języki programowania – pojęcia, klasyfikacja, przykłady. Wprowadzenie do programowania.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.1, 3.2	5.12 – 5.26
35	Struktura programu. Operacje wejścia-wyjścia	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.2.1, 3.2.2	5.12 – 5.26
36	Zmienne, stałe, wskaźniki i referencje. Odniesienie do podręcznika:	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności	3.2.3	5.12 – 5.26

			algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.		
37	Wyrażenia arytmetyczne, relacje i operatory logiczne.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.2.4	5.12 – 5.26
38	Priorytety relacji i działań. Funkcje matematyczne.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.2.5, 3.2.6	5.12 – 5.26
39	Liczby losowe. Komentarze.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.2.7, 3.2.8	5.12 – 5.26

			zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.		
40	Podstawowe konstrukcje algorytmiczne - instrukcja przypisania, instrukcja złożona i instrukcja warunkowa.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.3.1, 3.3.2, 3.3.3	5.12 – 5.26
41	Instrukcja wyboru.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.3.4	5.12 – 5.26
42	Instrukcje iteracyjne.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.3.5	5.12 – 5.26
43	Instrukcje sterujące.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym	3.3.6	5.12 – 5.26

			środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.		
44	Proste typy danych. Ćwiczenia utrwalające przerobiony materiał.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	Ćwiczenia do podrozdziałów: 3.1 – 3.4	5.12 – 5.26
45	Sprawdzian wiadomości.	1		3.1 – 3.4	
46	Strukturalizacja programu – struktura funkcji.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.5.1	5.12 – 5.26
47	Zmienne lokalne i globalne.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.5.2	5.12 – 5.26
48	Metody przekazywania parametrów w funkcjach.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i	3.5.3	5.12 – 5.26

			wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.		
49	Metody przekazywania parametrów w funkcjach – ćwiczenia.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.5.3	5.12 – 5.26
50	Przeładowanie funkcji.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.5.4	5.12 – 5.26
51	Przeładowanie funkcji ćwiczenia	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i	3.5.4	5.12 – 5.26

			wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.		
52	Strukturalne typy danych – tablice jednowymiarowe.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.6.1	5.12 – 5.26
53	Strukturalne typy danych – tablice wielowymiarowe.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.6.1	5.12 – 5.26
54	Strukturalne typy danych – tablice. Ćwiczenia.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.6.1	5.12 – 5.26
55	Łańcuchy; tablice znaków.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez	3.6.2	5.12 – 5.26

			algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.		
56	Łańcuchy: typ string. Konwersje łańcuchów.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.6.2	5.12 – 5.26
57	Łańcuchy – ćwiczenia.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.5.2	5.12 – 5.26
58	Struktury.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.6.3	5.12 – 5.26

59	Struktury – ćwiczenia.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.6.3	5.12 – 5.26
60	Sprawdzian wiadomości.	1		3.4 - 3.6.3	5.12 – 5.26
61	Dynamiczne struktury danych.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.7	5.12 – 5.26
62	Dynamiczna struktura danych – stos i kolejka.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.7.1, 3.7.2	5.12 – 5.26
63	Dynamiczna struktura danych – lista.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemów; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym	3.7.3	5.12 – 5.26



			środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.		
64	Dynamiczna struktura danych – drzewo binarne.	1	Uczeń opisuje podstawowe algorytmy i stosuje konstrukcje rekurencyjne: drzewo binarne, dywan Sierpińskiego, płatek Kocha.	3.7.4	5.11.f.6
65	Plikowe operacje wejścia-wyjścia.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemu; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.8	5.12 – 5.26
66	Plikowe operacje wejścia-wyjścia. Ćwiczenia.	1	Uczeń: projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu; dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne; stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; opisuje własności algorytmów na podstawie ich analizy; ocenia zgodność algorytmu ze specyfikacją problemu; oblicza liczbę operacji wykonywanych przez algorytm; szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu; bada efektywność komputerowych rozwiązań problemu; przeprowadza komputerową realizację algorytmu i rozwiązania problemu; sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu; dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	3.8	5.12 – 5.26
67,	Sprawdzian wiadomości.	1		3.7 - 3.8	
<b>8. Projekt programistyczny</b>					
Lp.	Temat lekcji	Liczba godzin do realizacji	Przewidywane osiągnięcia ucznia	Odwołanie do podręcznika	Odwołanie do podstawy programowej
68	Projekt programistyczny. Metody realizacji projektów. Inżynieria oprogramowania.	1	Uczeń realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.	4.1	5.28
69	Projekt programistyczny. Założenia i cele projektu.	1	Uczeń realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.	4.2	5.28
70	Projekt programistyczny. Realizacja projektu.	1	Uczeń realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.	4.2	5.28
71, 72, 73,	Projekt programistyczny. Realizacja projektu.	3	Uczeń realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.	4.2	5.28
74, 75,	Projekt programistyczny. Prezentacja zrealizowanych projektów.	2	Uczeń realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.	4.2	5.28

<b>76 - 90</b>	Rozwiązywanie zadań maturalnych	14	<b>DLA KL IV INF, ELN, GR</b>		
<b>90 - 120</b>	Rozwiązywanie zadań maturalnych	30	<b>DLA KL IV ELN, GR</b>		