

**WYMAGANIA EDUKACYJNE Z INFORMATYKI NIEZBĘDNE DO UZYSKANIA PRZEZ UCZNIA**

**POSZCZEGÓLNYCH ŚRÓDROCZNYCH I ROCZNYCH OCEN KLASYFIKACYJNYCH WYNIKAJĄCYCH Z REALIZOWANEGO PROGRAMU NAUCZANIA**

**LICEUM 4-LETNIE KLASA 2.**

**ZAKRES ROZSZERZONY**

Uczeń spełnia wszystkie wymagania edukacyjne dla poziomu podstawowego, a ponadto wymagania wyszczególnione poniżej.

|  |
| --- |
| **Szczegółowe wymagania edukacyjne dla klasy 2** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Reprezentacja danych w komputerze- zadania maturalne | | | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | | |
| zna pojęcie systemu pozycyjnego;  wie, co to jest system binarny;  analizuje gotowy przykład obliczający wartość dziesiętną liczby zapisanej w systemie dwójkowym i na tej podstawie wykonuje podobne ćwiczenie;  w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą; | definiuje pojęcie systemu pozycyjnego;  wie na czym polega działanie procesora;  potrafi dokonać konwersji liczby między systemem dziesiętnym a dwójkowym oraz dwójkowym a dziesiętnym;  do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych; | zna podwójne nazwy pozycyjnych systemów liczbowych tj. dwójkowy – binarny, dziesiętny – decymalny, szesnastkowy - heksadecymalny;  wyjaśnia, w jaki sposób procesor dodaje liczby;  dokonuje konwersji liczb między systemem dziesiętnym i szesnastkowym oraz szesnastkowym i dziesiętnym;  objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność; | zna elementy uproszczonego modelu komputera zgodny z ideą von Neumanna;  dokonuje konwersji liczb między systemem szesnastkowym i binarnym;  ilustruje i wyjaśnia rolę pojęć, obiektów i operacji matematycznych w projektowaniu rozwiązań problemów informatycznych i z innych dziedzin, posługuje się pojęciem logarytmu;  przedstawia sposoby reprezentowania w komputerze znaków, liczb, wartości logicznych, obrazów, dźwięków, animacji; | potrafi narysować uproszczony model komputera zgodny z ideą von Neumanna;  samodzielnie potrafi dokonać zamiany między trzema systemami pozycyjnymi (dwójkowym, dziesiętnym i szesnastkowym) w jednym zadaniu;  objaśnia sposoby wykonywania przez komputer działań arytmetycznych i operacji logicznych;  wyjaśnia, jakie może być źródło błędów pojawiających się w obliczeniach komputerowych: błąd zaokrąglenia, błąd przybliżenia;  dyskutuje na temat roli myślenia komputacyjnego i jego metod, takich jak: abstrakcja, reprezentacja danych, dekompozycja problemu, redukcja, myślenie rekurencyjne, podejście heurystyczne w rozwiązywaniu problemów z różnych dziedzin. | | |
| Wykonywanie operacji w arkuszu kalkulacyjnym- zadania maturalne | | | | | |
| Wymagania edukacyjne niezbędne do uzyskania **oceny dopuszczającej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dostatecznej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny bardzo dobrej** | | Wymagania edukacyjne niezbędne do uzyskania **oceny celującej** |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | | **Uczeń:** |
| z pomocą nauczyciela tworzy wykres funkcji liniowej, z pomocą nauczyciela pobiera surowe dane z czujników z pomocą nauczyciela planuje kolejne kroki symulacji w arkuszu porządkuje dane, aby móc utworzyć tabelę przestawną z pomocą nauczyciela tworzy ankietę w chmurze. | tworzy wykres funkcji liniowej przygotowuje dane do analizy korzysta z funkcji zaokrąglania wyników, tworzy wykresy przebiegu w czasie pobiera i importuje do arkusza wyniki ankiety | tworzy wykres funkcji kwadratowej zmienia wartości opracowuje pobrane dane, dobiera odpowiednie narzędzia przeprowadza symulację samodzielnie korzysta z Pomocy arkusza tworzy tabele przestawne stosuje