**Wymagania edukacyjne z informatyki – klasa 3 liceum – zakres podstawowy**

**Migra – Język programowania C++**

**Egzamin teoretyczny i praktyczny – test (moduł A) + zadania do wykonania (baza danych Access)**

**Prace zaliczeniowe do przygotowania w domu:**

1. Przygotuj program w języku C++ z wykorzystaniem algorytmu na tekście. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
2. Przygotuj program w języku C++ z wykorzystaniem szyfrowania. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
3. Przygotuj program w języku C++ z zastosowaniem algorytmu porządkowania. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
4. Przygotuj program w języku C++ z zastosowaniem rekurencji. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
5. Przygotuj projekt małego ogrodu lub swojego pokoju w programie SketchUp (wersja online). Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.

Programy nie mogą być identyczne jak w podręczniku!

Do programowania w języku C++ pobierz z internetu program Dev C++ Lub Code Blocks.

**Prace należy przesłać co najmniej tydzień przed egzaminem z pośrednictwem platformy Teams. Zostanie tam utworzone specjalne zadanie.**

| **Poziom wymagań** | | | | |
| --- | --- | --- | --- | --- |
| **ocena**  **dopuszczająca** | **ocena**  **dostateczna** | **ocena**  **dobra** | **ocena**  **bardzo dobra** | **ocena**  **celująca** |
| **Moduł A. Wokół komputera i sieci komputerowych** | | | | |
| **Reprezentacja danych w komputerze** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| zna pojęcie systemu pozycyjnego;  wie, co to jest system binarny;  analizuje gotowy przykład z podręcznika obliczający wartość dziesiętną liczby zapisanej w systemie dwójkowym i na tej podstawie wykonuje podobne ćwiczenie | definiuje pojęcie systemu pozycyjnego;  wie na czym polega działanie procesora;  potrafi dokonać konwersji liczby między systemem dziesiętnym a dwójkowym oraz dwójkowym a dziesiętnym | 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 | zna elementy uproszczonego modelu komputera zgodny z ideą von Neumanna;  dokonuje konwersji liczb między systemem szesnastkowym i binarnym | 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 |
| **Wybrane urządzenia cyfrowe** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| wymienia urządzenia cyfrowe wykorzystywane w szkole podczas zajęć (np. drukarka, drukarka 3D, tablica interaktywna, monitor, kamera);  podaje nazwy urządzeń cyfrowych wykorzystywane w domu i poza nim (np. płyta grzejna, okap kuchenny, odtwarzacze audio, system multiroom, system nawigacji, smartwatch) | omawia funkcje poznanych urządzeń używanych w szkole oraz w domu i poza nim;  potrafi zaprezentować w klasie wybrane urządzenie cyfrowe i omówić jego działanie | korzysta z wyszukiwarki internetowej celem opracowania informacji na temat wybranego urządzenia cyfrowego;  z pomocą nauczyciela przygotowuje model 3D do druku 3D, korzystając z odpowiednego oprogramowania  uruchamia drukarkę 3D i wykonuje przykładowy wydruk (lub omawia sposób drukowania – w przypadku braku drukarki w szkole) | objaśnia funkcje poznanych urządzeń używanych w domu i poza nim;  zna podstawowe możliwościami oprogramowania towarzyszącego wybranemu urządzeniu, np. drukarce 3D i przygotowuje model 3ZD do wydruku;  samodzielnie potrafi uruchomić drukarkę 3D i przygotować i wykonuje przykładowy wydruk (w przypadku, gdy szkoła ma takie możliwości) | wymienia parametry techniczne urządzeń cyfrowych podanych w specyfikacji technicznej;  potrafi posługiwać się instrukcją obsługi urządzeń cyfrowych i poznawać samodzielnie możliwości towarzyszącego im oprogramowania |
| **Moduł B. Wokół dokumentów komputerowych** | | | | |
