

Wymagania na ocenę z informatyki (poziom rozszerzony)

Przyjmuje się, że uczeń spełnia wymagania na ocenę wyższą, jeśli spełnia jednocześnie wymagania na ocenę niższą oraz dodatkowe wymagania.

Urządzenia komputerowe w sieci.

Ocenę celującą otrzymuje uczeń, który:

- ✓ obsługuje różne systemy operacyjne,
- ✓ korzysta z poleceń trybu tekstowego Windows,
- ✓ kopiuje pliki w trybie tekstowym Windows za pomocą ścieżek względnych i bezwzględnych,
- ✓ dokonuje istotnych zmian w BIOS,
- ✓ wyjaśnia zasadę działania sztucznego neuronu i sieci neuronowej,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ wymienia i opisuje zastosowania sieci internet,
- ✓ charakteryzuje różne topologie sieci komputerowych,
- ✓ wyjaśnia pojęcie i budowę ramki jako porcji informacji w transmisji danych,
- ✓ opisuje sposób adresowania urządzeń w sieci internet, wyjaśnia sposób komunikacji między urządzeniami tej samej oraz różnych sieci, opisuje sposób tworzenia i budowę domeny internetowej, konfiguruje urządzenie do pracy w internecie i omawia ten proces, wymienia i omawia protokoły usług internetowych, diagnozuje stan połączeń internetowych,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ opisuje, czym jest model warstwowy systemu komputerowego,
- ✓ wymienia i wyjaśnia zadania systemu operacyjnego,
- ✓ określa różnicę pomiędzy trybem jądra a trybem użytkownika,
- ✓ z prostych brył 3D i ich przekształceń tworzy modele 3D,
- ✓ instaluje i aktualizuje oprogramowanie,
- ✓ umiejętnie korzysta z Menedżera zadań w systemie Windows podczas zamykania aplikacji,
- ✓ korzysta z narzędzi oczyszczania dysku,
- ✓ opisuje procedurę wykonywania kopii zapasowej dla systemu operacyjnego w szkolnej pracowni,
- ✓ opisuje zastosowania rzeczywistości wirtualnej i rozszerzonej,
- ✓ podaje cechy różnych rodzajów licencji oprogramowania,
- ✓ stosuje symbole i wyrażenia w wyszukiwarkach internetowych,
- ✓ wymienia i opisuje urządzenia sieciowe,
- ✓ opisuje sieci komputerowe ze względu na zasięg ich działania,
- ✓ wyjaśnia budowę adresów MAC i sprawdza je na komputerze z systemem Windows,
- ✓ wyjaśnia pojęcia: adres IP, maska podsieci,
- ✓ opisuje modele klient-serwer oraz peer-to-peer,
- ✓ określa relacje między podmiotami rynku e-usług,
- ✓ korzysta z wybranych e-usług,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ wymienia urządzenia wchodzące w skład sieci komputerowej,
- ✓ identyfikuje wersję systemu operacyjnego swojego smartfona (komputera),
- ✓ wyjaśnia różnicę pomiędzy bezwzględną i względną ścieżką dostępu,
- ✓ określa różnicę pomiędzy BIOS a UEFI,

- ✓ rozumie pojęcie serwera,
- ✓ opisuje zasady bezpiecznego korzystania z systemu operacyjnego,
- ✓ wyjaśnia, jak założyć konto użytkownika w używanym przez siebie systemie operacyjnym,
- ✓ konstruuje bezpieczne hasła,
- ✓ kopiuje dane celem stworzenia kopii zapasowej na zewnętrznym nośniku,
- ✓ uruchamia komputer w trybie awaryjnym,
- ✓ sprawdza obciążenie procesora,
- ✓ wyjaśnia pojęcia fragmentacji i defragmentacji dysku, wyjaśnia różnicę pomiędzy systemami plików FAT32 oraz NTFS, definiuje pojęcie systemu operacyjnego, wyjaśnia różnicę pomiędzy wirtualną a rozszerzoną rzeczywistością, wyjaśnia pojęcia: prawo autorskie, licencja, rozróżnia i definiuje pojęcia wolnego i otwartego oprogramowania,
- ✓ nazywa różne porty urządzeń sieciowych,
- ✓ rozróżnia typy domen (krajowe, funkcjonalne),
- ✓ wyjaśnia pojęcie systemu DNS,
- ✓ opisuje budowę adresu URL,
- ✓ wyjaśnia, czym są e-usługi,
- ✓ wyjaśnia pojęcie licencji Creative Commons,
- ✓ wymienia wiarygodne źródła informacji w sieci internet,
- ✓ wyjaśnia, jak sprawdzić właściciela serwisu internetowego,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ wymienia urządzenia mobilne zaliczane do systemów komputerowych,
- ✓ wymienia elementy budowy systemu operacyjnego,
- ✓ rozumie pojęcie ścieżka dostępu w kontekście systemów plików,
- ✓ sprawdza i wymienia atrybuty pliku,
- ✓ opisuje, jak uruchomić system BIOS na komputerze,
- ✓ wyjaśnia konieczność tworzenia bezpiecznych haseł,
- ✓ wymienia metody zabezpieczania danych na komputerze,
- ✓ uruchamia Menedżera zadań w systemie Windows,
- ✓ wymienia problemy, jakie można napotkać podczas korzystania z komputera,
- ✓ wyjaśnia pojęcie sztucznej inteligencji,
- ✓ opisuje, czym jest chmura obliczeniowa,
- ✓ wymienia zastosowania automatów i robotów,
- ✓ zna i opisuje zagrożenia wynikające z rozwoju technologii,
- ✓ wyjaśnia pojęcia: sieci komputerowe i urządzenia sieciowe,
- ✓ wyjaśnia pojęcie cyfrowej tożsamości,
- ✓ wymienia sposoby uwierzytelniania użytkowników e-usług,
- ✓ wskazuje miejsca występowania e-zasobów,
- ✓ rozróżnia wyszukiwarki od przeglądarek internetowych,

Sieci komputerowe

Ocenę celującą otrzymuje uczeń, który:

