# Wymagania edukacyjne z informatyki klasa 3 – zakres podstawowy

**Szczegółowe wymagania edukacyjne dla klasy 3B, 3C, 3D, 3E**

| Lp | Temat | Wymagania na ocenę dopuszczającą | Wymagania na ocenę dostateczną | Wymagania na ocenę dobrą | Wymagania na ocenę bardzo dobrą | Wymagania na ocenę celującą |
|---|---|---|---|---|---|---|
| **Algorytmika i programowanie** | | | | | | |
| 1 | Podstawy tworzenia algorytmów | – wie, ze algorytm można zapisać w różnych postaciach i wymienia nazwy tych sposobów<br>– zna podstawowe zasady tworzenia schematów blokowych w tym dozwolone i niedozwolone połączenia | – prawidłowo interpretuje działanie bloku decyzyjnego i wie, jaką rolę odgrywa on w algorytmie<br>– wie, jaka jest różnica pomiędzy blokiem decyzyjnym a wejściowym/wyjściowym | – samodzielnie analizuje przykład algorytmu, np. z podręcznika z pomocą opisu<br>– buduje algorytmy prostych zagadnień z różnych dziedzin lub przedmiotów szkolnych<br>– umie ułożyć prosty algorytm w postaci schematu blokowego na podstawie algorytmu typu lista kroków zawierający blok decyzyjny | – samodzielnie tworzy algorytmy na podstawie specyfikacji zawierające bloki decyzyjne<br>– samodzielnie testuje algorytmy dla różnych przypadków<br>– samodzielnie dyskutuje sposób rozwiązania problemu na podstawie algorytmu, np. z podręcznika, i proponuje jego modyfikacje | – analizuje różne algorytmy i testuje je na samodzielnie i odpowiednio dobranych danych<br>– samodzielnie proponuje modyfikacje przykładowych algorytmów |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | Algorytmy warunkowe i iteracyjne | – opisuje działanie instrukcji warunkowej;<br>– z pomocą nauczyciela interpretuje algorytmy zawierające bloki decyzyjne;<br>– definiuje zmienną sterującą i omawia jej znaczenie w pętlach; | – korzystając z przykładów w podręczniku, układa programy z instrukcją warunkową na podstawie algorytmu z blokiem decyzyjnym;<br>– umie wybrać rodzaj instrukcji powtarzania do rozwiązania określonego problemu;<br>– pamięta o deklaracji zmiennej sterującej;<br>– układa proste programy z pętlami, korzystając z przykładów rozwiązań podobnych problemów; | – prawidłowo układa fragmenty programów z zastosowaniem procedur i funkcji w pętlach instrukcji powtarzania, np. wczytywanie znaku z klawiatury, oczekiwanie na wciśnięcie klawisza itp.; | Prawidłowo układa algorytmy z instrukcjami zagnieżdżonymi | – samodzielnie opracowuje algorytmy, które mogą być realizowane za pomocą instrukcji powtarzania, i na ich podstawie układa programy w języku C++; |
| 3 | Złożoność obliczeniowa, poprawność algorytmu | – wie, ze złożoność algorytmu można zapisać w | – prawidłowo interpretuje złożoność | - prawidłowo potrafi wskazać złożoność | - zna system zapisu złożoności obliczeniowej w | – analizuje różne algorytmy i testuje je na samodzielnie i |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | różnych postaciach i wymienia nazwy tych sposobów<br>- wie jakie kryteria musi spełnić poprawny algorytm | pamięciową i obliczeniową<br>- potrafi wskazać błędy w algorytmie | optymistyczną i pesymistyczną<br>- umie wprowadzić zmiany do algorytmu tak aby był poprawny | formie notacji dużego O. | odpowiednio dobranych danych |
| 4 | Sprawdzian | | | | | |