zaawansowane kryteria filtrowania oraz poprawne funkcje w formułach; | tworzy złożone wykresy funkcji automatyzuje proces tworzenia wykresów, opracowuje wyniki wykorzystuje linie trendu w wykresach funkcji liniowej wprowadza dynamiczne tytuły osi wykresów prawidłowo dobiera pola do wyświetlania w tabeli przestawnej dokonuje wizualizacji danych z wykorzystaniem wykresów przebiegu w czasie korzysta z fragmentatorów tworzy raporty z danych z wykorzystaniem tabeli przestawnych. Sum częściowych, odpowiednich funkcji i wykresów przebiegu w czasie, rozwiązuje zadania maturalne z niewielką pomocą nauczyciela; | | przygotowuje trudniejsze wykresy, np. wykresy przestrzenne funkcji dwóch zmiennych samodzielnie wykonuje doświadczenia i eksperymenty analizuje wyniki dodatkowych doświadczeń i eksperymentów samodzielnie planuje i realizuje symulacje, np. o charakterze przyrodniczym samodzielnie wykorzystuje tabele przestawne do analizy różnych danych samodzielnie planuje i realizuje badanie na porządkuje dane i tworzy raport wybrany temat – przeprowadza ankietę, rozwiązuje samodzielnie, bezbłędnie zadania maturalne w programie Excel |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stosowanie instrukcji iteracyjnych | | | | |
| Wymagania edukacyjne niezbędne do uzyskania **oceny dopuszczającej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dostatecznej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny bardzo dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny celującej** |
| Uczeń analizuje i omawia działanie gotowych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli **for**.  Analizuje i omawia działanie gotowych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli **while** | Uczeń zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady.  Zna postać i działanie instrukcji iteracyjnej **while** w językach C++ i/lub Python i stosuje ją w tworzonych programach komputerowych | Uczeń zna sposoby zakończenia iteracji. Określa kroki iteracji.  Stosuje instrukcję **while** w programach komputerowych.  W języku C++ stosuje instrukcję **do … while** w programach komputerowych.  Zna różne sposoby określania liczby iteracji w języku Python, np. poprzez powtarzanie poleceń zawartych wewnątrz pętli dla konkretnych wartości lub poprzez podanie liczby powtórzeń z zastosowaniem funkcji **range()**. | Uczeń rozumie różnicę pomiędzy instrukcją **for** a instrukcją **while** w wybranym języku programowania.  Modyfikuje programy, zamieniając pętlę **for** na **while** i odwrotnie. Ocenia program po zmianie (styl, czytelność).  Pisze programy, stosując poznane instrukcje iteracyjne.  Dobiera odpowiednią technikę algorytmiczną i odpowiednie struktury danych do rozwiązywanego problemu. | Uczeń potrafi samodzielnie zastosować odpowiedni rodzaj instrukcji pętli w tworzonym programie.  Potrafi samodzielnie dobrać odpowiednią instrukcję **while** lub **do… while**.  Omawia podobieństwa i różnice w działaniu wszystkich omówionych instrukcji pętli w języku C++.  Sprawnie pisze trudniejsze programy, stosując poznane instrukcje iteracyjne.  Wymyśla samodzielnie problem iteracyjny, formułuje zadanie, pisze jego specyfikację i listę kroków oraz zapisuje w wybranym języku programowania |
| Iteracyjna realizacja wybranych algorytmów, w tym algorytmów wyszukiwania | | | | |
| Wymagania edukacyjne niezbędne do uzyskania **oceny dopuszczającej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dostatecznej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny bardzo dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny celującej** |