| **Tworzenie baz danych** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| wyjaśnia, na czym polega przetwarzanie danych;  definiuje pojęcie baza danych;  na przykładzie gotowego pliku bazy danych potrafi omówić jej strukturę – określić, jakie informacje są w niej pamiętane | podaje obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, instytucji naukowych, społecznych i gospodarczych;  wyjaśnia pojęcia: baza danych, rekord i pole;  rozumie organizację danych w relacyjnych bazach danych;  potrafi przygotować schemat prostej relacyjnej bazy danych;  tworzy prostą bazę danych, składającą się z dwóch tabel: planuje zawartość tabel;  stosuje zasady tworzenia tabel | omawia etapy przygotowania bazy danych;  określa odpowiednio typy danych;  rozumie pojęcia relacji i klucza podstawowego;  przygotowuje projekt formularza i raportu;  tworzy tabele i korzysta z Widoku projektu | rozumie, co oznacza przetwarzanie danych w bazach danych;  definiuje relacje między tabelami;  potrafi uzasadnić, dlaczego warto umieszczać dane w kilku tabelach połączonych relacją;  podczas rozwiązywania nowego problemu korzysta z doświadczeń zdobytych przy rozwiązaniu innego, podobnego problemu | analizuje problemy występujące w utworzonej bazie danych i znajduje ich rozwiązanie;  samodzielnie przygotowuje projekt bazy danych (składającej z trzech tabel) i potrafi ją wykonać w programie do tworzenia baz danych |
| **Tworzenie i stosowanie formularzy w relacyjnej bazie danych** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| wymienia poznane obiekty bazy danych: tabele, formularze, zapytania, raporty; wskazuje je, korzystając z gotowej bazy danych;  korzysta z gotowych formularzy, wprowadzając przykładowe dane | potrafi dokonać modyfikacji wyglądu formularza;  planuje i tworzy nowe formularze zgodnie z treścią ćwiczenia;  korzysta z kreatora formularzy;  wie na czym polega sortowanie danych;  na podstawie przygotowanych formularzy ćwiczy wprowadzanie i aktualizację danych | potrafi wykonać sortowanie rekordów według wybranego pola;  wie, czym różni się formularz standardowy od formularza z podformularzem;  wie, jak utworzyć formularz z podformularzem;  umie zaimportować dane z arkusza kalkulacyjnego do bazy danych | planuje i tworzy formularz z podformularzem) zgodnie z treścią ćwiczenia | samodzielnie projektuje wygląd formularzy, tworzy je i modyfikuje, korzystając z zaawansowanych możliwości modyfikacji formularzy, np. zmieniając różne własności |
| **Wykonywanie podstawowych operacji na relacyjnej bazie danych** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| zna zastosowanie filtrów do wyszukiwania danych;  potrafi wyświetlić wynik gotowego zapytania i omówić, czego zapytanie dotyczy;  wie do czego służą raporty;  wyświetla gotowy raport i omawia, na podstawie jakich pól został utworzony;  omawia zastosowanie korespondencji seryjnej | definiuje pojęcie kwerendy;  tworzy kwerendę wybierającą w Widoku projektu;  przygotowuje raporty do wydruku;  zna sposób przygotowania korespondencji seryjnej z wykorzystaniem danych z bazy danych | modyfikuje gotowe zapytania;  tworzy kwerendę parametryczną;  potrafi utworzyć raport na podstawie kwerendy;  umieszcza w korespondencji seryjnej pola z tabeli bazy danych;  korzysta z gotowych szablonów listów seryjnych | samodzielnie modyfikuje i tworzy kwerendy oraz raporty;  wie, jaka jest korelacja między edytorem tekstu a bazą danych podczas tworzenia korespondencji seryjnej;  w edytorze tekstu przygotowuje listy seryjne i etykiety adresowe, korzystając z danych zapisanych w bazie danych | potrafi zaprojektować samodzielnie relacyjna bazę danych (składającą się z trzech tabel), ustala typy pól, projektuje wygląd formularzy;  potrafi budować złożone kwerendy z dwóch lub więcej tabel połączonych;  planuje i projektuje raporty;  opracowuje własny szablon listu seryjnego |