| 5-6 | Systemy Liczbowe | – charakteryzuje liczby szesnastkowe i dwójkowe<br>– wie, jak są zbudowane pozycyjne systemy liczbowe | – na przykładzie z podręcznika omawia działanie algorytmów zamiany postaci liczby dziesiętnej na dwójkową, dziesiętnej na szesnastkową i dwójkowej na szesnastkową<br>– na przykładzie z podręcznika omawia działanie funkcji zamieniających postaci liczb | – na podstawie opisu i algorytmów z podręcznika układa funkcje zamieniające prezentację liczb z dziesiętnej na dwojkową i szesnastkową oraz z dwojkowejna szesnastkową | – samodzielnie układa programy (funkcje) zamieniające prezentacje liczb<br>– układa programy wykorzystujące funkcje zamieniające prezentacje liczb | – samodzielnie układa uniwersalne programy oparte o funkcje zamieniające prezentacje liczb na wiele postaci |
| 7 | Sprawdzian | | | | | |
| 8 | Budowa programu w C++ | – wymienia i charakteryzuje kolejne etapy tworzenia | – korzystając z podręcznika, omawia podstawową strukturę | – samodzielnie omawia działanie poszczególnych operatorów wszystkich typów | – samodzielnie układa proste programy ze strumieniowaniem | – samodzielnie stosuje instrukcje warunkowe i pętle w prostych programach |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | programu komputerowego<br>– wie, czym jest badanie warunku w programie i kiedy się je stosuje w kontekście bloków warunkowych algorytmu<br>– wie, że istnieją różne typy operatorów i na podstawie podręcznika omawia rolę niektórych z nich<br>– wie, że w programie mogą występować biblioteki i funkcje | programu w języku C++<br>– zna znaczenie nawiasów klamrowych i "//" oraz ich rolę w programie C++<br>– na podstawie tabeli z podręcznika omawia znaczenie operatorów<br>– odróżnia operatory arytmetyczne od relacyjnych i logicznych i symboli porównawczych<br>– wie, czym są zmienne i stałe w programie komputerowym i wskazuje ich deklarację w przykładowym programie<br>– zna znaczenie i rolę funkcji i bibliotek<br>– zna pojęcie pętli i warunku | – umie zapisać warunki dla instrukcji warunkowej<br>– umie dołączyć bibliotekę do kodu programu<br>– wymienia różnice pomiędzy instrukcją warunkową a pętlą | – samodzielnie charakteryzuje najczęściej używane typy zmiennych liczbowych i logicznych<br>– samodzielnie zapisuje fragment programu z instrukcją warunkową na podstawie algorytmu<br>– wskazuje, w których przypadkach należy użyć danej instrukcji warunkowej lub pętli | układanych na podstawie algorytmów |

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | Instrukcja warunkowa | – opisuje działanie instrukcji warunkowej;<br>– z pomocą nauczyciela interpretuje algorytmy zawierające bloki decyzyjne; | – korzystając z przykładów w podręczniku, układa programy z instrukcją warunkową na podstawie algorytmu z blokiem decyzyjnym; | – samodzielnie układa programy z wykorzystaniem instrukcji warunkowej na podstawie algorytmu;<br>– korzystając z opcji pomocy i przykładów z podręcznika, układa programy z instrukcjami warunkowymi złożonymi i zagnieżdżonymi; | – samodzielnie układa programy z instrukcjami warunkowymi złożonymi i zagnieżdżonymi; | – samodzielnie opracowuje proste algorytmy z blokami decyzyjnymi i na ich podstawie układa programy w języku C++; |
| 10 | Sprawdzian | | | | | |
| 11-13 | Instrukcje iteracji | – omawia i rozumie różnice pomiędzy for, do..while oraz while;<br>– podaje przykłady zastosowań for, do..while oraz while;<br>– definiuje zmienną sterującą i omawia jej znaczenie w for; | – umie wybrać rodzaj instrukcji powtarzania do rozwiązania określonego problemu;<br>– pamięta o deklaracji zmiennej sterującej;<br>– układa proste programy z pętlami, korzystając z | – prawidłowo układa fragmenty programów z zastosowaniem procedur i funkcji w pętlach instrukcji powtarzania, np. wczytywanie znaku z klawiatury, oczekiwanie na | – układa programy, prawidłowo wykorzystując instrukcje powtarzania do budowy funkcjonalnego programu, np. w programie liczącym głosy;<br>– używa instrukcji powtarzania do układania