- ✓ zakłada ciekawe wątki na forach dyskusyjnych dotyczące informatyki i odpowiada na pytania forumowiczów,
- ✓ kieruje pracami zespołu przygotowującego i wybierającego chmurę informatyczną do konkretnego projektu,
- ✓ stosuje zaawansowane narzędzia edytorów z chmury informatycznej,
- ✓ korzysta z różnych chmur informatycznych, w których przechowuje, edytuje i współdzieli dokumenty,
- ✓ modyfikuje i sprawnie dobiera parametry wirtualnej maszyny w zależności od potrzeb uruchamianego w niej systemu,
- ✓ korzysta z programów narzędziowych systemu Linux,
- ✓ wykorzystuje programy narzędziowe do operacji na dyskach twardych i ich konserwacji np. defragmentacji, usuwania błędnych wpisów, przywracania systemu itp.
- ✓ dokładnie opisuje sposób transportu informacji w sieciach komputerowych TCP/IP,
- ✓ sprawnie konfiguruje ustawienia routera sieciowego dołączonego do Internetu,
- ✓ tworzy sieć z zastosowaniem przełączników sieciowych,
- ✓ zmienia kanały pracy sieci bezprzewodowej zmienia funkcje Access Pointa na urządzenie odbierające sygnał sieci,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ umie korzystać z kursów elearningowych,
- ✓ wie, jak zorganizować pracę zespołu w sieci,
- ✓ podaje przykłady rozmaitych stron z dokładnym określeniem ich rodzaju i przeznaczenia,
- ✓ organizuje pracę zespołu w chmurze informatycznej np. z wykorzystaniem kalendarza,
- ✓ wykorzystuje chmurowe narzędzia do komunikowania się w zespole w celu wspólnej realizacji projektu,
- ✓ przenosi dokumenty z chmury do lokalnego komputera oraz edytuje dokumenty zaimportowane,
- ✓ zarządza pracą zespołu współdzielącego dokument,
- ✓ korzysta z systemu operacyjnego uruchomionego w wirtualnej maszynie i wie, jakie to tworzy ograniczenia,
- ✓ zna i omawia warstwowy model systemu operacyjnego,
- ✓ posługuje się podstawowymi poleceniami systemowymi, takimi jak ls, mkdir, rmdir, pwd, cd, wykonywanymi za pośrednictwem Terminala systemu Linux,
- ✓ przeprowadza defragmentację dysku komputera za pomocą programu systemowego Defragmentator dysku,
- ✓ umie opisać funkcje ramki i nagłówek i urządzenia sieciowe w modelu TCP/IP,
- ✓ umie określić adres sieci na podstawie maski,
- ✓ posługuje się poleceniem ping do sprawdzenia połączenia sieciowego z komputerem w sieci i określenia adresu fizycznego serwera dowolnej strony www,
- ✓ posługuje się poleceniem ipconfig w celu odczytania pełnej konfiguracji karty sieciowej danego komputera,
- ✓ konfiguruje router i łączy za jego pośrednictwem sieć lokalną z Internetem,
- ✓ wykorzystuje program diagnostyczny np. inSSIDer do analizy ruchu w sieci bezprzewodowej i ustalenia optymalnego kanału,
- ✓ chroni konta przez wielokrotnymi próbami wpisywania haseł,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ wie, czym jest serwer IIS,

- ✓ włącza w systemie internetowe usługi informacyjne,
- ✓ odczytuje dane udostępnione przez innego użytkownika w ramach działania serwera IIS,
- ✓ wie, jak zabezpieczane są pliki PDF z książkami,
- ✓ odbiera transmisje strumieniowe w sieci,
- ✓ uczestniczy w forach dyskusyjnych,
- ✓ umie wymienić właściwości i zastosowanie chmur informatycznych,
- ✓ stawia wymagania chmurze informatycznej,
- ✓ korzysta z jej podstawowych programów ,
- ✓ wykorzystuje programy z chmury informatycznej do redagowania tekstów, rysowania itp.
- ✓ korzysta z dysku w chmurze do przechowywania plików,
- ✓ udostępnia dokumenty członkom zespołu oraz nadaje im odpowiednie uprawnienia do edycji,
- ✓ uruchamia system operacyjny w wirtualnej maszynie,
- ✓ posługuje się monitorem zasobów systemu,
- ✓ instaluje wybrane, potrzebne do pracy programy za pośrednictwem Centrum oprogramowania systemu Linux
- ✓ zna podstawowe cechy systemu Android,
- ✓ przeprowadza selektywną aktualizację systemu, odrzucając mniej znaczące elementy oferowane przez producenta,
- ✓ sprawnie posługuje się programami narzędziowymi, w tym CCleaner, do utrzymania odpowiedniego stanu systemu operacyjnego – kasuje niepotrzebne pliki,
- ✓ naprawia błędy w rejestrach i przywraca system od punktu przywracania,
- ✓ porównuje oba modele sieci informatycznych i opisuje różnice,
- ✓ umie opisać funkcje ramki i nagłówków i urządzenia sieciowe w modelu TCP/IP,
- ✓ używa polecenia tracert i programu diagnostycznego np. VisualRoute Lite Edition do śledzenia drogi połączenia sieciowego z dowolną stroną internetową,
- ✓ posługuje się poleceniem ipconfig w celu odczytania pełnej konfiguracji karty sieciowej danego komputera,
- ✓ zabezpiecza sieć bezprzewodową w dostępnych standardach, w tym WPA i WPA2 z zastosowaniem PSK,
- ✓ posługuje się konsolą MMC systemu Windows, tworząc szablony zabezpieczeń i odpowiednio blokując konta,
- ✓ dodaje przystawki zwiększające możliwości MMC,
- ✓ wie, czym jest serwer IIS,
- ✓ włącza w systemie internetowe usługi informacyjne,
- ✓ odczytuje dane udostępnione przez innego użytkownika w ramach działania serwera IIS,
- ✓ wie, jak zabezpieczane są pliki PDF z książkami,
- ✓ odbiera nagłówki RSS za pomocą email i czytników RSS np. Paseczek,
- ✓ odbiera transmisje strumieniowe w sieci,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ korzysta z usługi FTP i dysku sieciowego,
- ✓ przeszukuje fora dyskusyjne,
- ✓ umie wybrać odpowiednią chmurę informatyczną na podstawie określonych wymagań,
- ✓ zakłada konto w darmowej chmurze informatycznej np. w Google lub Windows Live,
- ✓ udostępnia pliki z dysku chmury innym użytkownikom,
- ✓ posługuje się podstawowymi funkcjami systemu Linux lub Mac OS,
- ✓ instaluje program z Windows Live Essentials, np. Poczta systemu Windows,
- ✓ przywraca system, korzystając z punktu przywracania systemu,
- ✓ tworzy punkt przywracania systemu za pomocą narzędzia systemowego,
- ✓ opisuje warstwy modelu OSI,
- ✓ drogę informacji w komunikacji sieciowej i warstwowy model TCP/IP,

- ✓ umie skonfigurować połączenie z Internetem na podstawie znajomości maski, adresu bramy i DNS,
- ✓ ustawia automatyczne łączenie się komputera z wykrytą siecią i Internetem,
- ✓ konfiguruje podstawowe urządzenia sieci bezprzewodowej, Access Point i karty sieciowe,
- ✓ uruchamia konsolę MMC Windows, zna funkcje szablonów zabezpieczeń,
- ✓ udostępnia w sieci foldery z plikami za pomocą opcji udostępniania systemu Windows, odróżnia HTTP od HTML,
- ✓ wie, na czym polega transmisja strumieniowa i jakie zastosowanie znajduje RSS,
- ✓ wie, jakimi formatami dźwięku posługują się internetowe stacje radiowe,
- ✓