| **Projektowanie modeli dwuwymiarowych i trójwymiarowych** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| wyjaśnia różnice między grafiką 2D i 3D;  wymienia nazwy programów do tworzenia grafiki 3D;  planuje kroki wykonania projektu ogrodu;  korzysta z podstawowych narzędzi programu SketchUp | potrafi zainstalować i skonfigurować program SketchUp;  realizuje przekształcenie modelu 2D w 3D;  korzysta z dodatkowych narzędzi programu SketchUp;  umieszcza gotowe elementy z biblioteki | tworzy obiekty z zachowaniem odpowiedniej skali;  przekształca pliki graficzne;  korzysta z dodatkowych narzędzi programu SketchUp jak Offset, Pull/Push, Orbit;  przekształca pliki graficzne, uwzględniając wielkość i jakość obrazów | wykonuje modele trójwymiarowe z zachowaniem skali i wytycznych;  tworzy modele, wykorzystując różne możliwości programu | zapoznaje się z możliwościami wybranego programu graficznego, korzystając z Pomocy i innych źródeł;  przygotowuje złożone projekty z różnych dziedzin;  uczestniczy w konkursach dotyczących grafiki komputerowej |
| **Moduł C. Wokół algorytmiki i programowania** | | | | |
| **Algorytmy na tekstach w językach C++ i Python** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| wie, że w edytorach tekstu wykorzystywane są algorytmy na tekstach – pokazuje przykłady wyszukiwania znaków w tekście, porównywania tekstów;  potrafi omówić, posługując się przykładami i pomocami dydaktycznymi, wybrany algorytm na tekstach;  analizuje gotowy program wykorzystujący dane tekstowe i objaśnia stosowanie zmiennych tekstowych;  testuje gotowe programy dla różnych danych | planuje kolejne kroki rozwiązania problemu porównywania tekstów, szukając rozwiązania;  deklaruje zmienne typu tekstowego char i string;  realizuje algorytm porównania dwóch tekstów;  wie, na czym polega algorytm szukania wzorca w tekście;  wie, jak odwoływać się do pojedynczego znaku łańcucha,  wie, jak wyznaczyć długość łańcucha – potrafi zastosować funkcję length() oraz strlen() (C++) i funkcję len() (Python);  analizuje funkcję realizującą np. algorytm porównywania tekstów i omawia działanie funkcji w tym zastosowane instrukcje; tworzy program wykorzystujący tę funkcję | przeprowadza analizę rozwiązania algorytmu szukania znaku w tekście;  szuka wystąpień wzorca w tekście metodą naiwną, analizując i uzupełniając kolejne kroki algorytmu z wykorzystaniem podręcznika;  definiuje pojęcie konkatencji;  formułuje treść zadania do przedstawionego kodu źródłowego;  definiuje funkcję szukającą i zliczającą wystąpienia znaków lub ciągów znaków w tekście;  korzysta z instrukcji if…elif (Python) do utworzenia menu programu | opracowuje program zliczający wystąpienie znaku w tekście;  korzysta z funkcji wyboru switch() (C++) do utworzenia menu programu;  umieszcza w kodzie źródłowym funkcję getline() do wprowadzania napisów składających się z wyrazów oddzielonych spacjami | tworzy zaawansowane programy wykorzystujące dane tekstowe i poznane funkcje;  potrafi samodzielnie utworzyć algorytm i program realizujący porównywanie tekstów;  tworzy rozbudowane menu wyboru z wykorzystaniem funkcji switch() (C++);  tworzy rozbudowane menu wyboru z wykorzystaniem z instrukcji if … elif (Python) |
| **Algorytmy szyfrowania** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| wie, czym jest szyfrowanie danych i w jakim celu się je stosuje;  potrafi, korzystając z przykładu z podręcznika, przeanalizować prosty przykład szyfrowania;  wie, jak się tworzy anagramy i podaje przykłady anagramów | zna pojęcia: szyfr, szyfrowanie, deszyfrowanie;  omawia schemat procesu szyfrowania i deszyfrowania;  zna przynajmniej jeden algorytm szyfrowania danych – szyfr Cezara;  korzystając z podręcznika szyfruje i deszyfruje wiadomość, korzystając z szyfru Cezara;  przedstawia sposób utworzenia anagramu;  analizuje i rozumie działanie funkcji (podanych w podręczniku) realizujących wybrany algorytm szyfrowania | wie, czym zajmuje się kryptologia i kryptoanaliza;  zna zasady programowania algorytmu szyfrowania przedstawieniowego  stosuje szyfrowanie tekstu metodą Cezara i przestawieniową;  pisze program w wybranym języku programowania tworzący anagramy;  tworzy program realizujący algorytm