| Uczeń potrafi odróżnić algorytm liniowy od algorytmu iteracyjnego.  Omawia algorytm znajdowania elementu najmniejszego. Analizuje listę kroków i rysuje schemat blokowy na jej podstawie.  Analizuje i omawia gotową listę kroków i schemat blokowy algorytmu Euklidesa w jednej z wersji. Testuje algorytm na podstawie listy lub schematu | Uczeń zna przykładowe algorytmy na liczbach naturalnych;  Wie, na czym polega metoda wyszukiwania liniowego i przez połowienie.  Zapisuje algorytm znajdowania najmniejszego (największego) elementu w postaci programu.  Zna iteracyjną postać algorytmu Euklidesa w wersji z odejmowaniem.  Potrafi napisać listy kroków algorytmu i narysować schemat blokowy Euklidesa w wersji z odejmowaniem.  Zna metodę „dziel i zwyciężaj”.  Zapisuje algorytm Euklidesa w wersji z odejmowaniem w postaci programu w wybranym języku programowania | Uczeń potrafi omówić algorytm naiwny i optymalny jednoczesnego znajdowania największego i najmniejszego elementu w zbiorze.  Zna iteracyjną postać algorytmu Euklidesa z resztą z dzielenia. Pisze listę kroków tego algorytmu.  Potrafi narysować schemat blokowy algorytmu Euklidesa w wersji z resztą z dzielenia.  Zapisuje algorytm Euklidesa w wersji z resztą z dzielenia w postaci programu w wybranym języku programowania.  Wyjaśnia na przykładzie różnicę między wersją algorytmu Euklidesa z odejmowaniem a wersją z resztą z dzielenia. | Uczeń określa liczbę porównań w algorytmie naiwnym i optymalnym znajdowania największego i najmniejszego elementu w zbiorze. Porównuje otrzymane wyniki.  Pisze listę kroków algorytmu jednoczesnego znajdowania minimum i maksimum z wykorzystaniem metody dziel i zwyciężaj. Zapisuje ten algorytm w postaci programu w języku C++ i/lub Python.  Omawia zastosowanie schematu Hornera do obliczania wartości wielomianu; pisze listę kroków, rysuje schemat blokowy tego algorytmu i pisze program realizujący algorytm obliczania wartości wielomianu według schematu Hornera.  Programując w/w algorytmy, definiuje odpowiednie funkcje, dobiera struktury danych. Dba o stosowanie podstawowych zasad programowania | Uczeń podaje przykłady problemów, w których można zastosować wyszukiwanie liniowe lub przez połowienie.  Pisze trudniejsze programy komputerowe, w których wykorzystuje poznane algorytmy.  Korzystając z dodatkowych źródeł, wyszukuje informacje o zastosowaniu metody „dziel i zwyciężaj” oraz pisze program według własnego pomysłu pokazujący zastosowanie tej metody.  Samodzielnie zapoznaje się ze schematem Hornera i zapisuje go w postaci programu, dopierając poprawne struktury danych.  Korzystając z dodatkowych źródeł, omawia przykłady zastosowań algorytmu Euklidesa i poznaje trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, algorytm znajdowania liczb bliźniaczych. Potrafi zapisać je w języku programowania. |
| Elementy analizy algorytmów | | | | |
| Wymagania edukacyjne niezbędne do uzyskania **oceny dopuszczającej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dostatecznej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny bardzo dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny celującej** |
| Uczeń wymienia własności algorytmów.  Potrafi przeanalizować przebieg prostego algorytmu zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego poprawność | Uczeń zna i omawia własności algorytmów.  Wie, kiedy algorytm jest poprawny.  Potrafi przeanalizować przebieg algorytmu (np. obliczania silni) zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego poprawność.  Analizuje program wybranego algorytmu (np. obliczania silni) i ocenia jego poprawność | Uczeń wie, jak sprawdzić, czy algorytm jest skończony.  Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją.  