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ posługuje się pocztą elektroniczną i komunikatorami,
- ✓ wie, do czego służą usługi FTP, HTTP, SMTP, DHCP
- ✓ zakłada konto w chmurze informatycznej,
- ✓ loguje się do chmury informatycznej,
- ✓ wie, jakie programy oferuje dana chmura,
- ✓ edytuje tekst wspólnie z innymi członkami zespołu, jeśli taka możliwość zostanie mu udostępniona,
- ✓ określa podobieństwa i różnice pomiędzy interfejsami użytkownika różnych systemów operacyjnych,
- ✓ omawia sposoby dbania o higienę dysku twardego,
- ✓ chroni komputer przed wirusami,
- ✓ nazywa warstwowe modele sieci,
- ✓ wie, jaką rolę pełni adres IP w sieciach komputerowych,
- ✓ zna podstawowe pojęcia sieciowe np. DNS, MAC,
- ✓ wie, do czego służą i nazywa podstawowe urządzenia sieci komputerowej, w tym także bezprzewodowe punkty dostępowe,
- ✓ zna pojęcie sieć klient - serwer i wie, na czym polega praca takiej sieci,
- ✓ wie, jakie elementy i zasoby można udostępniać innym użytkownikom sieci,
- ✓ zna podstawowe formaty przesyłanych informacji w sieciach komputerowych,
- ✓ uruchamia odbiór internetowych stacji radiowych,

Spółeczeństwo w internecie

Ocenę celującą otrzymuje uczeń, który:

- ✓ opisuje metody stosowane w kryptografii symetrycznej i asymetrycznej,
- ✓ wyjaśnia zasadę stosowania podpisu elektronicznego,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ wyjaśnia, jak zwiększyć swoje bezpieczeństwo w sieci poprzez stosowanie różnych technik

Ocenę dobrą otrzymuje uczeń, który:

- ✓ wymienia zasady ochrony danych osobowych,
- ✓ opisuje zastosowania technologii komputerowej w różnych dziedzinach życia,
- ✓ opisuje rodzaje ataków sieciowych,
- ✓ Ocenę dostateczną otrzymuje uczeń, który:
- ✓ wymienia zasady komunikacji w sieci internet (netykieta),
- ✓ wymienia zagrożenia wynikające ze złej komunikacji w sieci,
- ✓ opisuje wpływ rozwoju technologii na zmiany w społeczeństwie,
- ✓ wymienia i opisuje rodzaje szkodliwego oprogramowania,
- ✓

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ wyjaśnia pojęcia: wykluczenie i włączenie cyfrowe,
- ✓ podaje przykłady negatywnych zachowań w sieci internet,

Strony WWW i grafika komputerowa

Ocenę celującą otrzymuje uczeń, który:

- ✓ tworzy style opisujące wygląd strony WWW,
- ✓ dodaje do strony elementy odpowiedzialne za jej responsywność,
- ✓ buduje stronę z wykorzystaniem systemu CMS i publikuje ją w internecie,
- ✓ tworzy złożone modele 3D.

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ korzysta ze ścieżek względnych i bezwzględnych w kodzie HTML,
- ✓ poprawnie tworzy tabele o dowolnej strukturze,
- ✓ dołącza style kaskadowe do dokumentu HTML,
- ✓ tworzy ciekawą stronę WWW i publikuje ją w internecie,
- ✓ poprawnie używa narzędzia do rysowania krzywych Béziera,
- ✓ wycina dowolne elementy z obrazu rastrowego,
- ✓ tworzy w programach do grafiki wektorowej infografiki według wzoru,
- ✓ tworzy bryły obrotowe 3D na podstawie ich przekroju.

Ocenę dobrą otrzymuje uczeń, który:

- ✓ umieszcza zdjęcia na stronie WWW,
- ✓ tworzy linki do zasobów zewnętrznych oraz miejsc w obrębie jednej strony,
- ✓ poprawnie i na różne sposoby korzysta z opisu kolorów w języku HTML,
- ✓ wymienia podstawowe narzędzia programu GIMP,
- ✓ korzysta z warstw podczas pracy z programem GIMP,
- ✓ pracuje na warstwach w programie do grafiki wektorowej.

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ opisuje podstawową strukturę strony w języku HTML,
- ✓ tworzy nagłówki w języku HTML,
- ✓ wstawia komentarze w kodzie HTML,

- ✓ tworzy listy uporządkowane i nieuporządkowane,
- ✓ rozumie cel pozycjonowania stron WWW,
- ✓ skaluje i kadruje obraz, dostosowując go do zadanego rozmiaru,
- ✓ wymienia podstawowe narzędzia programu Inkscape.

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ zapisuje plik, nadając mu rozszerzenie .html,
- ✓ rozróżnia sekcje HEAD i BODY oraz opisuje różnicę między tymi częściami kodu,
- ✓ wymienia podstawowe znaczniki formatowania tekstu w języku HTML,
- ✓ opisuje budowę znacznika HTML,
- ✓ wyjaśnia pojęcie responsywności strony WWW,
- ✓ uruchamia stronę WWW na smartfonie,
- ✓ określa różnicę pomiędzy grafiką rastrową a wektorową,
- ✓ zapisuje wynik swojej pracy w różnych formatach graficznych,
- ✓ wyjaśnia, jak uruchomić środowisko do grafiki 3D online,
- ✓ stosuje podstawowe filtry w programie Gimp,
- ✓ tworzy obrazy na warstwach,
- ✓ tworzy autokorektę obrazu

Edytor tekstu i prezentacje

Ocenę celującą otrzymuje uczeń, który:

- ✓ korzysta z różnych narzędzi (w tym mobilnych) podczas prezentacji,
- ✓ bierze udział w projektach zespołowych jako odpowiedzialny lider projektu, wypełnia wszystkie zadania wynikające z powierzonej mu roli w projekcie, Ocenę bardzo dobrą otrzymuje uczeń, który: wyjaśnia zasady stosowania prawa autorskiego, wykorzystuje narzędzia współpracy zdalnej, korzysta z automatycznej numeracji tytułów oraz tworzy spis treści,
- ✓ tworzy spisy ilustracji i tabel,
- ✓ pracuje z dokumentem wspólnie z innymi osobami, korzystając z narzędzi pracy grupowej,
- ✓ wykorzystuje opcje recenzji dokumentu,
- ✓ wygłasza prelekcję na wybrany temat zgodnie z zasadami dobrego wystąpienia,
- ✓ tworzy dokładny plan wystąpienia na dowolny temat,
- ✓ stosuje efekty na slajdach prezentacji,
- ✓ umieszcza filmy i ścieżki audio w prezentacji, • prezentuje kompletny projekt na forum klasy,
- ✓ Ocenę dobrą otrzymuje uczeń, który:
- ✓ tworzy i modyfikuje własne szablony oraz style tekstowe,
- ✓ dzieli tekst na kolumny,
- ✓ pracuje z wielostronicowym dokumentem w widoku konspektu,
- ✓ wymienia cechy dobrej prezentacji,
- ✓ tworzy ciekawe przejścia między slajdami,
- ✓ Ocenę dostateczną otrzymuje uczeń, który:
- ✓ korzysta z szablonów w edytorze tekstów,
- ✓ poprawnie stosuje style nagłówkowe,
- ✓ generuje losowe bloki tekstowe,
- ✓ ustawia marginesy w dokumencie,
- ✓ wyjaśnia, czym są e-zasoby,
- ✓ tworzy stronę tytułową w dokumencie tekstowym,
- ✓ wyjaśnia, jak przygotować dobre wystąpienie,

- ✓ zna narzędzia, dzięki którym można dobrać zestaw pasujących do siebie kolorów,
- ✓ opisuje pojęcie cyfrowej tożsamości,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ korzysta w podstawowym zakresie z formatowania tekstów w edytorze tekstowym,
- ✓ wymienia etapy pracy nad dobrym wystąpieniem publicznym,
- ✓ wymienia programy komputerowe do tworzenia prezentacji,

Metody algorytmiczne

Ocenę celującą otrzymuje uczeń, który:

- ✓ charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,
- ✓ pisze programy o podwyższonym stopniu trudności,
- ✓ optymalizuje rozwiązania,
- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych,
- ✓ bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- ✓ w projektach zespołowych przyjmuje rolę lidera.

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ projektuje algorytm badający właściwości liczb (podzielność, pierwszość, cyfry) w podanym przedziale liczb naturalnych przy pomocy schematu blokowego lub pseudokodu,
- ✓ projektuje algorytm generujące n kolejnych liczb naturalnych o zadanych właściwościach (podzielność, pierwszość, cyfry) w podanym przedziale liczb naturalnych przy pomocy schematu blokowego lub pseudokodu.
- ✓ umie oszacować złożoność czasową dla algorytmu rekurencyjnego,
- ✓ stosuje w programie rekurencję do sortowania metodą przez scalanie,
- ✓ pisze program rozwiązujący problem pakowania plecaka i wydawania reszty metoda programowania dynamicznego,
- ✓ stosuje w programie rekurencję do sortowania interpolacyjnego,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ projektuje algorytm rozkładu liczby na czynniki pierwsze przy pomocy schematu blokowego lub pseudokodu,
- ✓ projektuje algorytm zliczający sumę cyfr z jakich składa się liczba przy pomocy schematu blokowego lub pseudokodu,
- ✓ Projektuje algorytm zliczający sumę liczb w podanym ciągu przy pomocy schematu blokowego lub pseudokodu,
- ✓ stosuje w programie rekurencję do sortowania metoda quicksort,
- ✓ stosuje w programie rekurencję do sortowania binarnego (lowerbound, upperbound),
- ✓ stosuje w programie obliczającym symbol Newtona metodę dynamiczną,
- ✓ implementuje algorytm symulujący problem wieży Hanoi,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ definiuje pojęcie zdania logicznego, charakteryzuje podstawowe operacje logiczne

- ✓ (koniunkcja, alternatywa, negacja) oraz operatory logiczne,
- ✓ pisze programy wykonujące działania na liczbach całkowitych,
- ✓ projektuje algorytm naiwny sprawdzający, czy liczba jest pierwsza przy pomocy schematu blokowego lub pseudokodu,
- ✓ projektuje algorytmy stosujące rekurencję do rozwiązania problemu,
- ✓ pisze programy wczytujące i zapisujące dane z pliku,
- ✓ pisze program obliczający symbol Newtona metodą rekurencyjną (korzystając z definicji),
- ✓ pisze programy rozwiązujące problem metodą zachłanną,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja,
- ✓ wymienia sposoby reprezentacji algorytmów,
- ✓ korzysta ze środowiska programistycznego: pisze w nim kod, kompiluje i uruchamia program, odczytuje i zapisuje pliki, projektuje programy o niewielkim stopniu trudności przy pomocy schematów blokowych lub pseudokodu,
- ✓ wyjaśnia, na czym polega metoda „dziel i zwyciężaj”,
- ✓ stosuje funkcję losującą w tworzonych programach,
- ✓ omawia metody zachłanne na przykładzie problemu kasjera, harmonogramu sali, pakowania plecaka i wyszukiwania drogi,
- ✓ porównuje metody zachłanną i dynamiczną,
- ✓ uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania o niewielkim stopniu trudności.

Algorytmy na liczbach całkowitych

Ocenę celującą otrzymuje uczeń, który:

- ✓ charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania, w środowisku programistycznym
- ✓ pisze programy o podwyższonym stopniu trudności,
- ✓ optymalizuje rozwiązania,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ pisze programy stosując pętle zagnieżdżone,
- ✓ stosuje złożone warunki logiczne w instrukcjach sterujących,
- ✓ umie analizować poprawność tworzonych programów,
- ✓ rozumie pojęcie złożoności obliczeniowej programu.

Ocenę dobrą otrzymuje uczeń, który:

- ✓ określa specyfikację algorytmu (dane, wynik),
- ✓ pisze programy o różnym stopniu trudności, szacuje ich efektywność,
- ✓ przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, schematu blokowego, pseudokodu,
- ✓ dobiera typy danych do realizacji problemu,
- ✓ umie zastosować iteracje do przetwarzania danych wejściowych,
- ✓ Umie zastosować i poprawnie wypisać typ rzeczywisty

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ definiuje pojęcie zdania logicznego, charakteryzuje podstawowe operacje logiczne
- ✓ (koniunkcja, alternatywa, negacja) oraz operatory logiczne,
- ✓ charakteryzuje wybrane typy zmiennych służących do zapisu liczb całkowitych w języku C++:
- ✓ short int, int, long int, long long int,
- ✓ pisze programy wykonujące działania na liczbach całkowitych,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja,
- ✓ wymienia sposoby reprezentacji algorytmów,
- ✓ korzysta ze środowiska programistycznego: pisze w nim kod, kompiluje i uruchamia program, odczytuje i zapisuje pliki,
- ✓ pisze programy o niewielkim stopniu trudności,

Algorytmy na liczbach całkowitych i rzeczywistych z wykorzystaniem funkcji

Ocenę celującą otrzymuje uczeń, który:

- ✓ optymalizuje rozwiązania,
- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych wyszukuje palindromy lub anagramy w plikach tekstowych,
- ✓ w projektach zespołowych przyjmuje rolę lidera.

