

## Plan wynikowy przedmiotu Informatyka poziom rozszerzony

### INFORMATYKA

Poziom rozszerzony, tematy pogrupowane działami

Dział	Tematy lekcji	Osiągnięcia ucznia – podstawowe (uczeń)	Osiągnięcia ucznia - rozszerzone (uczeń ponadto)
Lekcje organizacyjne	1. Regulamin pracowni. Przepisy BHP. Ocenianie, WSO, PSO. Sprawy organizacyjne.	zna regulamin i przepisy BHP; zna program nauczania;	
	Informatyka jako przedmiot maturalny. Omówienie podstawy programowej, wymagań maturalnych i kwestii organizacyjnych.	rozumie kwestię wybierania przedmiotu informatyka na maturze;	
Algorytmika – wstęp. Schematy blokowe.	2. Algorytmika – wprowadzenie.	zna różne systemy zapisu liczb, zna pojęcia algorytm, dane, wyniki, specyfikacja, algorytm liniowy, pętla, zmienna licznikowa, obsługa tablic, złożoność algorytmu, zna sposoby przedstawiania algorytmów i przedstawia je w sposób opisowy, za pomocą listy kroków lub schematu blokowego (wymienia i stosuje elementy schematu blokowego), przedstawia algorytmy z rozgałęzzeniami i pętlami, wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy, tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą, określa sytuacje warunkowe, podaje przykłady zadań, w których występują sytuacje warunkowe, wie, na czym polega powtarzanie tych samych operacji, potrafi omówić, na przykładzie, algorytm znajdowania najmniejszego z trzech elementów, potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzzeniami), zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego, podaje ich przykłady, wie, od czego zależy liczba powtórzeń, potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy, omawia i analizuje wybrane techniki sortowania w postaci gotowych schematów blokowych, potrafi zanalizować przebieg algorytmu dla przykładowych danych i ocenić w ten sposób jego poprawność, potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją, określa liczbę prostych działań zawartych w algorytmie, tworzy proste algorytmy na wzór poznanych na lekcji,	potrafi ocenić obecny poziom technologii informacyjnej i perspektywy jej rozwoju, przedstawia w różnych postaciach algorytmy szkolne z modyfikacjami, potrafi wyznaczyć złożoność obliczeniową i pamięciową różnych algorytmów, potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów, przedstawia dokładną specyfikację dowolnego zadania, zna znaczenie i działanie instrukcji symbolicznego języka programowania (pseudojęzyka). Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci, zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji), m.in. w pseudojęzyku, schemacie blokowym, zapisuje algorytmy z pętlą zagnieżdżoną, potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego, analizuje działanie algorytmu dla przykładowych danych, przestrzega zasad zapisu algorytmów w zadanej postaci (notacji), potrafi trafnie dobrać do algorytmu sposób prezentacji, stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia, analizuje algorytmy, w których występują iteracje, zna sposoby zakończenia iteracji, określa kroki iteracji, potrafi zapisać w wybranej notacji w tym np. algorytm sumowania n liczb, algorytm obliczania silni, znajdowania minimum w ciągu n liczb, algorytmy wyszukiwania określonych właściwości ciągu liczb, algorytm rozwiązywania równania liniowego, algorytm rozwiązywania układu równań, zna iteracyjną postać algorytmów, rozumie, co to jest złożoność algorytmu i potrafi określić liczbę operacji wykonywanych na elementach zbioru w wybranym algorytmie sortowania, potrafi porównać złożoność różnych algorytmów tego samego zadania dla tych samych danych, wie, kiedy algorytm jest uniwersalny, zna odpowiednie wzory.
	3. Cechy algorytmów.		
	4. Schematy blokowe - wstęp		
	5. Ćw. – przykłady prostych algorytmów szkolnych.		
	6. Schematy blokowe – cd. Pętle w algorytmach.		
	7. Algorytmy tablicowe.		
	8. Wyznaczanie cech ciągów liczb. Złożoność obliczeniowa algorytmów.		
	9. Schematy blokowe - podsumowanie		
	Programowanie - wstęp	10. Programowanie – wstęp.	wymienia sposoby reprezentacji liczb w komputerze, zna pojęcia: kod źródłowy, edytor, translator, kod wynikowy, porusza się płynnie w środowisku programistycznym, zapisuje i otwiera pliki, operuje na blokach i strukturze programu, zna słowa kluczowe, komentarze, identyfikatory, instrukcje I/O, przypisania, instrukcje warunkowe i powtórzeniowe (potrafi tworzyć poprawne warunki końcowe pętli) generuje liczby pseudolosowe, zapisuje szkielec programu, używa komentarzy, konstruuje identyfikatory, zapisuje prawidłowo liczby całkowite i rzeczywiste, wczytuje dane z klawiatury, wyświetla dane na ekranie, potrafi deklorować zmienne (prostych typów i prostych typów złożonych), oblicza podstawowe wielkości na podstawie wzorów matematycznych, stosuje operatory arytmetyczne, logiczne i relacji oraz podstawowe
11. Środowisko programistyczne.			
12. Wykorzystanie komp. do obliczeń.			
13. Ćw. – podst. typy, operatory, funkcje.			
14. Instrukcje warunkowe			
15. Ćw. – IF, SWITCH.			
16. Instrukcje powtórzeniowe			
17. Ćw.- FOR, WHILE			
18. Podstawy obsługi tablic.			

	19. Tablice – rozwiązywanie zadań. 20. Strukturalizacja programu. 21. Ćw. – funkcje	funkcje standardowe, rozumie zagadnienie liczb losowych, zna proste algorytmy liniowe i iteracyjne, zna klasyfikację języków programowania, zna ogólną budowę programu w języku programowania i najważniejsze elementy języka – słowa kluczowe, instrukcje, wyrażenia, zasady składni, potrafi zapisać prosty algorytm liniowy w języku programowania, potrafi pisać proste programy na wzór przykładów prezentowanych na lekcji, zapisuje program w czytelnej postaci – stosuje wcięcia, komentarze, potrafi samodzielnie pisać proste programy liczbowe, rozumie pojęcia: implementacja, kompilacja, uruchomienie, testowanie, rozumie znaczenie i działanie instrukcji wybranego języka programowania, rozróżnia i poprawia błędy kompilacji i błędy wykonania, potrafi zrealizować algorytmy iteracyjne w języku programowania, zna podstawowe zasady poprawnego programowania, testuje tworzone programy, wie, jak uniknąć problemów, takich jak np. zapętlenie się programu,	wysokiego poziomu a językiem maszynowym, potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów, ocenia efektywność działania programu, bardzo dobrze zna składnię języka, sprawnie korzysta z dodatkowej, fachowej literatury.	
Programowanie strukturalne. Metody reprezentowania informacji.	22. Strukturalizacja programu – cd. 23. Ćw. - funkcje 24. Zmienne łańcuchowe. 25. Podsumowanie (funkcje, układy liczbowe.) 26. Ćw.- podst. działania na zm. typu string. 27. Zmienne łańcuchowe - cd 28. Ćw. – palindromy, anagramy 29. Systemy liczbowe 30. Struktury danych w c++ 31. Ćw. – def. i obsługa struktur. 32. Obsługa plików tekstowych w c++. 33. Ćw. – podstawowe operacje na plikach. 34. Ćw. – podstawowe operacje na plikach. 35. Ćw. – podstawowe operacje na plikach. 36. Schemat Hornera. Złożoność schematu. 37. Ćw. - przekazywanie do i z typ referencyjny 38. Szybkie potęgowanie, potęga LP i PL. 39. Rekurencja, iteracja. 40. Rekurencyjny algorytm Hornera. 41. Ruchy Browna, ciąg Fibonacciego. 42. Wieże Hanoi – algorytm rekurencyjny 43. Rekurencja w zadaniach maturalnych.	zna pojęcia funkcji (składnia, przekazywanie parametrów, typ referencyjny, procedury i funkcje bezparametrowe, parametrowe, zmienne globalne i lokalne, zasięg zmiennych); zna i rozumie algorytm szybkiego potęgowania, schematu Hornera i Wieży Hanoi; rozumie pojęcie rekurencji; deklaruje i stosuje typ łańcuchowy (kopiowanie łańcuchów, łączenie łańcuchów, przypisywanie zmiennym łańcuchowym wartości, proste algorytmy sprawdzające własności napisów, proste algorytmy liczbowe, określa długość łańcucha, używa funkcji określającej pozycję znaku w łańcuchu, zna algorytm tworzenia statystyki tekstu); wykorzystuje typ strukturalny i plikowy do obsługi prostych baz danych, zna różne systemy zapisu liczb, potrafi przeliczać liczby między systemami, pojęcie biblioteki STL, zawartość	rozumie i wykorzystuje przesłanianie zmiennych i uniwersalność procedur i funkcji; potrafi wskazać wywołania stosu rekurencyjnego; płynnie wykorzystuje rozbudowane operacje na łańcuchach (zna sposoby odwoływania się do poszczególnych liter, zamiany liter małych na duże bez użycia funkcji <i>toupper</i> , różne algorytmy sprawdzające własności napisów, algorytmy liczbowe, algorytmy rekurencyjne, potrafi zabezpieczyć program przed wprowadzeniem wartości nie będącej liczbą; zamienia zmienną łańcuchową na liczbową i odwrotnie; tworzy rozbudowany projekt obsługi baz danych typu magazyn, biblioteka, szkoła itp. z wykorzystaniem obsługi plików, podstawowe polecenia STL (wektor)	
	Zarządzanie pamięcią	44. Typ wskaźnikowy 45. Ćw. – operacje na zm. wskaźnikowych 46. Wskaźniki do tablic i struktur 47. Ćw. – wskaźniki do tablic i struktur	zna pojęcie adresu komórek pamięci i typu wskaźnikowego; potrafi wykonywać proste operacje na zmiennych z wykorzystaniem odwołania się poprzez adres; zna określenia wskaźnik do tablicy i	sprawnie operuje na zmiennych statycznych i na zmiennych dynamicznych; rozumie istotę dynamicznego zarządzania pamięcią i jego