Określa liczbę prostych działań zawartych w algorytmie.  Określa liczbę prostych działań zawartych w algorytmie.  Potrafi poprawić program, który jest niepoprawny | Uczeń wie, kiedy algorytm jest skończony.  Potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego skończoność.  Analizuje program realizujący wybrany algorytm i ocenia jego skończoność. Modyfikuje program, aby działał poprawnie.  Sprawdza poznane własności algorytmów, rozwiązując zadania, m.in. uzasadnia skończoność algorytmu znajdowania największego wspólnego dzielnika (NWD) dwóch liczb naturalnych.  Oblicz liczbę operacji porównania w algorytmie wyboru minimum z tablicy zawierającej *n* losowo uporządkowanych liczb. | Uczeń potrafi samodzielnie ocenić poprawność i skończoność wybranych algorytmów.  Potrafi samodzielnie wymyśleć zadanie (problem), napisać specyfikację zadania, listę kroków i program realizujący to zadanie.  Korzysta samodzielnie z dodatkowej literatury fachowej.  Poznane dodatkowe możliwości wybranego języka programowania (np. standardowe funkcje) i stosuje w programach.  Rozwiązuje przykładowe zadania z olimpiady informatycznej. |
| Praca w chmurze i zadania projektowe | | | | |
| Wymagania edukacyjne niezbędne do uzyskania **oceny dopuszczającej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dostatecznej** | Wymagania edukacyjne niezbędne do uzyskania **oceny dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny bardzo dobrej** | Wymagania edukacyjne niezbędne do uzyskania **oceny celującej** |
| Uczeń zna etapy pracy nad projektem i bierze udział w pracy grupowej jako członek zespołu.  Uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje | Uczeń planuje temat projektu.  Bierze aktywny udział w pracy grupowej jako członek zespołu, gromadząc i selekcjonując materiały do projektu.  Bierze udział w testowaniu projektu.  Uczestniczy w przygotowaniu dokumentacji projektu, korzystając z edytora tekstu; ewentualnie z wybranego szablonów. | Korzystając z chmury, potrafi udostępnić pliki, linki do folderu; umożliwić współdzielenie danego folderu.  Realizuje projekt na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania; przygotowuje dokumentację projektu.  Współpracuje w grupie, wykonując projekt na temat projektowania zakupu nowego zestawu komputerowego oraz oprogramowania dla ucznia szkoły ponadpodstawowej; oraz wpływu trendów w historycznym rozwoju pojęć i metod informatyki oraz technologii na możliwości rozwoju grafiki komputerowej | Uczeń korzysta z oprogramowania dostępnego w chmurze, m.in. tworzy dokumenty w edytorze tekstu i umieszcza w chmurze. Współdzieli je z innymi współużytkownikami.  Przygotowuje wybrane zadanie szczegółowe zgodnie z etapami przygotowania projektu. Gromadzi materiały i inne pomoce, opracowuje dokumentację projektu, wykorzystując m.in. możliwości pracy w chmurze.  Udostępnia pliki, linki do folderu.  Wyszukuje informacje na e-platformach do e-nauczania.  Prezentuje wykonane zadanie projektowe wykorzystaniem projektora innych przygotowanych pomocy lub materiałów. Inicjuje dyskusje | Uczeń sprawnie posługuje się środowiskiem przeznaczonym do współpracy i realizacji projektów zespołowych, w tym środowiskiem w chmurze.  Potrafi pełnić funkcję koordynatora grupy.  Koordynuje wykonywanie zadań szczegółowych na poszczególnych etapach. Zarządza folderami (współdzieleniem) i pracą nad dokumentami, w tym dokumentacją projektu. Ustala sposób prezentacji projektu i wyznacza osobę (osoby) do prezentacji.  W miarę możliwości współtworzy zasoby udostępniane na platformach do e-nauczania.  Przygotowuje projekt na wybrany przez siebie temat |