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ umie zastosować technikę zstępującą (rozkładać skomplikowane problemy na prostsze) i rozwiązywać je w wydzielonych funkcjach,
- ✓ budować uniwersalne funkcje do wielokrotnego wykorzystania,
- ✓ stosuje zaawansowane funkcje środowiska i języka programowania (np. z biblioteki STL),
- ✓ dobiera struktury danych i metody do rodzaju problemu,
- ✓ pisze programy konwertujące liczby między różnymi systemami pozycyjnymi,
- ✓ w programach wykonujących działania na liczbach w różnych systemach pozycyjnych wykorzystuje bibliotekę string i strukturalne typy danych,
- ✓ wykorzystuje rozwinięcie binarne liczby dziesiętnej w algorytmie szybkiego podnoszenia do potęgi, wykonuje operacje arytmetyczne na liczbach w różnych systemach, implementuje je w języku C++.

Ocenę dobrą otrzymuje uczeń, który:

- ✓ określa specyfikację algorytmu (dane, wynik),
- ✓ pisze programy o różnym stopniu trudności, szacuje ich efektywność,
- ✓ przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, schematu blokowego, pseudokodu,
- ✓ dobiera typy danych do realizacji problemu,
- ✓ stosuje zmienne typu unsigned w tworzonych programach, stosuje różne sposoby przekazywania parametrów do funkcji: przez wartość, referencję lub wskaźnik,
- ✓ implementuje w języku C++ i optymalizuje algorytm sprawdzający, czy liczba jest pierwsza,
- ✓ pisze program rozkładający liczby na czynniki pierwsze,
- ✓ stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW,

- ✓ wykorzystuje algorytm Euklidesa do działań na ułamkach,
- ✓ pisze programy konwertujące liczby między systemem dziesiętnym i binarnym,
- ✓ implementuje w języku C++ algorytmy wykonujące działania arytmetyczne na liczbach w różnych systemach,
- ✓ w algorytmach zamiany wykorzystuje zależności między systemami binarnym, ósemkowym i heksadecymalnym,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ implementuje w języku C++ algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem,
- ✓ przy pisaniu programów stosuje własne funkcje różnych typów, w tym funkcję typu void,
- ✓ wyjaśnia różnicę między parametrami formalnym i aktualnym, a także między zmiennymi lokalną i globalną,
- ✓ implementuje w języku C++ algorytm naiwny sprawdzający, czy liczba jest pierwsza,
- ✓ dodaje liczby binarne,
- ✓ konwertuje liczby między pozycyjnymi systemami liczbowymi,
- ✓ wykonuje działania arytmetyczne na liczbach w systemach liczbowych o różnych podstawach,
- ✓ przedstawia liczby w kodzie U2,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ korzysta z podstawowych funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych,
- ✓ wymienia podstawowe typy danych, operacje arytmetyczne i logiczne,
- ✓ podaje definicje liczby pierwszej i liczby złożonej,
- ✓ implementuje w języku C++ algorytm zliczający dzielniki danej liczby,
- ✓ omawia geometryczną interpretację algorytmu Euklidesa,
- ✓ omawia pojęcia: złożoność obliczeniowa algorytmu, algorytm naiwny, algorytm optymalny, złożoność pesymistyczna, złożoność oczekiwana (średnia),
- ✓ definiuje pojęcie systemów liczbowych,
- ✓ wymienia systemy liczbowe używane w informatyce,
- ✓ konwertuje liczby między systemami binarnym i decymalnym,
- ✓ dodaje pisemnie liczby binarne,

Rozwiązywanie problemów z wykorzystaniem struktur danych

Ocenę celującą otrzymuje uczeń, który:

- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych wyszukuje palindromy lub anagramy w plikach tekstowych,
- ✓ pisze program rozkładający liczbę złożoną na dwie liczby pierwsze (hipoteza Goldbacha),
- ✓ implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję,
- ✓ implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej,
- ✓ pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary),
- ✓ stosuje zaawansowane algorytmy i struktury danych do wyszukiwania spójnych podciągów,
- ✓ stosuje zaawansowane algorytmy wyszukiwania, np. najlepszego wyboru (trwałych par), stosując rekurencję,
- ✓ stosuje w programach algorytmy sortowania inne niż omawiane na lekcjach (np. heapsort), • bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- ✓ w projektach zespołowych przyjmuje rolę lidera.

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ pisze program realizujący algorytm sita Eratostenesa,
- ✓ implementuje w języku C++ algorytmy wyszukiujące spójne podciągi o różnych cechach,
- ✓ stosuje w programach algorytmy wyszukiwania lidera i idola w zbiorze,
- ✓ porównuje algorytmy iteracyjne i rekurencyjne (liczbę wykonywanych operacji), szacuje ich złożoność czasową,
- ✓ zapisuje w postaci programu rozszerzony algorytm Euklidesa, wyjaśnia jego działanie i zastosowanie, przy testowaniu liczby na pierwszość stosuje funkcję typu logicznego,
- ✓ wyszukuje liczby bliźniacze,
- ✓ wykorzystuje algorytm Euklidesa do działań na ułamkach, stosując struktury lub pary (typ pair),
- ✓ pisze program wyszukiujący jednocześnie minimum i maksimum w zbiorze z wykorzystaniem metody „dziel i zwyciężaj” oraz podaje wzór na liczbę wykonywanych operacji,
- ✓ szacuje złożoność obliczeniową programów sortujących, modyfikuje funkcje sortujące, zmieniając porządek sortowania,
- ✓ wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach,
- ✓ optymalizuje program realizujący algorytm sita Eratostenesa i szacuje jego złożoność czasową,
- ✓ pisze programy wyszukiujące lidera i idola w zbiorze, optymalizuje je, szacuje złożoność czasową,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ implementuje w języku C++ i optymalizuje algorytm sprawdzający, czy liczba jest pierwsza,
- ✓ pisze program rozkładający liczby na czynniki pierwsze,
- ✓ stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu, schematu blokowego,
- ✓ implementuje w języku C++ algorytmy wyszukiwania liniowego i liniowego z wartownikiem, porównuje ich efektywność,
- ✓ przedstawia w postaci listy kroków lub schematu blokowego algorytmy sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie, określa operacje dominujące,
- ✓ omawia algorytm sita Eratostenesa,
- ✓ przedstawia algorytmy znajdowania spójnych podciągów, wyznaczania najdłuższego z nich oraz podciągu o największej sumie elementów,
- ✓ omawia algorytm znajdowania idola i lidera w zbiorze,
- ✓ implementuje w języku C++ algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi,
- ✓ omawia rozszerzony algorytm Euklidesa,
- ✓ omawia algorytm zliczania znaków w tekście oraz wyszukiujący maksimum z wykorzystaniem tablic,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ omawia pojęcia: złożoność obliczeniowa algorytmu, algorytm naiwny, algorytm optymalny, złożoność pesymistyczna, złożoność oczekiwana (średnia),
- ✓ korzysta z podstawowych funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych, wymienia podstawowe typy danych, operacje arytmetyczne i logiczne,