	48. Dynamiczne zarządzanie pamięcią	wskaźnik do struktur; potrafi odwołać się do pól struktury i je modyfikować; zna podstawowe polecenia zarządzania pamięcią (new, delete)	konsekwencje; wykorzystuje wskaźniki w implementowaniu algorytmów;
	49. New, delete – ćw.		
	50. Podsumowanie wskaźników		
Porządkowanie i poszukiwanie informacji w zbiorze.	51. Porządkowanie danych – wprowadzenie.	zna zamianę wartości dwóch zmiennych, zna działanie algorytmów: bubblesort, selectionsort, mergesort, quicksort; rozumie pojęcie „in situ”, zna klasę złożoności algorytmu, zna scalanie ciągów uporządkowanych, zna zasadę „dziel i zwyciężaj”, sortuje algorytmem losowe ciągi liczb; opisuje algorytm scalania liczb uporządkowanych, generuje losowe ciągi liczb uporządkowanych; zna pojęcie rekurencji i iteracji, opisuje algorytmy: NWD – naiwny, Euklides – iteracyjny; silnia – definicja rekurencyjna i iteracyjna, zna algorytm wykonywania działań na ułamkach; przedstawia i programuje jedną z metod Euklidesa, oraz naiwny, wykonuje działania na ułamkach oraz program realizujący te działania; zna zamianę wartości dwóch zmiennych, zna klasę złożoności algorytmu, zna scalanie ciągów uporządkowanych, zna zasadę „dziel i zwyciężaj”, sortuje algorytmem losowe ciągi liczb; opisuje algorytm scalania liczb uporządkowanych, generuje losowe ciągi liczb uporządkowanych; zna pojęcie rekurencji i iteracji, opisuje algorytmy: NWD – naiwny, Euklides – iteracyjny; silnia – definicja rekurencyjna i iteracyjna, zna algorytm wykonywania działań na ułamkach; przedstawia i programuje jedną z metod Euklidesa, oraz naiwny, wykonuje działania na ułamkach oraz program realizujący te działania	zna modyfikacje algorytmów sortowania; dokładnie określa ich złożoność obliczeniową i pamięciową; dokonuje pomiaru czasu działania; zna mechanizm sortowania przez scalanie, rozumie zagadnienie efektywności algorytmu, problem sortowania leksykograficznego, zna inne algorytmy sortowania: przez wstawianie, kubelkowe, przedstawia w kodzie źródłowym zmodyfikowane wersje algorytmów sortowania; przedstawia mechanizm działania poznanych algorytmów na przykładzie liczbowym; rozumie algorytm Euklidesa – rekurencyjne, określa cechy iteracji i rekurencji i określa różnice między nimi rozumie problemy z rekurencją, określa różnicę w złożoności obliczeniowej NWD met. naiwną i Euklidesa; programuje algorytmy NWD Euklidesa iteracyjnie i rekurencyjnie; rozumie algorytm przelewania wody i potrafi wykorzystać go w rozwiązywaniu zadań ,zna modyfikacje algorytmów sortowania; przedstawia w kodzie źródłowym zmodyfikowane wersje algorytmów sortowania; przedstawia mechanizm działania poznanych algorytmów na przykładzie liczbowym;
	52. Ćw. – SelectSort.		
	53. Scalanie ciągów uporządkowanych.		
	54. Ćw. – MergeSort.		
	55. Sortowanie „szybkie” QuickSort.		
	56. Ćw. – QuickSort.		
	57. Sortowanie przez zliczanie.		
	58. Ćw. – CountingSort.		
	59. BucketSort - sortowanie kubelkowe dla elem. całkowitych.		
	60. Przeszukiwanie zbioru danych.		
	61. Ćw. – liniowe, binarne.		
	62. InsertionSort - sortowanie przez wstawianie liniowe lub wstawianie binarne		
	63. InsertionSort - sortowanie przez wstawianie liniowe lub wstawianie binarne		
	64. Sortowania – podsumowanie.		
	65. STL – sortowanie, sortowanie z przeciążeniem, wektor.		
	66. Przeszukiwanie zbioru danych.		
	67. Ćw. – liniowe, binarne.		
	68. Centralność danych – wstęp.		
	69. Wyszukiwanie mody zbioru, lidera, idola.		
	70. Wyszukiwanie k-tego największego elementu.		
71. Podsumowanie porządkowania			
72. NWD – naiwnie i z modyfikacjami			
73. NWD – algorytm Euklidesa (z różnicą).			
74. Algorytm przelewania wody			
75. Ćw. – „rozszerzony Euklides”			
76. Liczby pierwsze – algorytm naiwny			
77. Liczby pierwsze – sito Eratostenesa.			
Przetwarzanie informacji (szyfrowa	78. Transliteracja - wprowadzenie	zna podstawowe pojęcia związane z liczbami pierwszymi; opisuje algorytm naiwny i sita Eratostenesa do generowania liczb pierwszych; zna pojęcie szyfrowania symetrycznego i asymetrycznego;	potrafi analizować złożoność obliczeniową algorytmów generowania liczb pierwszych; wykorzystuje sito Eratostenesa w rozwiązywaniu zadań; implementuje poznane algorytmy szyfrowania z wykorzystaniem
	79. Ćw. – szyfr Cezara, ROT13		
	80. Szyfrowanie – cd.		
	81. Ćw. – liczby losowe, różnica symetryczna.		

	82. Szyfrowanie – cd.	opisuje metody szyfrowania Cezara, ROT13, Vigenere, Playfair’a, RSA; potrafi sklasyfikować metody szyfrowania pod względem bezpieczeństwa; opisuje podstawowe cechy podpisu elektronicznego; zna jego ideę i potrafi wykorzystać go w praktyce; określa sytuacje z życia codziennego z wykorzystaniem szyfrowania i podpisu; potrafi napisać programy szyfrujące / deszyfrujące z wykorzystaniem poznanych metod szyfrowania; zna zagadnienie kompresji danych; rozróżnia kompresję stratną i bezstratną; opisuje podstawowe algorytmy stosowane przy kompresji ; potrafi napisać kod kompresji naiwnej; zna zasady tworzenia kodu Hoffmana; zna pojęcie prefiksowości kodu	klucza napisowego, zarodka liczb losowych, funkcji różnicy symetrycznej; rozumie wpływ długości klucza szyfrowania na poziom jego bezpieczeństwa; sprawnie obsługuje tablice wielowymiarowe; rozumie algorytm RSA , zna jego matematyczne podstawy i potrafi to wytłumaczyć w kontekście podpisu elektronicznego; potrafi napisać program generujący klucze RSA; szyfrujący / deszyfrujący metodą RSA; potrafi zanalizować podstawowe algorytmy kompresji bezstratnej; określa stopień kompresji danych; wykorzystuje kod Huffmana do tworzenia kodów prefiksowych
	83. Ćw. – szyfr Vigenere		
	84. Tablice dwuwymiarowe, n-wymiarowe		
	85. Ćw. – szyfr Playfair’a		
	86. Elementy kryptografii współczesnej. Algorytm RSA		
	87. Podpis elektroniczny		
	88. Elektroniczna Platforma Usług Administracji Publicznej		
	89. Kompresja danych		
	90. Ćw. – algorytmy kompresji, kod Huffmana		
Programowanie zachłanne i dynamiczne	91. Programowanie zachłanne	rozumie „technikę zachłanną”; zna problem wydawania reszty i na tej podstawie potrafi określać optymalność i dopuszczalność rozwiązań; potrafi omówić zachłanne i dynamiczne metody rozwiązywania problemów plecakowych i najkrótszej drogi; zna ich złożoność obliczeniową i pamięciową, potrafi korzystać z STL, zna gotowe komponenty	potrafi zaimplementować algorytmy zachłanne i dynamiczne problemów plecakowych; projektuje odpowiednie struktury danych; rozumie problem „trudności” rozwiązań dynamicznych przy dużych zestawach danych;
	92. Wydawanie reszty		
	93. Pakowanie plecaka ogólnego, decyzyjnego.		
	94. Pakowanie plecaka decyzyjnego zachłanne.		
	95. Porządkowanie ułamków; przeciążanie <		
	96. Plecak ogólny, decyzyjny dynamicznie		
	97. Plecak - implementacja.		
	98. Najkrótsza droga – algorytm dynamiczny		
	99. Najkrótsza droga – algorytm dynamiczny		
Obliczenia numeryczne	100. Obliczenia numeryczne – wstęp. Błędy obliczeń. Niestabilność numeryczna	zna konsekwencje reprezentacji danych w komputerze; potrafi porównywać dane niecałkowite; zna pojęcie szacunku obliczeń; rozumie problem dokładności obliczeń i powstawania niestabilności algorytmów; potrafi przedstawić algorytm „szkolny” i stabilny równania kwadratowego; zna algorytm Herona ; wykorzystuje schemat Hornera do obliczenia wartości wielomianu n-tego stopnia; zna metodę bisekcji na znalezienie przybliżeń miejsc zerowych; zna pojęcie kwadratur (całek numerycznych) i potrafi to wykorzystać przy obliczaniu pola pod krzywą;	znajduje w algorytmach miejsca z potencjalną niestabilnością obliczeniową; potrafi usuwać niestabilność numeryczną; rozumie istotę iteracyjnych metod rozwiązywania problemu z narzuconą dokładnością i potrafi to wykorzystać do obliczeń na podstawie wzorów, w których występują nieskończona liczba elementów np. obliczanie liczby Pi;
	101. Równanie kwadratowe metoda „szkolna”, metoda stabilna		
	102. Pierwiastki kwadratowe z liczby.		
	103. Metoda Herona.		
	104. Wielomian n-tego stopnia.		
	105. Miejsca zerowe wielomianu. Bisekcja.		
	106. Całkowanie numeryczne.		
	107. Kwadratura prostokątów, trapezów.		
	108. Podsumowanie obliczeń numerycznych.		
Geometria obliczeniowa	109. Geometria obliczeniowa – wstęp.	zna podstawowe problemy reprezentacji danych w komputerze; zna pojęcie macierzy, potrafi obliczyć wyznacznik 2 i 3-ego stopnia; zna podstawowe algorytmy geometryczne: po której stronie leży punkt; czy 2 punkty leżą po tej samej stronie; czy punkt należy do odcinka; czy 2 odcinki się przecinają; zna twierdzenie o trójkącie kratowym; umie obliczyć pole	reprezentuje macierz w sposób statyczny lub dynamiczny w zależności od potrzeb; zna pojęcie iloczynu skalarnego; zna algorytmy wyznaczające otoczkę wypukłą (Grahama, Jarvisa) i potrafi to wykorzystać w zadaniach; potrafi wykorzystać biblioteki graficzne do wizualizacji fraktali; zna określenia zbiorów Julii, Mandelbrota
	110. Wyznaczniki macierzy.		
	111. Geometria – podstawowe algorytmy.		
	112. Geometria – podstawowe algorytmy.		
	113. Algorytmy geometryczne – trójkąt kratowy.		
	114. Iloczyn wektorowy.		