funkcji i | – samodzielnie opracowuje algorytmy, które mogą być realizowane za pomocą instrukcji powtarzania, i na ich podstawie układa programy w języku C++;<br>– przewiduje w algorytmach konieczność |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | – omawia znaczenie warunku w pętlach; | przykładów rozwiązań podobnych problemów; | wciśnięcie klawisza itp.;<br>– prawidłowo wybiera pomiędzy for, do..while oraz while instrukcję realizującą pętle z określoną lub nieokreśloną liczbą powtórzeń; | procedur wywoływanych przez program główny, np. sprawdzania, czy został wciśnięty klawisz, podano prawidłowe hasło itp.; | stosowania instrukcji for, do..while lub while; |
| 14 | Znajdowanie elementu max i min | – omawia istotę metody wyszukiwania elementów max i min w zbiorze | – na podstawie podręcznika omawia istotę metody znajdowania liniowego elementu max i min | – omawia algorytm w postaci pseudokodu znajdowania liniowego elementu max i min | – samodzielnie tworzy algorytm w postaci pseudokodu znajdowania elementu max i min. | – układa program dla metody liniowej |
| 15 | Dzielniki liczby naturalnej, liczba pierwsza i złożona | – zna definicję liczby pierwszej i umie wymienić kilka z nich, wskazując spełnienie podstawowej cechy<br>– wymienia nazwę metod badania, czy | – na podstawie tabeli z podręcznika umie objaśnić metodę sita Erastotenesa<br>– objaśnia algorytmy badania, czy liczba jest liczbą pierwszą | – zna algorytmy zapisane w różnych postaciach wykrywające liczby pierwsze<br>– sprawdza na przykładach działanie algorytmów wykrywających liczby pierwsze | – samodzielnie tworzy i omawia działanie algorytmów wykrywających liczby pierwsze<br>– przedstawia algorytmy w różnych zapisach w tym schematu blokowego i listy kroków | – samodzielnie układa algorytm testowania liczb na podstawie opisu metody |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | liczba jest liczbą pierwszą | – zna zastosowanie liczb pierwszych | | – samodzielnie testuje algorytmy i dobiera odpowiednie dane wejściowe | |
| 16 | Wyznaczanie NWD i NWW | – definiuje NWD i omawia jego zastosowanie w matematyce<br>– podaje kilka przykładów NWD dla wybranych liczb | – na podstawie gotowego zapisu przykładu algorytmu Euklidesa, np. z podręcznika, omawia istotę tej metody<br>– podaje i uzasadnia dziedzinę liczb, dla których przeznaczony jest algorytm Euklidesa | – omawia różnicę pomiędzy metodą rekurencyjną a iteracyjną<br>– analizuje gotowy przykład zastosowania metod Euklidesa<br>– przedstawia algorytmy Euklidesa, np. w formie schematu blokowego, i tłumaczy ich istotę | – analizuje obie metody Euklidesa pod kątem wydajności i szybkości działania dla różnych zestawów zmiennych wejściowych | – samodzielnie przeprowadza analizę wydajności algorytmu Euklidesa dla różnych danych i przewiduje wyniki swojej analizy |
| 17 | Ciąg Fibonacciego | – wie kim był i kiedy żył Fibonacci;<br>– zna zasługi Fibonacciego dla rozwoju cywilizacji. | – na podstawie podręcznika lub innych wiarygodnych źródeł omawia na gotowym przykładzie istotę ciągu Fibonacciego;<br>– na podstawie podręcznika lub innych | – samodzielnie omawia na gotowym przykładzie istotę ciągu Fibonacciego;<br>– samodzielnie analizuje działanie przykładowego algorytmu obliczający | – układa program w języku C++ obliczający *n* kolejnych elementów ciągu Fibonacciego;<br>– testuje poprawność działania swojego programu na przykładzie. | – proponuje rozwiązanie problemu obliczania *n*-tego elementu ciągu Fibonacciego. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | wiarygodnych źródeł analizuje działanie przykładowego algorytmu obliczający kolejne elementy ciągu, zapisanego w postaci schematu blokowego. | kolejne elementy ciągu, zapisanego w postaci schematu blokowego;<br>– układa algorytm obliczający określoną liczbę liczb ciągu Fibonacciego;<br>– analizuje program w języku C++ ułożony według przykładowego algorytmu. | | |