szyfrowania szyfrem Cezara;  formułuje algorytm deszfrowania podstawieniowego i pisze funkcję realizująca ten algorytm | pisze program realizujący algorytm szyfrowania przedstawieniowego ( opracowuje odpowiednie funkcje pomocnicze);  pisze funkcję deszyfrowania przestawieniowego;  objaśnia szyfrowanie symetryczne i asymetryczne | omawia dziedziny gospodarki, w których wykorzystywane jest szyfrowanie danych;  opracowuje rozbudowane anagramy i potrafi je zaprogramować;  wyszukuje dodatkowe informacje na temat szyfrowania danych;  omawia, czym się zajmuje stenografia, samodzielnie wyszukując informacje na ten temat |
| **Metoda połowienia, podejście zachłanne i rekurencja** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| objaśnia na przykładzie algorytm wyszukiwania przez połowienie;  wie, na czym polega metoda zachłanna – podaje przykłady jej stosowania, wykonując proste ćwiczenia z podręcznika (np. problem umieszczenia książek w samochodzie);  podaje przykłady zjawisk rekurencyjnych | analizuje algorytm wyszukiwania przez połowienie;  potrafi dokonać analizy algorytmu wydawania reszty metodą zachłanną;  pisze program realizujący algorytm obliczania silni ( w wersji iteracyjnej);  zna rekurencyjną definicję funkcji obliczającej silnię liczby naturalnej | korzystając z opisów w podręczniku, definiuje funkcję wyszukiwania liczby w zbiorze uporządkowanym metoda połowienia;  korzystając z podręcznika, definiuje funkcję realizującą algorytm wydawania reszty metodą zachłanną w wybranym języku programowania;  zapisuje w postaci programu rekurencyjną realizację algorytmu obliczającego silnię liczby naturalnej;  testuje programy dla różnych danych | pisze program realizujący algorytm wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia;  pisze program realizujący algorytm zachłanny wydawania reszty z wykorzystaniem odpowiedniej funkcji;  potrafi zapisać w postaci programu rekurencyjną realizację algorytmu obliczającego liczby Fibonacciego;  rozumie różnicę między rekurencją a iteracją | rozumie dokładnie technikę rekurencji (znaczenie stosu);  potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję;  samodzielnie formułuje problem, do którego rozwiązania można zastosować rekurencję;  zapisuje w postaci programu rekurencyjna wersję poznanego wcześniej algorytmu (np. algorytmu Euklidesa);  korzysta samodzielnie z dodatkowej literatury fachowej |
| **Algorytmy na liczbach** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| przeprowadza konwersję liczby z systemu dwójkowego na dziesiętny oraz z dziesiętnego na dwójkowy (bazując na przykładach z tematu A1);  wyjaśnia, na czym polega wyznaczanie NWD i NWW | korzystając z fragmentu programów z podręcznika, pisze programy w wybranym języku programowania (C++, Python): obliczający wartość dziesiętną liczby dwójkowej i wyznaczający rozwinięcie dwójkowe liczby dziesiętnej;  testuje programy dla różnych danych;  opracowuje funkcję w języku programowania wyznaczającą NWD;  opracowuje funkcję w  języku programowania wyznaczającą NWW | analizuje algorytm zapisywania liczby dziesiętnej w systemie liczbowym o określonej podstawie (listę kroków i schemat blokowy), wyróżniając podproblemy;  testuje działanie algorytmów: obliczania wartości dziesiętnej liczby zapisanej w dowolnym systemie i zapisywania liczby dziesiętnej w systemie liczbowym o określonej podstawie, korzystając z odpowiednich list kroków (z podręcznika);  wie, jak można reprezentować ułamek zwykły w języku C++ (deklaruje rekordy za pomocą słowa kluczowego struct) lub w języku Python (omawia przykład klasy i tworzy zmienną danej klasy);  zna wzory na obliczanie sumy ułamków zwykłych oraz mnożenie i dzielenie ułamków zwykłych | na podstawie list kroków (podanych w podręczniku) tworzy programy: obliczania wartości dziesiętnej liczby zapisanej w dowolnym systemie i zapisujący liczbę dziesiętną w systemie liczbowym o określonej podstawie;  potrafi