	115. Pole równoległoboku, trójkąta, wielokąta.	trójkąta i na tej podstawie sprawdzić przynależność punktu do wielokąta wypukłego; zna pojęcie iloczynu wektorowego i na jego podstawie potrafi wyznaczyć pole równoległoboku, trójkąta i wielokąta nie wypukłego; zna pojęcie fraktal; potrafi zdefiniować rekurencyjne podstawowe fraktale	
	116. Podstawy grafiki – allegro.		
	117. Fraktale – wstęp.		
	118. Płatek Kocha, drzewo binarne, dywan Sierpińskiego		
	119. Podsumowanie geometrii obliczeniowej.		
Przeszukiwanie tekstu	120. Przetwarzanie tekstów – podstawowe pojęcia.	zna podstawowe pojęcia: alfabet, łańcuch, wzorzec, prefiks, sufiks, prefikso-sufiks; potrafi wyszukiwać wzorzec naiwnie i z wartownikami; rozumie ideę wartowników; zna podstawowe operacje na łańcuchach; rozumie znaczenie algorytmów wyszukiwania wzorca w tekście; rozumie ideę algorytmu Morrisa_Pratta oraz Boyera-Moore'a.	potrafi wyznaczyć tablicę MPNext i zaimplementować algorytm MP; wie jak wykorzystać tablicę Last w algorytmie Boyera-Moore'a;
	121. Wyszukiwanie wzorca – naiwnie.		
	122. Wyszukiwanie wzorca naiwnie z wartownikami.		
	123. Wyszukiwanie wzorca – algorytm KMP.		
	124. Algorytm Boyera –Moore'a.		
	125. Podsumowanie przetwarzania tekstów.		
Arkusz kalkulacyjny	126. Arkusz kalkulacyjny – sprawdzenie wiadomości i umiejętności	zna pojęcie arkusza, komórki, zakresu komórek, zna możliwe działania wykonywane na komórkach, pojęcie funkcji i formuły, typy adresowań, typy danych wprowadzanych do komórki arkusza, serie danych, rodzaje operatorów, elementy wykresu; zna poprawność obliczeniową i dokładność arkusza, funkcję warunkową; zna warunki utworzenia i zastosowanie sum częściowych (pośrednich) i tabel przestawnych; zna funkcje przeszukiwania baz danych licz.jeżeli, wyszukaj.pionowo, suma.jeżeli; zna możliwości zastosowania solvera; zna pojęcie i sposoby tworzenia i wykorzystanie makrodefinicji; wykonuje obliczenia, stosuje podstawowe funkcje w formułach, formatuje komórki, kopiuje zawartość komórek, tworzy wykresy w oparciu o serie danych, formatuje wykresy, blokuje okienka, obsługuje bazę poprzez formularz; wyszukuje, sortuje i filtruje dane, stosuje autofiltr; stosuje bazy do korespondencji seryjnej; tworzy arkusze symulujące prosty eksperyment; używa poprawnie pomocy,	stosuje filtr zaawansowany, nadaje nazwy zakresom komórek, stosuje funkcje częstość, licz.jeżeli; stosuje konspektowanie, sumy złożone, tworzy układ tabel przestawnych i wykresów przestawnych.; potrafi przeprowadzić hipotetyczny eksperyment z wykorzystaniem arkusza kalkulacyjnego; zna parametry konfiguracyjne solvera, analizuje zaawansowane dane w oparciu o solver; edytuje makrodefinicje (w VBA tworzy zaawansowane formularze); potrafi wykorzystać arkusz do implementowania algorytmów dyskretnych i numerycznych
	127. Arkusz – geometryczna interpretacja danych.		
	128. Wykresy funkcji trygonometrycznych		
	129. Symulacja zjawisk. Krzywe Lissajoux.		
	130. Symulacja zjawisk.		
	131. Symulacja eksperymentów.		
	132. Bazy danych w arkuszu.		
	133. Sumy pośrednie, tabele przestawne, korespondencja seryjna.		
	134. Automatyzacja – makrodefinicje.		
	135. Makra – formanty		
	136. Optymalizacja zadań – Solver.		
	137. Rozwiązywanie zadań maturalnych.		
	138. Rozwiązywanie zadań maturalnych.		
	139. Rozwiązywanie zadań maturalnych.		
140. Rozwiązywanie zadań maturalnych.			
141. Podsumowanie arkusza kalkulacyjnego			
Komputerowe środowisko pracy.	142. Elektronika cyfrowa; bramki logiczne	zna podst. pojęcia informatyki, podst. budowę komp., sposoby i rodz. kodowania danych, systemy liczbowe: binarny oktalny, heksadecymalny; przelicza liczby między systemami; charakteryzuje ideową budowę komputera, potrafi wyjaśnić zakres zastosowania technologii informacyjnej zna podstawowe bloki	zna generacje komputerów, omawia różnice pomiędzy poszczególnymi generacjami, zna budowę diody, lampy elektronowej, tranzystora npn i pnp, potrafi ocenić obecny poziom technologii informacyjnej i perspektywy jej rozwoju; zna parametry charakteryzujące w/w
	143. Układy z pamięcią - przerzutniki		
	144. Budowa komputera - podzespoły		
	145. Systemy operacyjne – Windows (przypomnienie)		
	146. Ubuntu – środowisko Linux; możliwości		

	147. Ubuntu – obsługa systemu Linux	funkcjonalne komputera, budowę płyty głównej, podstawowe urządzenia I/O (klawiatura, urządzenie wskazujące, skaner, tablet, kontrolery gry, karta sieciowa, dźwiękowa, modem, napędy dysków, wyświetlacz z kartą graficzną, drukarka, ploter), podział pamięci rozumie ideę „Open Source”; zna określenia GNU GPL; wykorzystuje podstawowe aplikacje środowiska Ubuntu;	podzespoły, zna funkcje i bloki funkcjonalne procesora, charakteryzuje poszczególne bloki płyty głównej; potrafi zainstalować środowisko Ubuntu i skonfigurować je do własnych potrzeb;
Sieci komputerowe	148. Sieci komputerowe - wprowadzenie	zna składniki i rodzaje sieci, pojęcie protokołu, charakteryzuje protokół TCP/IP, sposoby łączenia komputerów w sieć; zna topologie sieci, funkcje serwera, pojęcie wolumentu, podstawowe polecenia serwera, mapowanie charakteryzuje budowę Internetu, zadania protokołów, zna możliwości połączenia komputera z siecią, pojęcie i właściwości hipertekst charakteryzuje WWW, potrafi rozpoznać podstawowe człony nazwy domenowej, budowę URL, podstawowe możliwości przeglądarki, adresy wyszukiwarek i katalogów stron, zna pojęcie hipertekstu i jego właściwości, operatory stosowane przy wyszukiwaniu; charakteryzuje e-mail, listserv, usenet news, zna możliwości programów poczty, problematykę wirusową, pola odbiorców, konfigurację konta, adresy serwerów, moderowanie, zna usługi IRC, chat, Instant Messenger, VoIP, FTP, P2P, telnet, ssh, ping, prawne aspekty użytkowania sieci P2P, podaje najważniejsze cechy usług, zastosowanie i możliwości komunikatorów, bramki SMS; wykorzystuje narzędzia do komunikacji w sieci(np. NetMeeting), określa IP komputera, wyznacza IP sieci, broadcast, udostępnia i zabezpiecza zasoby, loguje się do serwera, wysyła komunikaty sieciowe, przelogowuje się bez przerwania pracy, korzysta z zasobów serwera;obsługuje przeglądarkę, obsługuje mechanizm zakładek/ulubionych, wyszukuje wiadomości w Internecie, pobiera dane z Internetu, wysyła wiadomości pocztowe, konfiguruje tożsamości i skrzynki emailowe, używa książki adresowej, instaluje oprogramowanie komunikatora, porozumiewa się komunikatorem i programem VoIP, użytkuje klienta FTP	charakteryzuje sieci, zna sposób działania urz. (hub, switch, router, bridge, gateway), zna charakterystykę zabezpieczeń zasobów w sieci, składnię UNC; zna charakterystykę topologii, model ISO/OSI, technologie sieciowe sieci WAN, pojęcie grup, praw grupowych, filtrów IRF, funkcje bezpieczeństwa danych w sieciach;zna historię Internetu, potrafi wyznaczyć podsieci danej sieci, ilość hostów podsieci, zadania TCP, pojęcie enkapsulacji i komutacji pakietów, charakteryzuje protokoły, maski sieci, zna DNS, konfigurację przeglądarki, zabezpieczenia prywatnych danych i przeglądarek, przeglądanie stron w trybie off-line, pojęcie, cechy i zadania portali, mechanizm działania wyszukiwarek, multiwyszukiwarek, katalogów stron, cookie, cache, zna netykię ; stosowanie PGP, usługi i konfiguracja programu pocztowego, przyporządkowuje usługi do protokołów pop3, imap, nntp, smtp, standard MIME,;zna klientów IRC, FTP, telnet, serwery, klientów, klientów danych strumieniowych, zastosowania telnet, ssh, konfiguruje połączenie sieci, odzyskuje pliki z kosza sieciowego, proponuje adresy witryn firm, konfiguruje przeglądarkę, tworzy obrazy off-line witryn zabezpiecza przesyłki, używa grupy dyskusyjne, reguły wiadomości, wizytówki, podpisy, żądania potwierdzenia, priorytety, szyfruje wiadomości, pobiera klucz; sprawdza adres IP serwera, sprawdza trasę pakietu w sieci
	149. Sieci – model OSI, protokoły sieciowe		
	150. Sieci – topologie, urządzenia, kable		
	151. Sieci – topologie, urządzenia, kable – ćw.		
	152. Sieci – elementy łączące		
	153. Internet – podst. pojęcia; protokół TCP/IP		
	154. Internet – komutacja, adresacja IP, DNS		
	155. Internet – usługi ( www, poczta)http, pop3, smtp, telnet, ftp,		
	156. Internet – rodzaje zagrożeń i bezpieczeństwo		
	157. Internet – netykieta		