| 18 | Sprawdzian | | | | | |
| 19-20 | Funkcje C++ | – zna różnicę pomiędzy działaniem i zastosowaniem procedury i funkcji;<br>– zna budowę funkcji;<br>– wie, na czym polega wywołanie funkcji; | – wywołuje funkcje w treści programu;<br>– analizuje działanie funkcji; | – stosuje unikalne nazwy zmiennych wewnątrz bloku;<br>– prawidłowo układa mało skomplikowane procedury i funkcje wykorzystujące globalne i lokalne zmienne, np. obliczające; | – omawia istotę deklaracji wewnętrznych zmiennych i stałych, typów i podprogramów lokalnych;<br>– układa i wywołuje procedury z parametrami i bez;<br>– prawidłowo układa funkcje wykorzystujące | – samodzielnie planuje użycie funkcji do rozwiązywania problemów informatycznych;<br>– samodzielnie układa procedury i funkcje, stosując zmienne lokalne i globalne; |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | – określa przypadki, w których niezbędne jest zastosowanie procedury lub funkcji; | | – wywołuje funkcjey z parametrami;<br>– prawidłowo i czytelnie dobiera nazwy zmiennych w deklaracjach procedur i funkcji;<br>– wie, jak działa przekazywanie przez wartość;<br>– wie, jak działa przekazywanie przez zmienną; | globalne i lokalne zmienne;<br>– stosuje unikalne nazwy zmiennych wewnątrz bloku;<br>– układa procedury z przekazywaniem parametrów przez wartość lub zmienną; | |
| 21 | Sprawdzian | | | | | |
| 22-23 | Stosowanie tablic w języku C++ | – podaje definicję typu tablicowego i z pomocą nauczyciela podaje różne przykłady zastosowania tablic;<br>– z pomocą nauczyciela analizuje definicję typu tablicowego, wskazując | – zna znaczenie słów kluczowych w definicjach tablic;<br>– określa wartość zmiennej w tablicy za pomocą wartości typu indeksowego;<br>– umie zaplanować tablicę o odpowiednim do rozwiązywanego | – definiuje własne tablice zmiennych różnych typów i rozmiarów;<br>– prawidłowo zastępuje zmienną tablicową osobno występujące zmienne, uzasadniając swoją decyzję i wykazując | – prawidłowo zastępuje zmienną tablicową osobno występujące zmienne, uzasadniając swoją decyzję i wykazując zasadność takiego postępowania;<br>– nadaje wartości początkowe | – samodzielnie opracowuje algorytmy, które wykorzystują tablice, i na ich podstawie układa programy w języku C++;<br>– stosuje tablice wielowymiarowe do grupowania danych, np. wprowadzanych |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | najważniejsze jej elementy;<br>– obrazowo opisuje zastosowanie tablic jedno-, dwu-, trój- i więcej wymiarowych;<br>– graficznie wyjaśnia budowę tablic; | problemu rozmiarze;<br>– samodzielnie definiuje własne typy tablicowe jednowymiarowe; | zasadność takiego postępowania;<br>– podaje przykłady definiowania różnych tablic i objaśnia zasadność przyjęcia takiej konstrukcji zmiennej tablicowej, np. używanych w przykładzie z podręcznika; | zmiennym tablicowym;<br>– samodzielnie modyfikuje zmienne tablicowe, np. poprzez zwiększenie rozmiaru;<br>– analizuje zajęcie pamięci przez daną tablicę;<br>– umie wskazać błąd w deklaracjach tablic zarówno składniowy, jak i logiczny; | imion, nazwisk itp.; |