zastosować schemat Hornera do obliczenia wartości wielomianu;  analizuje funkcje skracającą ułamki zwykłe i dodająca ułamki zwykłe oraz stosuje je do napisania programów: skracającego ułamki zwykłe oraz dodającego ułamki zwykłe;  definiuje funkcje: odejmującą, mnożącą oraz dzielącą ułamki zwykłe; wywołuje funkcje w programie głównym i testuje programy dla różnych danych | pisze program wykonujący konwersję liczb zapisanych w dowolnych podstawach;  pisze program realizujący rekurencyjny algorytm zamiany liczby dziesiętnej na postać binarną;  bierze udział w konkursach informatycznych i/lub olimpiadzie informatycznej |
|  |  |  |  |  |
| **Moduł D. Wokół Internetu i projektów** | | | | |
| **Elementy robotyki – projekty** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| wie czym zajmuje się robotyka;  potrafi scharakteryzować funkcje mikrokontrolera Arduino  zna etapy tworzenia projektu grupowego i wykonuje proste zadania szczegółowe z projektu grupowego | zna podstawową strukturę programu dla mikrokontrolera Arduino;  potrafi nazwać moduły składające się na system sterowania nawadnianiem ogrodu;  wykonuje przydzielone zadania szczegółowe | na podstawie wytycznych wykonuje układ elektroniczny oraz pisze program obsługujący system „podlewania ogrodu”;  prawidłowo zapisuje, przechowuje i udostępnia dokumenty potrzebne do realizacji projektu;  prezentuje efekty wspólnej pracy | rozszerza system „podlewania ogrodu”, zgodnie z opisem w podręczniku;  konstruuje robota, stosując elementy elektroniczne oraz programuje jego funkcje;  pełni rolę koordynatora projektu grupowego, m.in.: określa i przydziela zadania szczegółowe;  scala elementy projektu wykonane przez członków grupy w jeden projekt | samodzielnie tworzy programy sterujące mikrokontrolerem;  wykonuje wybrany inny system, np. system „Wycisz radio” po otwarciu drzwi do pokoju;  proponuje tematykę własnego projektu, samodzielnie wyznacza zadania szczegółowe i sposób ich realizacji |
| **Więcej na temat tworzenia stron internetowych – projekt** | | | | |
| **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** | **Uczeń:** |
| wie, czym są blogi; potrafi znaleźć blog o wybranej tematyce;  zna najważniejsze narzędzia do tworzenia stron internetowych;  wie na czym polega tworzenie strony internetowej;  wykonuje proste zadania szczegółowe z projektu grupowego;  przestrzega zasad korzystania z cudzych materiałów | wie, jak założyć blog;  wie, czym jest system zarządzania treścią;  omawia etapy tworzenia strony internetowej;  uczestniczy w przygotowaniu projektu graficznego strony internetowej;  wie, jak ustalić tło strony internetowej i uzyskać efekt po najechaniu myszą;  wykonuje przydzielone zadania szczegółowe;  prezentuje efekty wspólnej pracy | potrafi założyć prosty blog o wybranej tematyce; korzysta z szablonów do tworzenia stron;  przygotowuje projekt graficzny strony internetowej;  potrafi ustawić listy w wierszach i kolumnach na stronie internetowej;  omawia wybrane atrybuty CSS i podaje przykłady ich stosowania;  prawidłowo zapisuje, przechowuje i udostępnia dokumenty potrzebne do realizacji projektu | tworzy stronę internetową wzbogaconą o dodatkowe elementy;  potrafi tworzyć przyciski na stronie internetowej z elementów listy poprzez dodanie obramowania i innych atrybutów;  stosuje wybrane atrybuty CSS;  testuje stronę internetową, określając czy, projekt został wykonany zgodnie ze specyfikacją;  pełni rolę koordynatora projektu grupowego;  przydziela zadania szczegółowe;  scala dokumenty wykonane przez członków grupy | wykonuje samodzielnie projekt graficzny strony internetowej na wybrany przez siebie temat;  tworzy własną stronę internetową wzbogaconą o dodatkowe elementy, w tym tabelami, listami, elementy dynamiczne; posługuje się arkuszem stylów; publikuje stronę w Internecie;  proponuje tematykę własnego projektu, samodzielnie wyznacza zadania szczegółowe i sposób ich realizacji |