Relacyjne bazy danych	158. Wstęp do baz danych	zna podstawowe problemy organizacji baz danych, potrafi projektować tabele tworzące relacyjny model baz danych, rozumie pojęcie kluczy; zna funkcje najważniejszych obiektów relacyjnych b.d., potrafi projektować zaawansowane formularze, kwerendy i raporty, potrafi tworzyć zapytania (kwerendy)	potrafi tworzyć formularze stosując słowniki; zna pojęcie kreatora odnośników; projektuje relacyjną bazę danych składającą się z wielu tabel połączonych relacją (na zadany temat); swobodnie projektuje formularz i raport; stosuje zasady definiowania kluczy podstawowych; łączy informacje z bazy danych z dokumentami innych programów, np. edytora tekstu czy arkusza kalkulacyjnego; stosuje różne rodzaje kwerend; samodzielnie ustala zawartość bazy (rodzaj informacji); samodzielnie projektuje wygląd formularza i raportu; sprawnie korzysta z dodatkowej, fachowej literatury;
	159. Środowisko systemu zarządzania baz danych.		
	160. Tworzenie struktury b.d.		
	161. Tabele – obiekt b.d.		
	162. Anomalie i redundancje.		
	163. Normalizacja baz danych.		
	164. Relacje w b.d.(rodzaje, tworzenie, klucze).		
	165. Wyszukiwanie inf. w b.d.		
	166. Kwerendy – rodz., tworzenie, zastosowanie, SQL		
	167. Obsługa baz danych – formularze.		
	168. Obiekt: formularze - ćwiczenia		
	169. Bazy danych – rozwiązywanie zadań maturalnych.		
170. Bazy danych – rozwiązywanie zadań maturalnych.			
171. Bazy danych – rozwiązywanie zadań maturalnych.			
172. Bazy danych – rozwiązywanie zadań maturalnych.			
Standard Query Language	173. Podstawy języka SQL	zna w podstawowym zakresie składnię języka SQL, zna podstawowe polecenia SQL: SELECT FROM WHERE; stosuje operatory i kryteria, sortowanie,	potrafi tworzyć zapytania SQL i wykorzystując złożone polecenia SQL; obliczenia w SQL, zna i potrafi użyć środowisko serwera SQL;
	174. Podstawy SQL – ćwiczenia.		
	175. Funkcje agregujące.		
	176. Relacje.		
	177. Tworzenie tabel, struktur tabel.		
	178. Wypełnianie, aktualizacja i usuwanie danych SQL.		
Grafika komputerowa	179. Grafika komputerowa - wprowadzenie	zna pojęcia grafika komp., modele barw RGB, CMYK, grafika rastrowa, wektorowa; potrafi świadomie stosować formaty plików graficznych; obsługuje aparat cyfrowy i skaner; wykonuje podst. operacje na obiektach i wbudowanych figurach wektorowych (przesuwanie, skalowanie, ścieśnianie, obracanie, ścinanie, tworzenie kopii), stosuje tekst akapitowy i ozdobny	sprawnie posługuje się edytorem graficznym przy obróbce obrazu; świadomie wykorzystuje różnego rodzaju efekty przekształcania obrazu; wykorzystuje warstwy i maski; potrafi obliczyć wielkość pliku przy określonej rozdzielczości i głębi koloru; stosuje wypełnienia, przezroczystości i cieniowanie interaktywne
	180. CorelPhotoPaint / GIMP – edycja grafiki rastrowej.		
	181. CorelPhotoPaint / GIMP – edycja grafiki rastrowej.		
	182. Edytor grafiki wektorowej – Corel Draw		
	183. Grafika wektorowa – operacje na obiektach		
Strony www	184. Serwis www – ramki, tabele, inne elementy	zna mechanizm znaczników HTML; umie wygenerować kod HTML i go modyfikować; potrafi optymalizować dobór grafiki; potrafi przekazać sterowanie do kolejnego dokumentu, generuje tabele i ramki; zna podstawy PHP; rozumie funkcjonowanie języka zapytań SQL i bazy MySQL; rozumie konieczność funkcjonowania mechanizmu baz danych na serwerach internetowych;	potrafi samodzielnie uczyć się praktycznego wykorzystania znaczników i ich atrybutów, zna zasady wykorzystania stylów, potrafi dowolnie pozycjonować tekst i grafikę, potrafi zbudować serwis www za pośrednictwem dowolnej techniki opartej na wykorzystaniu „wtyczki”; tworzy strony z wykorzystaniem instrukcji PHP i SQL; potrafi samodzielnie obsługiwać formularze i pliki; zarządzać bazami, tabelami, rekordami
	185. Formularze na stronach www		
	186. Java Script - wprowadzenie; przykłady skryptów		
	187. Instalacja serwera WAMP (WWW, MySQL, PHP)		
	188. Pierwsze skrypty w języku PHP		
	189. Budowa skryptów PHP (współpracy z b. danych)		
	190. Tworzenie kwerend.		

Pozostałe godziny przeznaczone na powtórzenia i sprawdziany.



## **Wymagania edukacyjne niezbędne do uzyskania poszczególnych ocen klasyfikacyjnych**

### **Ocena dopuszczająca:**

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) posługiwać się użytkowymi programami biurowymi w zakresie podstawowym
- b) wyszukiwać informacje w Internecie
- c) podstawy składni języka programowania C++
- d) pisać proste programy na wzór przykładów prezentowanych podczas lekcji
- e) tworzyć strony WWW przy użyciu języka HTML w zakresie podstawowym
- f) wykorzystywać arkusz kalkulacyjny do badania podstawowych cech funkcji matematycznych
- g) rozwiązywać proste problemy numeryczne na wzór przykładów realizowanych na zajęciach
- h) tworzyć tabele i proste kwerendy w bazach danych według przedstawionych przykładów
- i) pisać podstawowe polecenie SQL
- j) podstawowe metody przenoszenia plików za pomocą sieci
- k) tworzyć proste strony WWW z wykorzystaniem języka PHP

### **Ocena dostateczna:**

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) biegle posługiwać się użytkowymi programami biurowym
- b) wyszukiwać informacje w Internecie z wykorzystaniem zaawansowanych technik
- c) składnię języka programowania C++
- d) pisać programy rozwiązujące proste problemy liczbowe i tekstowe
- e) tworzyć strony WWW przy użyciu języka HTML, podstaw stylów kaskadowych CSS i języka PHP
- f) sprawnie wykorzystuje arkusz kalkulacyjny do badania cech funkcji matematycznych
- g) rozwiązuje problemy numeryczne na podstawie wzorów przedstawionych na zajęciach
- h) tworzy tabele, kwerendy i proste relacje w bazach danych według przedstawionych przykładów
- i) tworzy podstawowe polecenia SQL z użyciem dodatkowych klauzul

### **Ocena dobra:**

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) biegle posługuje się użytkowymi programami biurowym
- b) wyszukuje informacje w Internecie z wykorzystaniem zaawansowanych technik
- c) dobra znajomość składni języka programowania C++
- d) pisze programy rozwiązujące rozmaite problemy liczbowe i tekstowe z wykorzystaniem technik programowania proceduralnego
- e) rozwiązuje złożone problemy numeryczne
- f) zna i stosuje standardowe algorytmy
- g) tworzy złożone strony WWW w systemie Linux z wykorzystaniem zaawansowanych stylów kaskadowych CSS oraz skryptów w języku Javascript
- h) tworzy kwerendy zaawansowane w oparciu o złożone bazy danych
- i) tworzy proste polecenia SQL w oparciu o kilka tabel

## Ocena bardzo dobra:

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) biegle posługiwać się użytkowymi programami biurowym
- b) wyszukiwać informacje w Internecie z wykorzystaniem zaawansowanych technik
- c) bardzo dobrze składnię języka programowania C++
- d) pisze programy rozwiązujące złożone problemy liczbowe i tekstowe z wykorzystaniem techniki programowania proceduralnego, zna podstawy programowania obiektowego
- e) tworzenie złożonych stron WWW przy użyciu języka HTML i zaawansowanych stylów kaskadowych CSS
- f) biegle wykorzystuje arkusz kalkulacyjny do rozwiązywania różnych zadań matematycznych i z życia codziennego
- g) rozwiązuje złożone problemy numeryczne
- h) zna i stosuje standardowe algorytmy (np. dotyczących dynamicznych struktur danych, geometrycznych)
- i) tworzy tabele, złożone kwerendy i relacje w bazach danych według przedstawionych projektów
- j) tworzy złożone polecenia SQL w oparciu o kilka tabel