- ✓ w pisanych programach korzysta ze strukturalnych typów danych: napisów, struktur, tablic,
- ✓ definiuje pojęcie systemów liczbowych,
- ✓ omawia metody sortowania prostego (bąbelkowe, przez wybieranie) oraz szybkiego i przez scalanie na przykładowych danych,
- ✓ wypisuje liczby pierwsze z zadanego przedziału, stosując metodę sita Eratostenesa,
- ✓ wyszukuje w ciągu liczb spójne podciągi (nierosnący, niemalejący, stały), wskazuje najdłuższe, oblicza ich sumę,
- ✓ wskazuje idola i lidera w zbiorze danych,
- ✓ definiuje pojęcia iteracji i rekurencji,
- ✓ omawia zasadę złotego podziału,
- ✓ opisuje rozszerzony algorytm Euklidesa,
- ✓ omawia algorytmy wyszukiwania liczby w zbiorach uporządkowanym i nieuporządkowanym,

Rozwiązywanie problemów z wykorzystaniem dynamicznych struktur danych

Ocenę celującą otrzymuje uczeń, który:

- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych, implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję,
- ✓ implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej,
- ✓ pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary),
- ✓ stosuje zaawansowane algorytmy i struktury danych do wyszukiwania spójnych podciągów,
- ✓ stosuje zaawansowane algorytmy wyszukiwania, np. najlepszego wyboru (trwałych par), stosując rekurencję,
- ✓ pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,
- ✓ stosuje w programach algorytmy sortowania inne niż omawiane na lekcjach (np. heapsort),
- ✓ bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- ✓ w projektach zespołowych przyjmuje rolę lidera.
- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych,
- ✓ optymalizuje programy, szacuje ich efektywność,
- ✓ wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach, np. sprawdzanie spójności grafu,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ wyszukuje spójne podciągi w plikach tekstowych, stosując optymalne algorytmy (w tym programowanie dynamiczne), wyjaśnia ich działanie,
- ✓ pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,
- ✓ do implementacji rozszerzonego algorytmu Euklidesa stosuje zarówno iterację, jak i rekurencję,
- ✓ stosuje metody zachłanną i dynamiczną w problemach kasjera, harmonogramu wykorzystania sali, pakowania plecaka i wyszukiwania drogi, wskazuje wady i zalety obu metod, szacuje złożoność czasową,
- ✓ aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy.

- ✓ charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,
- ✓ pisze programy o podwyższonym stopniu trudności: rozwiązuje zadania oznaczone trzema gwiazdkami w podręczniku,
- ✓ optymalizuje rozwiązania,
- ✓ stosuje zaawansowane funkcje środowiska i języka programowania,
- ✓ dobiera struktury danych i metody do rodzaju problemu,
- ✓ szacuje złożoność algorytmów, implementuje algorytmy grafowe – BFS, DFS, algorytm Dijkstry,
- ✓ implementuje algorytmy badające własności geometryczne,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ stosuje algorytm wyszukiwania binarnego i oszacowuje jego złożoność czasową,
- ✓ pisze programy sortujące metodami prostymi z zastosowaniem funkcji typu void,
- ✓ stosuje algorytmy sortowania szybkiego i przez scalanie,
- ✓ stosuje metodę zachłanną w programach – problem kasjera, harmonogram wykorzystania sali, wyszukiwanie drogi, pakowanie plecaka,
- ✓ uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej.
- ✓ pisze programy o różnym stopniu trudności, szacuje ich efektywność,
- ✓ dobiera typy danych do rozwiązania problemu,
- ✓ do przeglądania grafu stosuje algorytm przeszukiwania w głąb (DFS) oraz algorytm przeszukiwania grafu wszerek (BFS),
- ✓ omawia algorytm Dijkstry,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ formułuje algorytm wydawania reszty minimalną liczbą monet, harmonogramu wykorzystania sali, pakowania plecaka, znajdowania drogi metodami zachłanną i dynamiczną,
- ✓ uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania.
- ✓ wyróżnia operacje, które można wykonywać na dynamicznych strukturach danych (stosie, kolejce, liście, typie vector),
- ✓ omawia zastosowanie dynamicznych struktur danych na różnych przykładach,
- ✓ zapisuje wyrażenia algebraiczne bez użycia nawiasów, w tym w postaci odwrotnej notacji polskiej,
- ✓ oblicza wartość wyrażenia arytmetycznego zapisanego w odwrotnej notacji polskiej,
- ✓ omawia algorytmy znajdowania wyjścia z labiryntu z wykorzystaniem iteracji i rekurencji,
- ✓ symuluje problem Flawiusza,
- ✓ sortuje dane leksykograficznie,
- ✓ stosuje typ vector do reprezentacji grafu w postaci list sąsiedztwa,
- ✓ omawia algorytm przeszukiwania grafu w głąb (DFS),
- ✓ omawia algorytm przeszukiwania grafu wszerek (BFS),
- ✓ wyjaśnia, do czego służy algorytm Dijkstry,
- ✓ wyjaśnia różnicę między przekazywaniem parametrów do funkcji przez wartość i przez referencję,
- ✓ wykorzystuje pliki tekstowe do wczytywania danych i zapisywania wyników,
- ✓ omawia algorytm znajdujący rozwinięcie binarne nieskracalnego ułamka właściwego,
- ✓ omawia algorytmy badające własności geometryczne – położenie punktu względem prostej, przecinania się odcinków, przynależności punktu do figury,
- ✓ podaje przykłady fraktali (zbiór Cantora, drzewo binarne, dywan Sierpińskiego, płatek Kocha), wyjaśnia sposób tworzenia tych fraktali,
- ✓ wyjaśnia, na czym polegają metoda zstępująca i metoda wstępująca,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ pisze programy o niewielkim stopniu trudności,
- ✓ wyjaśnia, co to jest notacja infiksowa, notacja prefiksowa, odwrotna notacja polska, drzewo wyrażenia algebraicznego,
- ✓ definiuje pojęcie dynamicznej struktury danych,
- ✓ definiuje dynamiczne struktury danych takie jak: stos, kolejka, lista, vector,
- ✓ wymienia rodzaje list,
- ✓ wyjaśnia, na czym polega sortowanie leksykograficzne,
- ✓ definiuje graf, wymienia elementy i rodzaje grafów, wymienia sposoby reprezentacji grafu (macierz sąsiedztwa, lista sąsiedztwa),