| 24 | Sortowanie bąbelkowe, sortowanie przez wstawianie | – na podstawie podręcznika lub innych wiarygodnych źródeł omawia budowę tablicy jednowymiarowej;<br>– na podstawie podręcznika lub innych wiarygodnych źródeł omawia | – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu blokowego; | – omawia istotę metody sortowania bąbelkowego;<br>– omawia działanie przykładowego algorytmu opartego o metodę sortowania bąbelkowego; | – układa algorytm sortowania bąbelkowego;<br>– weryfikuje poprawność działania programu na przykładach. | – układa program sortujący metodą bąbelkową w innym języku niż C++ (np. Java). |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | istotę sortowania bąbelkowego. | – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykład sprawdzający poprawność działania algorytmu. | – sprawdza działanie algorytmu na przykładach. | | |
| 25 | Sprawdzian | | | | | |
| 26-27 | Praca z danymi tekstowymi w języku C++ | – definiuje kod ASCII | – omawia znaczenie kodu ASCII<br>– definiuje plagiat i odnosi tę definicję także do rzeczywistości szkolnej, np. do kopiowania prac domowych | – wykorzystuje klawiaturę numeryczną do wprowadzania znaków za pomocą kodów ASCII | – interpretuje przepisy dotyczące plagiatów<br>– wie, czym jest JSA i jakie ma znaczenie w zwalczaniu kopiowania prac naukowych | Nie przewiduje się oceny celującej. |
| 28 | Sprawdzian | | | | | |
| 29 | Szyfr Cezara i szyfr przedstawieniowy | – omawia cele szyfrowania danych i informacji<br>– tłumaczy, na czym polega podstawieniowy sposób szyfrowania informacji | – na przykładzie tabeli tłumaczy metodę przestawieniową i umie zaszyfrować tekst tą metodą<br>– omawia na podstawie rysunku z podręcznika | – wie, na czym polega szyfrowanie szyfrem wieloalfabetowym<br>– tłumaczy potrzebę szyfrowania | – samodzielnie układa algorytm dla szyfru Cezara | – samodzielnie układa program komputerowy szyfrujący szyfrem Cezara |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | – wie, jak odróżnić strony internetowe z szyfrowaną transmisją danych od pozostałych | metodę szyfrowania szyfrem Cezara | niektórych transmisji w sieci | | |
| 30 | Zamiana liczb między systemami pozycyjnymi | – charakteryzuje liczby szesnastkowe i dwójkowe<br>– wie, jak są zbudowane pozycyjne systemy liczbowe | – na przykładzie z podręcznika omawia działanie algorytmów zamiany postaci liczby dziesiętnej na dwójkową, dziesiętnej na szesnastkową i dwójkowej na szesnastkową<br>– na przykładzie z podręcznika omawia działanie funkcji zamieniających postaci liczb | – na podstawie opisu i algorytmów z podręcznika układa funkcje zamieniające prezentację liczb z dziesiętnej na dwojkową i szesnastkową oraz z dwojkowejna szesnastkową | – samodzielnie układa programy (funkcje) zamieniające prezentacje liczb<br>– układa programy wykorzystujące funkcje zamieniające prezentacje liczb | – samodzielnie układa uniwersalne programy oparte o funkcje zamieniające prezentacje liczb na wiele postaci |
| 31 | Iteracja<br><br>w algorytmach | – omawia na prawdziwych przykładach iterację | – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje przykładowe algorytmy iteracyjne i rekurencyjne; | – samodzielnie analizuje przykładowe algorytmy iteracyjne i rekurencyjne;<br>– samodzielnie rozpoznaje | – układa algorytmy z podejściem iteracyjnym i rekurencyjnym i układa na ich podstawie programy;<br>– wskazuje instrukcje, które | – rozwiązuje problemy obiema metodami i ocenia ich skuteczność. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | – na podstawie podręcznika lub innych wiarygodnych źródeł rozpoznaje procesy iteracyjne. | procesy iteracyjne;<br>– wskazuje w przykładowych algorytmach miejsca, które decydują o iteracyjności lub rekurencyjności opisywanego procesu;<br>– analizuje przykładowy program. | decydują o iteracyjności podejścia do realizacji algorytmu. | |