## Ocena celująca:

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) zna dodatkowe biblioteki, sprawnie proponuje różne funkcje obsługujące to samo zadanie, potrafi omówić różnice C++
- b) pisze programy rozwiązujące złożone problemy liczbowe i tekstowe z wykorzystaniem techniki programowania proceduralnego i obiektowego
- c) tworzenie złożonych stron WWW przy użyciu różnych technik tworzenia stron
- d) biegle wykorzystuje arkusz kalkulacyjny do badania rozmaitych cech funkcji matematycznych również z wykorzystaniem współrzędnych biegunowych
- e) rozwiązuje złożone problemy numeryczne
- f) zna i stosuje standardowe algorytmy (np. dotyczących dynamicznych struktur danych)
- g) tworzy tabele, złożone kwerendy i relacje w bazach danych, stosuje różnorodne narzędzia baz danych, tworzy makrodefinicje
- h) proponuje i tworzy struktury baz danych z użyciem SQL, modyfikuje dane
- i) realizuje zadania znacząco wykraczające poza tematykę lekcji w oparciu o dostępną literaturę fachową

## Plan wynikowy przedmiotu Informatyka poziom rozszerzony Informatyczna Klasa Akademicka

### INFORMATYKA

Rozszerzenie w klasach IKA, tematy pogrupowane działami

Dział	Tematy lekcji	Osiągnięcia ucznia – podstawowe (uczeń)	Osiągnięcia ucznia - rozszerzone (uczeń ponadto)
Lekcje organizacyjne	1. Regulamin pracowni. Przepisy BHP. Ocenianie, WSO, PSO. Sprawy organizacyjne.	zna regulamin i przepisy BHP; zna program nauczania;	
	Informatyka jako przedmiot maturalny. Omówienie podstawy programowej, wymagań maturalnych i kwestii organizacyjnych.	rozumie kwestię wybierania przedmiotu informatyka na maturze;	
Programowanie strukturalne. Metody reprezentowania informacji.	2. Strukturalizacja programu – cd.	zna pojęcia funkcji (składnia, przekazywanie parametrów, typ referencyjny, procedury i funkcje bezparametrowe, parametrowe, zmienne globalne i lokalne, zasięg zmiennych); zna i rozumie algorytm szybkiego potęgowania, schematu Hornera i Wieży Hanoi; rozumie pojęcie rekurencji; deklaruje i stosuje typ łańcuchowy (kopiowanie łańcuchów, łączenie łańcuchów, przypisywanie zmiennym łańcuchowym wartości, proste algorytmy sprawdzające własności napisów, proste algorytmy liczbowe, określa długość łańcucha, używa funkcji określającej pozycję znaku w łańcuchu, zna algorytm tworzenia statystyki tekstu); wykorzystuje typ strukturalny i plikowy do obsługi prostych baz danych, zna różne systemy zapisu liczb, potrafi przeliczać liczby między systemami, pojęcie biblioteki STL, zawartość	rozumie i wykorzystuje przesłanianie zmiennych i uniwersalność procedur i funkcji; potrafi wskazać wywołania stosu rekurencyjnego; płynnie wykorzystuje rozbudowane operacje na łańcuchach (zna sposoby odwoływania się do poszczególnych liter, zamiany liter małych na duże bez użycia funkcji <i>toupper</i> , różne algorytmy sprawdzające własności napisów, algorytmy liczbowe, algorytmy rekurencyjne, potrafi zabezpieczyć program przed wprowadzeniem wartości nie będącej liczbą; zamienia zmienną łańcuchową na liczbową i odwrotnie; tworzy rozbudowany projekt obsługi baz danych typu magazyn, biblioteka, szkoła itp. z wykorzystaniem obsługi plików, podstawowe polecenia STL (wektor)
	3. Ćw. - funkcje		
	4. Zmienne łańcuchowe.		
	5. Podsumowanie (funkcje, układy liczbowe,)		
	6. Ćw.- podst. działania na zm. typu string.		
	7. Zmienne łańcuchowe - cd		
	8. Ćw. – palindromy, anagramy		
	9. Systemy liczbowe		
	10. Struktury danych w c++		
	11. Ćw. – def. i obsługa struktur.		
	12. Obsługa plików tekstowych w c++.		
	13. Ćw. – podstawowe operacje na plikach.		
	14. Ćw. – podstawowe operacje na plikach.		
	15. Ćw. – podstawowe operacje na plikach.		
	16. Ćw. – podstawowe operacje na plikach.		
	17. Schemat Hornera. Złożoność schematu.		
	18. Ćw. - przekazywanie do i z typ referencyjny		
	19. Szybkie potęgowanie		
	20. Ćw. – „potęga od prawej do lewej”		
	21. Ćw. – „potęga od lewej do prawej”		
	22. Podstawy STL - wektor		
	23. Rekurencja, iteracja.		
	24. Rekurencyjny algorytm Hornera.		
	25. Ruchy Browna, ciąg Fibonacciego.		
	26. Wieże Hanoi – algorytm rekurencyjny		
	27. Rekurencja w zadaniach maturalnych.		

Zarządzanie pamięcią	28. Typ wskaźnikowy	zna pojęcie adresu komórek pamięci i typu wskaźnikowego; potrafi wykonywać proste operacje na zmiennych z wykorzystaniem odwołania się poprzez adres; zna określenia wskaźnik do tablicy i wskaźnik do struktur; potrafi odwołać się do pól struktury i je modyfikować; zna podstawowe polecenia zarządzania pamięcią (new, delete)	sprawnie operuje na zmiennych statycznych i na zmiennych dynamicznych; rozumie istotę dynamicznego zarządzania pamięcią i jego konsekwencje; wykorzystuje wskaźniki w implementowaniu algorytmów;
	29. Ćw. – operacje na zm. wskaźnikowych		
	30. Wskaźniki do tablic i struktur		
	31. Ćw. – wskaźniki do tablic i struktur		
	32. Dynamiczne zarządzanie pamięcią		
	33. New, delete – ćw.		
	34. New, delete – ćw.		
	35. Podsumowanie wskaźników		
Porządkowanie i poszukiwanie informacji w zbiorze.	36. Porządkowanie danych – wprowadzenie.	zna zamianę wartości dwóch zmiennych, zna działanie algorytmów: bubblesort, selectionsort, mergesort, quicksort; rozumie pojęcie „in situ”, zna klasę złożoności algorytmu, zna scalanie ciągów uporządkowanych, zna zasadę „dziel i zwyciężaj”, sortuje algorytmem losowe ciągi liczb; opisuje algorytm scalania liczb uporządkowanych, generuje losowe ciągi liczb uporządkowanych; zna pojęcie rekurencji i iteracji, opisuje algorytmy: NWD – naiwny, Euklides – iteracyjny; silnia – definicja rekurencyjna i iteracyjna, zna algorytm wykonywania działań na ułamkach; przedstawia i programuje jedną z metod Euklidesa, oraz naiwny, wykonuje działania na ułamkach oraz program realizujący te działania; zna zamianę wartości dwóch zmiennych, zna klasę złożoności algorytmu, zna scalanie ciągów uporządkowanych, zna zasadę „dziel i zwyciężaj”, sortuje algorytmem losowe ciągi liczb; opisuje algorytm scalania liczb uporządkowanych, generuje losowe ciągi liczb uporządkowanych; zna pojęcie rekurencji i iteracji, opisuje algorytmy: NWD – naiwny, Euklides – iteracyjny; silnia – definicja rekurencyjna i iteracyjna, zna algorytm wykonywania działań na ułamkach; przedstawia i programuje jedną z metod Euklidesa, oraz naiwny, wykonuje działania na ułamkach oraz program realizujący te działania	zna modyfikacje algorytmów sortowania; dokładnie określa ich złożoność obliczeniową i pamięciową; dokonuje pomiaru czasu działania; zna mechanizm sortowania przez scalanie, rozumie zagadnienie efektywności algorytmu, problem sortowania leksykograficznego, zna inne algorytmy sortowania: przez wstawianie, kubełkowe, przedstawia w kodzie źródłowym zmodyfikowane wersje algorytmów sortowania; przedstawia mechanizm działania poznanych algorytmów na przykładzie liczbowym; rozumie algorytm Euklidesa – rekurencyjne, określa cechy iteracji i rekurencji i określa różnice między nimi rozumie problemy z rekurencją, określa różnicę w złożoności obliczeniowej NWD met. naiwną i Euklidesa; programuje algorytmy NWD Euklidesa iteracyjnie i rekurencyjnie; rozumie algorytm przelewania wody i potrafi wykorzystać go w rozwiązywaniu zadań ,zna modyfikacje algorytmów sortowania; przedstawia w kodzie źródłowym zmodyfikowane wersje algorytmów sortowania; przedstawia mechanizm działania poznanych algorytmów na przykładzie liczbowym;
	37. Ćw. – SelectSort.		
	38. Scalanie ciągów uporządkowanych.		
	39. Ćw. – MergeSort.		
	40. Sortowanie „szybkie” QuickSort.		
	41. Ćw. – QuickSort.		
	42. Sortowanie przez zliczanie.		
	43. Ćw. – CountingSort.		
	44. BucketSort - sortowanie kubełkowe dla elem. całkowitych.		
	45. Przeszukiwanie zbioru danych.		
	46. Ćw. – liniowe, binarne.		
	47. InsertionSort - sortowanie przez wstawianie liniowe lub wstawianie binarne		
	48. InsertionSort - sortowanie przez wstawianie liniowe lub wstawianie binarne		
	49. Sortowanie pozycyjne.		
	50. Sortowania – podsumowanie.		
	51. STL – sortowanie, sortowanie z przeciążeniem.		
	52. Przeszukiwanie zbioru danych.		
	53. Ćw. – liniowe, binarne.		
	54. Centralność danych – wstęp.		
	55. Wyszukiwanie mody zbioru.		
56. Wyszukiwanie lidera.			
57. Wyszukiwanie k-tego największego elementu.			
58. Wyszukiwanie idola			