Algorytmy numeryczne

Ocenę celującą otrzymuje uczeń, który:

- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ w reprezentacji liczb rzeczywistych w komputerze stosuje reprezentację stało- lub zmiennoprzecinkową zgodnie ze specyfikacją algorytmu, minimalizując błędy w obliczeniach,
- ✓ stosuje schemat Hornera do szybkiego podnoszenia do potęgi,
- ✓ implementuje algorytmy numeryczne: znajdowania miejsc zerowych funkcji oraz obliczania pierwiastka kwadratowego metodą bisekcji, obliczania pierwiastka kwadratowego metodą Newtona–Raphsona, obliczania pola obszaru zamkniętego metodą prostokątów i metodą trapezów, znajdowania przybliżenia liczby pi oraz symulacja ruchów Browna metodą Monte Carlo,
- ✓ pisze algorytmy stabilne dla problemów generujących przekroczenia błędów zaokrąglenia,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ świadomie używa typów float i double w zadaniach,
- ✓ stosuje schemat Hornera do zamiany liczby w systemie pozycyjnym o wybranej podstawie na liczbę dziesiętną,
- ✓ stosuje metodę Monte Carlo w obliczeniach przybliżonych,
- ✓ oblicza błąd względny i bezwzględny dla reprezentacji rzeczywistej liczby i podstawowych działań arytmetycznych,
- ✓ znajduje reprezentację liczby zapisanej w systemie dziesiętnym jako liczby pojedynczej i liczby podwójnej precyzji,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ znajduje pierwiastki równania kwadratowego algorytmem stabilnym i algorytmem niestabilnym,
- ✓ implementuje algorytm obliczający wartość wielomianu z zastosowaniem schematu Hornera,
- ✓ stosuje w algorytmach numerycznych metody: bisekcji, Newtona–Raphsona, trapezów, prostokątów,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ zapisuje liczby w postaci znormalizowanej,
- ✓ definiuje liczby pojedynczej precyzji i liczby podwójnej precyzji,
- ✓ wykonuje działania na liczbach zmiennoprzecinkowych,
- ✓ wskazuje różnice między algorytmem stabilnym a algorytmem niestabilnym,
- ✓ omawia różnice między stałoprzecinkową a zmiennoprzecinkową reprezentacją liczb rzeczywistych w komputerze,
- ✓ wymienia rodzaje błędów w obliczeniach komputerowych, rozróżnia błąd względny i bezwzględny,

Algorytmy na tekstach

Ocenę celującą otrzymuje uczeń, który:

- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych lub oznaczone trzema gwiazdkami w podręczniku,
- ✓ wyszukuje palindromy lub anagramy w plikach tekstowych,
- ✓ tworzy palindromy z napisów, dopisując minimalną liczbę znaków,
- ✓ implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej, pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary),
- ✓ bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- ✓ w projektach zespołowych przyjmuje rolę lidera.

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ stosuje różne sposoby przekazywania parametrów do funkcji, uzasadnia ich użycie,
- ✓ pisze funkcje typu logicznego, np. sprawdzającą, czy napis jest palindromem,
- ✓ sprawdza, czy napisy są anagramami, stosując sortowanie lub zliczanie znaków,
- ✓ szyfruje dane wczytane z pliku z uwzględnieniem polskich znaków diakrytycznych,
- ✓ pisze program odczytujący informację ukrytą za pomocą szyfru Cezara z wykorzystaniem analizy częstości znaków w tekście,
- ✓ stosuje funkcję haszującą oraz algorytm Karpa–Rabina w programach wyszukiwujących wzorzec w tekście,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ implementuje w języku C++ algorytmy sprawdzające, czy napis jest palindromem,
- ✓ pisze programy sprawdzające, czy dwa napisy są anagramami, wykorzystując funkcję sort z biblioteki STL,
- ✓ szyfruje dane wczytane z pliku tekstowego,
- ✓ implementuje w języku C++ algorytm zliczania znaków w tekście oraz wyszukiwujący maksimum z wykorzystaniem tablic,
- ✓ omawia algorytm Karpa–Rabina do wyszukiwania wzorca w tekście z zastosowaniem funkcji haszującej,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ korzysta z biblioteki string do operacji na łańcuchach znaków,
- ✓ wykonuje operacje na napisach, wykorzystując słowa kluczowe: size, find, substr, erase, toupper, tolower,
- ✓ wczytuje napisy ze spacjami, wykorzystując słowo kluczowe getline,
- ✓ tworzy algorytmy sprawdzające, czy napis jest palindromem,
- ✓ przedstawia w postaci algorytmu problem wyszukiwania anagramów,
- ✓ pisze program szyfrujący napis szyfrem Cezara,
- ✓ implementuje algorytm naiwny wyszukiwania wzorca w tekście,
- ✓ wyjaśnia metodę haszowania,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ wyjaśnia, czym jest tablica kodów ASCII,
- ✓ wyjaśnia, czym są palindrom i anagram, podaje przykłady,
- ✓ omawia szyfr Cezara jako przykład szyfru podstawieniowego i szyfr kolumnowy jako przykład szyfru przestawieniowego,
- ✓ omawia algorytm zliczania znaków w tekście,
- ✓ wczytuje dane z pliku tekstowego, zapisuje wyniki w pliku,
- ✓ wyszukuje wzorec w tekście algorytmem naiwnym,
- ✓ rozumie działanie funkcji haszującej,

Rozwiązywanie różnych problemów z wykorzystaniem komputera

Ocenę celującą otrzymuje uczeń, który:

- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych, implementuje w języku C++,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ pisze program znajdujący punkty kratowe zawarte w okręgu w czasie liniowym,
- ✓ pisze program znajdujący punkty kratowe zawarte w figurze,
- ✓ pisze program obliczający iloczyn wektorowy,
- ✓ pisze program badający przecięcie dwóch odcinków,
- ✓ pisze program badający wypukłość figury,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ pisze program obliczający punkty przecięcia okręgów
- ✓ pisze program sortujący podane punkty względem odległości od początku układu,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ pisze program obliczający pole wielokąta,
- ✓ pisze program badający współliniowość trzech punktów,
- ✓ pisze program znajdujący punkty kratowe zawarte w okręgu metodą naiwną,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ znajduje wartość wielomianu algorytmem naiwnym,
- ✓ umie reprezentować wektor na płaszczyźnie,
- ✓ zna proste algorytmy badające własności geometryczne (np. położenie punktu względem prostej),
- ✓ wyjaśnia, co to jest fraktal, wskazuje przykłady struktur fraktalnych występujących w przyrodzie,

Szyfrowanie informacji, aspekty bezpieczeństwa

Ocenę celującą otrzymuje uczeń, który:

- ✓ charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- ✓ pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych,
- ✓ bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- ✓ w projektach zespołowych przyjmuje rolę lidera.