	59. Podsumowanie porządkowania		
	60. NWD – naiwnie i z modyfikacjami		
	61. NWD – algorytm Euklidesa (z różnicą).		
	62. Algorytm przelewania wody		
	63. Ćw. – „rozszerzony Euklides”		
	64. Liczby pierwsze – algorytm naiwny		
	65. Liczby pierwsze – sito Eratostenesa.		
	66. Sortowanie pozycyjne liczb, dat.		
	67. Sortowanie leksykograficzne.		
Przetwarzanie informacji (szyfrowanie, kompresja).	68. Transliteracja - wprowadzenie	zna podstawowe pojęcia związane z liczbami pierwszymi; opisuje algorytm naiwny i sita Eratostenesa do generowania liczb pierwszych; zna pojęcie szyfrowania symetrycznego i asymetrycznego; opisuje metody szyfrowania Cezara, ROT13, Vigenere, Playfair’a, RSA; potrafi sklasyfikować metody szyfrowania pod względem bezpieczeństwa; opisuje podstawowe cechy podpisu elektronicznego; zna jego ideę i potrafi wykorzystać go w praktyce; określa sytuacje z życia codziennego z wykorzystaniem szyfrowania i podpisu; potrafi napisać programy szyfrujące / deszyfrujące z wykorzystaniem poznanych metod szyfrowania; zna zagadnienie kompresji danych; rozróżnia kompresje stratną i bezstratną; opisuje podstawowe algorytmy stosowane przy kompresji ; potrafi napisać kod kompresji naiwnej; zna zasady tworzenia kodu Hoffmana; zna pojęcie prefiksowości kodu	potrafi analizować złożoność obliczeniową algorytmów generowania liczb pierwszych; wykorzystuje sito Eratostenesa w rozwiązywaniu zadań; implementuje poznane algorytmy szyfrowania z wykorzystaniem klucza napisowego, zarodka liczb losowych, różnicy symetrycznej; rozumie wpływ długości klucza szyfrowania na poziom jego bezpieczeństwa; sprawnie obsługuje tablice wielowymiarowe; rozumie algorytm RSA , zna jego matematyczne podstawy i potrafi to wytłumaczyć w kontekście podpisu elektronicznego; potrafi napisać program generujący klucze RSA; szyfrujący / deszyfrujący metodą RSA; potrafi zanalizować podstawowe algorytmy kompresji bezstratnej; określa stopień kompresji danych; wykorzystuje kod Huffmana do tworzenia kodów prefiksowych
	69. Ćw. – szyfr Cezara, ROT13		
	70. Szyfrowanie – cd.		
	71. Ćw. – liczby losowe, różnica symetryczna		
	72. Szyfrowanie – cd.		
	73. Ćw. – szyfr Vigenere		
	74. Tablice dwuwymiarowe, n-wymiarowe		
	75. Ćw. – szyfr Playfair’a		
	76. Elementy kryptografii współczesnej		
	77. Ćw. – algorytm RSA		
	78. Podpis elektroniczny		
	79. Ćw. – podpis elektroniczny w praktyce		
80. Elektroniczna Platforma Usług Administracji Publicznej			
81. Kompresja danych			
82. Ćw. – algorytmy kompresji, kod Huffmana			
Programowanie zachłanne i dynamiczne	83. Programowanie zachłanne	rozumie „technikę zachłanną”; zna problem wydawania reszty i na tej podstawie potrafi określać optymalność i dopuszczalność rozwiązań; potrafi omówić zachłanne i dynamiczne metody rozwiązywania problemów plecakowych i najkrótszej drogi; zna ich złożoność obliczeniową i pamięciową, potrafi korzystać z STL, zna gotowe komponenty	potrafi zaimplementować algorytmy zachłanne i dynamiczne problemów plecakowych; projektuje odpowiednie struktury danych; rozumie problem „trudności” rozwiązań dynamicznych przy dużych zestawach danych;
	84. Wydawanie reszty		
	85. Pakowanie plecaka ogólnego, decyzyjnego.		
	86. Pakowanie plecaka decyzyjnego zachłanne.		
	87. Porządkowanie ułamków; przeciążanie <		
	88. Plecak ogólny dynamicznie		
	89. Plecak ogólny – implementacja.		
	90. Plecak decyzyjny dynamicznie		
	91. Plecak decyzyjny dynamicznie.		
	92. Złożoność problemów plecakowych.		
	93. Najkrótsza droga – algorytm dynamiczny		
	94. Najkrótsza droga – algorytm dynamiczny		

	95. Podsumowanie pr. dynamicznego i zachłannego.		
Grafy	96. STL – kolejka, stos, para, zbiór, map.	zna podstawowe pojęcia: graf, wierzchołek, krawędź, graf ważony, graf skierowany i zna ich reprezentacje; tworzy grafy losowe; zna podstawowe metody przeszukiwania grafów; potrafi korzystać z biblioteki STL w rozwiązywaniu problemów wykorzystujących stos, wektor; rozumie problem notacji np. ONP i potrafi zaimplementować ją przy pomocy stosu	potrafi korzystać z biblioteki STL w rozwiązywaniu problemów wykorzystujących kolejkę; potrafi zaimplementować podstawowe metody przeszukiwań grafów i potrafi je wykorzystać w rozwiązywaniu zadań; zna algorytm Dijkstry i potrafi go zaimplementować; zna algorytm stacji rozrządowej Dijkstry; zna i rozumie problem cykli w grafach; rozumie problem TSP (komiwojażera); zna istotę programowania z nawrotami;
	97. STL – kolejka, stos, para, zbiór, map.		
	98. Grafy – podstawowe pojęcia		
	99. Macierz sąsiedztwa		
	100. ONP – własna implementacja stosu		
	101. ONP – wykorzystanie stosu z STL.		
	102. Grafy – cd.		
	103. Lista sąsiedztwa		
	104. Najkrótsze ścieżki w grafie		
	105. Algorytm Dijkstry.		
	106. Przeszukiwanie grafów – DFS.		
	107. DFS – implementacja.		
108. Przeszukiwanie grafów – BFS.			
109. BFS – implementacja.			
110. Algorytmy grafowe – zadania.			
111. Algorytmy grafowe – zadania.			
Obliczenia numeryczne	112. Obliczenia numeryczne – wstęp. Błędy obliczeń.	zna konsekwencje reprezentacji danych w komputerze; potrafi porównywać dane niecałkowite; zna pojęcie szacunku obliczeń; rozumie problem dokładności obliczeń i powstawania niestabilności algorytmów; potrafi przedstawić algorytm „szkolny” i stabilny równania kwadratowego; zna algorytm Herona ; wykorzystuje schemat Hornera do obliczenia wartości wielomianu n-tego stopnia; zna metodę bisekcji na znalezienie przybliżeń miejsc zerowych; zna pojęcie kwadratur (całek numerycznych) i potrafi to wykorzystać przy obliczaniu pola pod krzywą;	znajduje w algorytmach miejsca z potencjalną niestabilnością obliczeniową; potrafi usuwać niestabilność numeryczną; rozumie istotę iteracyjnych metod rozwiązywania problemu z narzuconą dokładnością i potrafi to wykorzystać do obliczeń na podstawie wzorów, w których występują nieskończona liczba elementów np. obliczanie liczby Pi;
	113. Niestabilność numeryczna		
	114. Równanie kwadratowe metoda „szkolna”		
	115. Równanie kwadratowe metoda stabilna.		
	116. Pierwiastki kwadratowe z liczby.		
	117. Metoda Herona.		
	118. Wielomian n-tego stopnia.		
	119. Miejsca zerowe wielomianu. Bisekcja.		
	120. Całkowanie numeryczne.		
	121. Kwadratura prostokątów.		
	122. Kwadratura trapezów		
	123. Podsumowanie obliczeń numerycznych.		
Geometria obliczeniowa	124. Geometria obliczeniowa – wstęp.	zna podstawowe problemy reprezentacji danych w komputerze; zna pojęcie macierzy, potrafi obliczyć wyznacznik 2 i 3-ego stopnia; zna podstawowe algorytmy geometryczne: po której stronie leży punkt; czy 2 punkty leżą po tej samej stronie; czy punkt należy do odcinka; czy 2 odcinki się przecinają; zna twierdzenie o trójkącie kratowym; umie obliczyć pole trójkąta i na tej podstawie sprawdzić przynależność punktu do wielokąta wypukłego; zna pojęcie iloczynu	reprezentuje macierz w sposób statyczny lub dynamiczny w zależności od potrzeb; zna pojęcie iloczynu skalarnego; zna algorytmy wyznaczające otoczkę wypukłą (Grahama, Jarvisa) i potrafi to wykorzystać w zadaniach; potrafi wykorzystać biblioteki graficzne do wizualizacji fraktali; zna określenia zbiorów Julii, Mandelbrota
	125. Wyznaczniki macierzy.		
	126. Geometria – podstawowe algorytmy.		
	127. Geometria – podstawowe algorytmy.		
	128. Algorytmy geometryczne – trójkąt kratowy.		
	129. Algorytmy geometryczne – zad. wielokąt		
	130. Iloczyn wektorowy.		
	131. Pole równoległoboku, trójkąta, wielokąta.		

	132. Otoczka wypukła.	wektorowego i na jego podstawie potrafi wyznaczyć pole równoległoboku, trójkąta i wielokąta nie wypukłego; zna pojęcie fraktal; potrafi zdefiniować rekurencyjne podstawowe fraktale	
	133. Otoczka – algorytm Jarvisa.		
	134. Podstawy grafiki – allegro.		
	135. Podstawy grafiki – allegro.		
	136. Fraktale – wstęp.		
	137. Płatek Kocha		
	138. Drzewo binarne.		
	139. Dywan Sierpińskiego.		
140. Podsumowanie geometrii obliczeniowej.			
Przeszukiwanie tekstu	141. Przetwarzanie tekstów – podstawowe pojęcia.	zna podstawowe pojęcia: alfabet, łańcuch, wzorzec, prefiks, sufiks, prefikso-sufiks; potrafi wyszukiwać wzorzec naiwnie i z wartownikami; rozumie ideę wartowników; zna podstawowe operacje na łańcuchach; rozumie znaczenie algorytmów wyszukiwania wzorca w tekście; rozumie ideę algorytmu Morrisa_Pratta oraz Boyera-Moore’a.	potrafi wyznaczyć tablicę MPNext i zaimplementować algorytm MP; wie jak wykorzystać tablicę Last w algorytmie Boyera-Moore’a;
	142. Wyszukiwanie wzorca – naiwnie.		
	143. Wyszukiwanie wzorca naiwnie z wartownikami.		
	144. Przetwarzanie tekstu – tablica MPNext.		
	145. Wyszukiwanie wzorca – alg. Morrisa_Pratta.		
	146. Algorytm Boyera –Moore’a.		
	147. Algorytm Boyera –Moore’a (tablica Last).		
	148. Podsumowanie przetwarzania tekstów.		
Arkusz kalkulacyjny	149. Arkusz kalkulacyjny – sprawdzenie wiadomości i umiejętności	zna pojęcie arkusza, komórki, zakresu komórek, zna możliwe działania wykonywane na komórkach, pojęcie funkcji i formuły, typy adresowań, typy danych wprowadzanych do komórki arkusza, serie danych, rodzaje operatorów, elementy wykresu; zna poprawność obliczeniową i dokładność arkusza, funkcję warunkową; zna warunki utworzenia i zastosowanie sum częściowych (pośrednich) i tabel przestawnych; zna funkcje przeszukiwania baz danych licz.jeżeli, wyszukaj.pionowo, suma.jeżeli; zna możliwości zastosowania solvera; zna pojęcie i sposoby tworzenia i wykorzystanie makrodefinicji; wykonuje obliczenia, stosuje podstawowe funkcje w formułach, formatuje komórki, kopiuje zawartość komórek, tworzy wykresy w oparciu o serie danych, formatuje wykresy, blokuje okienka, obsługuje bazę poprzez formularz; wyszukuje, sortuje i filtruje dane, stosuje autofiltr; stosuje bazy do korespondencji seryjnej; tworzy arkusze symulujące prosty eksperyment; używa poprawnie pomocy,	stosuje filtr zaawansowany, nadaje nazwy zakresom komórek, stosuje funkcje częstość, licz.jeżeli; stosuje konspektowanie, sumy złożone, tworzy układ tabel przestawnych i wykresów przestawnych.; potrafi przeprowadzić hipotetyczny eksperyment z wykorzystaniem arkusza kalkulacyjnego; zna parametry konfiguracyjne solvera, analizuje zaawansowane dane w oparciu o solver; edytuje makrodefinicje (w VBA tworzy zaawansowane formularze); potrafi wykorzystać arkusz do implementowania algorytmów dyskretnych i numerycznych
	150. Arkusz – geometryczna interpretacja danych.		
	151. Wykresy funkcji trygonometrycznych		
	152. Symulacja zjawisk. Krzywe Lissajoux.		
	153. Symulacja zjawisk.		
	154. Symulacja eksperymentów.		
	155. Symulacja eksperymentów		
	156. Bazy danych w arkuszu.		
	157. Sumy pośrednie, tabele przestawne, korespondencja seryjna.		
	158. Automatyzacja – makrodefinicje.		
	159. Makra – formanty		
	160. Optymalizacja zadań – Solver.		
	161. Rozwiązywanie zadań maturalnych.		
	162. Rozwiązywanie zadań maturalnych.		
163. Rozwiązywanie zadań maturalnych.			
164. Rozwiązywanie zadań maturalnych.			
165. Rozwiązywanie zadań maturalnych.			
166. Rozwiązywanie zadań maturalnych.			

	167. Rozwiązywanie zadań maturalnych.		
	168. Podsumowanie arkusza kalkulacyjnego		
Komputerowe środowisko pracy.	169. Elektronika cyfrowa; bramki logiczne	zna podst. pojęcia informatyki, podst. budowę komp., sposoby i rodz. kodowania danych, systemy liczbowe: binarny oktalny, heksadecymalny; przelicza liczby między systemami; charakteryzuje ideową budowę komputera, potrafi wyjaśnić zakres zastosowania technologii informacyjnej zna podstawowe bloki funkcjonalne komputera, budowę płyty głównej, podstawowe urządzenia I/O (klawiatura, urządzenie wskazujące, skaner, tablet, kontrolery gry, karta sieciowa, dźwiękowa, modem, napędy dysków, wyświetlacz z kartą graficzną, drukarka, ploter), podział pamięci rozumie ideę „Open Source”; zna określenia GNU GPL; wykorzystuje podstawowe aplikacje środowiska Ubuntu;	zna generacje komputerów, omawia różnice pomiędzy poszczególnymi generacjami, zna budowę diody, lampy elektronowej, tranzystora npn i pnp, potrafi ocenić obecny poziom technologii informacyjnej i perspektywy jej rozwoju; zna parametry charakteryzujące w/w podzespoły, zna funkcje i bloki funkcjonalne procesora, charakteryzuje poszczególne bloki płyty głównej; potrafi zainstalować środowisko Ubuntu i skonfigurować je do własnych potrzeb;
	170. Układy z pamięcią - przerzutniki		
	171. Budowa komputera - podzespoły		
	172. Systemy operacyjne – Windows (przypomnienie)		
	173. Ubuntu – środowisko Linux; możliwości		
	174. Ubuntu – obsługa systemu Linux		
Sieci komputerowe	175. Sieci komputerowe - wprowadzenie	zna składniki i rodzaje sieci, pojęcie protokołu, charakteryzuje protokół TCP/IP, sposoby łączenia komputerów w sieć; zna topologie sieci, funkcje serwera, pojęcie woluminu, podstawowe polecenia serwera, mapowanie charakteryzuje budowę Internetu, zadania protokołów, zna możliwości połączenia komputera z siecią, pojęcie i właściwości hipertekst charakteryzuje WWW, potrafi rozpoznać podstawowe człony nazwy domenowej, budowę URL, podstawowe możliwości przeglądarki, adresy wyszukiwarek i katalogów stron, zna pojęcie hipertekstu i jego właściwości, operatory stosowane przy wyszukiwaniu; charakteryzuje e-mail, listserv, usenet news, zna możliwości programów poczty, problematykę	charakteryzuje sieci, zna sposób działania urz. (hub, switch, router, bridge, gateway), zna charakterystykę zabezpieczeń zasobów w sieci, składnię UNC; zna charakterystykę topologii, model ISO/OSI, technologie sieciowe sieci WAN, pojęcie grup, praw grupowych, filtrów IRF, funkcje bezpieczeństwa danych w sieciach; zna historię Internetu, potrafi wyznaczyć podsieci danej sieci, ilość hostów podsieci, zadania TCP, pojęcie enkapsulacji i komutacji pakietów, charakteryzuje protokoły, maski sieci, zna DNS, konfigurację przeglądarki, zabezpieczenia prywatnych danych i przeglądarek, przeglądanie stron w trybie off-line, pojęcie, cechy i zadania portali, mechanizm działania wyszukiwarek, multiwyszukiwarek, katalogów stron, cookie, cache, zna
	176. Sieci – model OSI, protokoły sieciowe		
	177. Sieci – topologie, urządzenia, kable		
	178. Sieci – topologie, urządzenia, kable – ćw.		
	179. Sieci – elementy łączące		
	180. Internet – podst. pojęcia; protokół TCP/IP		
	181. Internet – komutacja, adresacja IP, DNS		
	182. Internet – rodz. dostępu		
	183. Internet – usługi ( www, poczta)		
	184. Internet – http, pop3, smtp, telnet, ftp,		
	185. Internet – rodzaje zagrożeń i bezpieczeństwo		
	186. Internet – netykieta		
	187. Sieci - podsumowanie		