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ pisze programy szyfrujące i deszyfrujące informacje w algorytmie RSA,
- ✓ pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere'a) i różnych kluczy,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ pisze program generujący klucz prywatny i klucz publiczny w algorytmie RSA,
- ✓ pisze program stosujący haszowanie w technice słownikowej

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ wyjaśnia, na czym polega łamanie szyfru,
- ✓ rozumie działanie funkcji haszującej,
- ✓ pisze program konwertujący tekst na kod Morse'a i odwrotnie.
- ✓ pisze program obliczający liczby względnie pierwsze w przedziale,
- ✓ rozumie mechanizmu działania podpisu cyfrowego,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ definiuje pojęcia: kryptologia, kryptografia, kryptoanaliza, tekst jawny, klucz, szyfrogram,
- ✓ rozróżnia szyfry podstawieniowe i przestawieniowe,
- ✓ wskazuje różnice między kryptografią symetryczną i kryptografią asymetryczną, definiuje pojęcia klucz publiczny i klucz prywatny, wyjaśnia, do czego służy algorytm RSA, i wyróżnia główne etapy tego algorytmu (generowanie kluczy, szyfrowanie z kluczem publicznym oraz deszyfrowanie z kluczem prywatnym)

Relacyjne bazy danych

Ocenę celującą otrzymuje uczeń, który:

- ✓ projektuje rozbudowane relacyjne bazy danych, zarządza nimi, wykorzystując zaawansowane narzędzia oraz klauzule języka SQL,
- ✓ tworzy rozwinięte responsywne aplikacje internetowe wymagające dużego nakładu pracy i znajomości nowoczesnych technologii,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ projektuje zaawansowane relacyjne bazy danych, zarządza nimi, tworzy tabele pomostowe, formularze, kwerendy i raporty,
- ✓ formułuje zapytania w języku SQL, stosując selekcję, sortowanie, projekcję oraz agregowanie danych,
- ✓ tworzy aplikacje internetowe z przejrzystym interfejsem użytkownika korzystające z sieciowej bazy danych, testuje je i wprowadza poprawki,
- ✓ wyszukuje informacje w bazach danych, stosując różne techniki (w tym konstruowanie rozbudowanych zapytań),
- ✓ umie wprowadzić dane bezpośrednio do tabeli lub poprzez formularz,
- ✓ umie poprawnie korzystać z formularzy w celu wprowadzania, wyszukiwania i edycji danych,
- ✓ potrafi drukować tabele i formularze,
- ✓ tworzy i modyfikuje formularze oraz formanty za pomocą kreatora i w widoku projektu,
- ✓ dodaje etykiety, formanty, nagłówek lub stopkę do formularza, zmienia jego szatę graficzną,
- ✓ tworzy proste raporty przy użyciu narzędzia Autoraport,
- ✓ umie korzystać z kreatora raportów do generowania dowolnych rodzajów raportów,
- ✓ potrafi grupować informacje w raporcie,
- ✓ umie korzystać z podsumowania statystycznego,
- ✓ przygotowuje raport do druku,
- ✓ definiuje makropolecenia składające się z sekwencji kilku akcji,
- ✓ stosuje 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,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ projektuje i tworzy proste bazy danych,
- ✓ korzysta z danych przechowywanych w innych bazach programu Access oraz utworzonych w innych aplikacjach,
- ✓ 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,
- ✓ konstruuje pola obliczeniowe, wprowadza korekty,
- ✓ stosuje kwerendy parametryczne,
- ✓ tworzy kwerendy funkcjonalne generujące tabele, aktualizujące dane, usuwające i dołączające dane,
- ✓ tworzy zestawienia krzyżowe, odpowiednio definiując kryteria, nagłówki wierszy i kolumn,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ wyszukuje informacje w bazach danych, tworzy formularze, kwerendy i raporty,
- ✓ wykorzystuje język SQL do tworzenia i usuwania baz danych, dodawania tabel do baz danych, usuwania tabel z baz, dodawania rekordów do tabel, importowania danych do tabel, edycji rekordów,
- ✓ tworzy konta użytkowników i przydziela im uprawnienia do wybranej bazy, używając języka SQL,
- ✓ formułuje zapytania zwracające określone dane, sortuje wyniki zapytań,
- ✓ 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,
- ✓ 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.
- ✓ 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,
- ✓ wykonuje podstawowe czynności edycyjne na obiektach bazy danych (kopiowanie, usuwanie, zmiana nazwy),
- ✓ potrafi wstawiać obiekty OLE,
- ✓ odróżnia osadzenie obiektu w tabeli od jego połączenia,
- ✓ 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,

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ wie, co to jest język SQL, zna podstawowe klauzule tego języka,
- ✓ zna zasady tworzenia zapytań do bazy z wykorzystaniem języka SQL,
- ✓ na przykładzie istniejącej relacyjnej bazy danych wymienia obiekty związane z takimi bazami (tabele, kwerendy, formularze, raporty, makrodefinicje).

Arkusz kalkulacyjny

Ocenę celującą otrzymuje uczeń, który:

- ✓ potrafi korzystać z dodatku Solver
- ✓ pisze własne funkcje w module Visual Basic

Ocenę bardzo dobrą otrzymuje uczeń, który:

- ✓ 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),
- ✓ świadomie wybiera właściwy sposób rozwiązania zadania,
- ✓ wykorzystuje funkcje arkusza kalkulacyjnego i poznane metody wyszukiwania informacji do rozwiązywania problemu,

Ocenę dobrą otrzymuje uczeń, który:

- ✓ pisze własne makropolecenia automatyzujące pracę,
- ✓ 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,
- ✓ potrafi dobrać odpowiedni typ wykresu do prezentowanych na nim danych,
- ✓ umie tworzyć różnego typu wykresy do danych zawartych w tabelach zwykłych i przestawnych,

Ocenę dostateczną otrzymuje uczeń, który:

- ✓ potrafi stosować złożone kryteria wyboru w filtrach zaawansowanych,
- ✓ stosuje odpowiedni rodzaj filtru w celu wybrania potrzebnych w danej sytuacji informacji,
- ✓ 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,
- ✓ za pomocą wbudowanych funkcji arkusza kalkulacyjnego dokonuje operacji na tekście.

Ocenę dopuszczającą otrzymuje uczeń, który:

- ✓ zna i stosuje reguły, jakim podlega tabela stanowiąca źródło danych,
- ✓ odwołuje się komórek za pomocą adresów względnych, bezwzględnych i mieszanych.
- ✓ 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,
- ✓ korzysta z autofiltru w celu wyselekcjonowania danych,
- ✓ tworzy wykresy przestawne za pomocą kreatora
- ✓ potrafi korzystać z wbudowanych funkcji arkusza kalkulacyjnego do rozwiązania problemu.