		wirusową, pola odbiorców, konfigurację konta, adresy serwerów, moderowanie, zna usługi IRC, chat, Instant Messenger, VoIP, FTP, P2P, telnet, ssh, ping, prawne aspekty użytkownika sieci P2P, podaje najważniejsze cechy usług, zastosowanie i możliwości komunikatorów, bramki SMS; wykorzystuje narzędzia do komunikacji w sieci(np. NetMeeting), określa IP komputera, wyznacza IP sieci, broadcast, udostępnia i zabezpiecza zasoby, loguje się do serwera, wysyła komunikaty sieciowe, przelogowuje się bez przerwania pracy, korzysta z zasobów serwera;obsługuje przeglądarkę, obsługuje mechanizm zakładek/ulubionych, wyszukuje wiadomości w Internecie, pobiera dane z Internetu, wysyła wiadomości pocztowe, konfiguruje tożsamości i skrzynki emailowe, używa książki adresowej, instaluje oprogramowanie komunikatora, porozumiewa się komunikatorem i programem VoIP, użytkuje klienta FTP	netykietę ; stosowanie PGP, usługi i konfiguracja programu pocztowego, przyporządkowuje usługi do protokołów pop3, imap, nntp, smtp, standard MIME,;zna klientów IRC, FTP, telnet, serwery, klientów, klientów danych strumieniowych, zastosowania telnet, ssh, konfiguruje połączenie sieci, odzyskuje pliki z kosza sieciowego, proponuje adresy witryn firm, konfiguruje przeglądarkę, tworzy obrazy off-line witryn zabezpiecza przesyłki, używa grupy dyskusyjne, reguły wiadomości, wizytówki, podpisy, żądania potwierdzenia, priorytety, szyfruje wiadomości, pobiera klucz; sprawdza adres IP serwera, sprawdza trasę pakietu w sieci
Relacyjne bazy danych	188. Wstęp do baz danych	zna podstawowe problemy organizacji baz danych, potrafi projektować tabele tworzące relacyjny model baz danych, rozumie pojęcie kluczy; zna funkcje najważniejszych obiektów relacyjnych b.d., potrafi projektować zaawansowane formularze, kwerendy i raporty, potrafi tworzyć zapytania (kwerendy)	potrafi tworzyć formularze stosując słowniki; zna pojęcie kreatora odnośników; projektuje relacyjną bazę danych składającą się z wielu tabel połączonych relacją (na zadany temat); swobodnie projektuje formularz i raport ; stosuje zasady definiowania kluczy podstawowych; łączy informacje z bazy danych z dokumentami innych programów, np. edytora tekstu czy arkusza kalkulacyjnego; stosuje różne rodzaje kwerend; samodzielnie ustala zawartość bazy (rodzaj informacji); samodzielnie projektuje wygląd formularza i raportu; sprawnie korzysta z dodatkowej, fachowej literatury;
	189. Środowisko systemu zarządzania baz danych.		
	190. Tworzenie struktury b.d.		
	191. Tabele – obiekt b.d.		
	192. Anomalie i redundancje.		
	193. Normalizacja baz danych.		
	194. Relacje w b.d.(rodzaje, tworzenie, klucze).		
	195. Wyszukiwanie inf. w b.d.		
	196. Kwerendy – rodz., tworzenie, zastosowanie, SQL		
	197. Obsługa baz danych – formularze.		
	198. Obiekt: formularze - ćwiczenia		
	199. Bazy danych – rozwiązywanie zadań maturalnych.		
	200. Bazy danych – rozwiązywanie zadań maturalnych.		
201. Bazy danych – rozwiązywanie zadań maturalnych.			
202. Bazy danych – rozwiązywanie zadań maturalnych.			
Standard Query Language	203. Podstawy języka SQL	zna w podstawowym zakresie składnię języka SQL, zna podstawowe polecenia SQL: SELECT FROM WHERE; stosuje operatory i kryteria, sortowanie,	potrafi tworzyć zapytania SQL i wykorzystując złożone polecenia SQL; obliczenia w SQL, zna i potrafi użyć środowisko serwera SQL;
	204. Podstawy SQL – ćwiczenia.		
	205. Funkcje agregujące.		
	206. Relacje.		
	207. Tworzenie tabel, struktur tabel.		
	208. Wypełnianie, aktualizacja i usuwanie danych SQL.		

Grafika komputerowa	209. Grafika komputerowa - wprowadzenie	zna pojęcia grafika komp., modele barw RGB, CMYK, grafika rastrowa, wektorowa; potrafi świadomie stosować formaty plików graficznych; obsługuje aparat cyfrowy i skaner; wykonuje podst. operacje na obiektach i wbudowanych figurach wektorowych (przesuwanie, skalowanie, ściężnianie, obracanie, ścinanie, tworzenie kopii), stosuje tekst akapitowy i ozdobny	sprawnie posługuje się edytorem graficznym przy obróbce obrazu; świadomie wykorzystuje różnego rodzaju efekty przekształcania obrazu; wykorzystuje warstwy i maski; potrafi obliczyć wielkość pliku przy określonej rozdzielczości i głębi koloru; stosuje wypełnienia, przezroczystości i cieniowanie interaktywne
	210. CorelPhotoPaint / GIMP – edycja grafiki rastrowej.		
	211. CorelPhotoPaint / GIMP – edycja grafiki rastrowej.		
	212. Edytor grafiki wektorowej – Corel Draw		
	213. Grafika wektorowa – operacje na obiektach		
Strony www	214. Serwis www – ramki, tabele, inne elementy	zna mechanizm znaczników HTML; umie wygenerować kod HTML i go modyfikować; potrafi optymalizować dobór grafiki; potrafi przekazać sterowanie do kolejnego dokumentu, generuje tabele i ramki; zna podstawy PHP; rozumie funkcjonowanie języka zapytań SQL i bazy MySQL; rozumie konieczność funkcjonowania mechanizmu baz danych na serwerach internetowych;	potrafi samodzielnie uczyć się praktycznego wykorzystania znaczników i ich atrybutów, zna zasady wykorzystania stylów, potrafi dowolnie pozycjonować tekst i grafikę, potrafi zbudować serwis www za pośrednictwem dowolnej techniki opartej na wykorzystaniu „wtyczki”; tworzy strony z wykorzystaniem instrukcji PHP i SQL; potrafi samodzielnie obsługiwać formularze i pliki; zarządzać bazami, tabelami, rekordami
	215. Formularze na stronach www		
	216. Java Script - wprowadzenie; przykłady skryptów		
	217. Instalacja serwera WAMP (WWW, MySQL, PHP)		
	218. Pierwsze skrypty w języku PHP		
	219. Budowa skryptów PHP (współpracy z b. danych)		
	220. Tworzenie kwerend.		

Pozostałe godziny przeznaczone na powtórzenia i sprawdziany.

## **Wymagania edukacyjne niezbędne do uzyskania poszczególnych ocen klasyfikacyjnych**

### **Ocena dopuszczająca:**

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) posługiwać się użytkowymi programami biurowymi w zakresie podstawowym
- b) wyszukiwać informacje w Internecie
- c) podstawy składni języka programowania C++
- d) pisać proste programy na wzór przykładów prezentowanych podczas lekcji
- e) tworzyć strony WWW przy użyciu języka HTML w zakresie podstawowym
- f) wykorzystywać arkusz kalkulacyjny do badania podstawowych cech funkcji matematycznych
- g) rozwiązywać proste problemy numeryczne na wzór przykładów realizowanych na zajęciach
- h) tworzyć tabele i proste kwerendy w bazach danych według przedstawionych przykładów
- i) pisać podstawowe polecenie SQL
- j) podstawowe metody przenoszenia plików za pomocą sieci
- k) tworzyć proste strony WWW z wykorzystaniem języka PHP

### **Ocena dostateczna:**

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) biegle posługiwać się użytkowymi programami biurowym
- b) wyszukiwać informacje w Internecie z wykorzystaniem zaawansowanych technik
- c) składnię języka programowania C++
- d) pisać programy rozwiązujące proste problemy liczbowe i tekstowe
- e) tworzyć strony WWW przy użyciu języka HTML, podstaw stylów kaskadowych CSS i języka PHP
- f) sprawnie wykorzystuje arkusz kalkulacyjny do badania cech funkcji matematycznych
- g) rozwiązuje problemy numeryczne na podstawie wzorów przedstawionych na zajęciach
- h) tworzy tabele, kwerendy i proste relacje w bazach danych według przedstawionych przykładów
- i) tworzy podstawowe polecenia SQL z użyciem dodatkowych klauzul

### **Ocena dobra:**

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) biegle posługuje się użytkowymi programami biurowym
- b) wyszukuje informacje w Internecie z wykorzystaniem zaawansowanych technik
- c) dobra znajomość składni języka programowania C++
- d) pisze programy rozwiązujące rozmaite problemy liczbowe i tekstowe z wykorzystaniem technik programowania proceduralnego
- e) rozwiązuje złożone problemy numeryczne
- f) zna i stosuje standardowe algorytmy
- g) tworzy złożone strony WWW w systemie Linux z wykorzystaniem zaawansowanych stylów kaskadowych CSS oraz skryptów w języku Javascript
- h) tworzy kwerendy zaawansowane w oparciu o złożone bazy danych
- i) tworzy proste polecenia SQL w oparciu o kilka tabel

## Ocena bardzo dobra:

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) biegle posługiwać się użytkowymi programami biurowym
- b) wyszukiwać informacje w Internecie z wykorzystaniem zaawansowanych technik
- c) bardzo dobrze składnię języka programowania C++
- d) pisze programy rozwiązujące złożone problemy liczbowe i tekstowe z wykorzystaniem techniki programowania proceduralnego, zna podstawy programowania obiektowego
- e) tworzenie złożonych stron WWW przy użyciu języka HTML i zaawansowanych stylów kaskadowych CSS
- f) biegle wykorzystuje arkusz kalkulacyjny do rozwiązywania różnych zadań matematycznych i z życia codziennego
- g) rozwiązuje złożone problemy numeryczne
- h) zna i stosuje standardowe algorytmy (np. dotyczących dynamicznych struktur danych, geometrycznych i podstawowe algorytmy grafowe)
- i) tworzy tabele, złożone kwerendy i relacje w bazach danych według przedstawionych projektów
- j) tworzy złożone polecenia SQL w oparciu o kilka tabel

## Ocena celująca:

Uczeń opanował wiadomości i umiejętności przewidziane podstawą programową w takim zakresie, że potrafi/zna:

- a) zna dodatkowe biblioteki, sprawnie proponuje różne funkcje obsługujące to samo zadanie, potrafi omówić różnice C++
- b) pisze programy rozwiązujące złożone problemy liczbowe i tekstowe z wykorzystaniem techniki programowania proceduralnego i obiektowego
- c) tworzenie złożonych stron WWW przy użyciu różnych technik tworzenia stron
- d) biegle wykorzystuje arkusz kalkulacyjny do badania rozmaitych cech funkcji matematycznych również z wykorzystaniem współrzędnych biegunowych
- e) rozwiązuje złożone problemy numeryczne
- f) zna i stosuje standardowe algorytmy (np. zaawansowane algorytmy grafowe)
- g) tworzy tabele, złożone kwerendy i relacje w bazach danych, stosuje różnorodne narzędzia baz danych, tworzy makrodefinicje
- h) proponuje i tworzy struktury baz danych z użyciem SQL, modyfikuje dane
- i) realizuje zadania znacząco wykraczające poza tematykę lekcji w oparciu o dostępną literaturę fachową