

**CONTINUTURI PROGRAMĂ**  
**RECOMANDATE PENTRU EXAMENE DE DIFERENȚĂ**

*Programe școlare în vigoare: (<http://programe.ise.ro/#>) – Ordin nr. 5099/09.09.2009*

- Matematică informatică
- Matematica informatică intensive informatică
- Științele naturii

*Clasa a IX-a – Matematică informatică / Științele naturii*

Unitatea de învățare	Conținuturi
<b>Algoritmi</b>	<p>Etapele rezolvării problemelor. Exemple            Noțiunea de algoritm. Caracteristici. Exemple.            Date cu care lucrează algoritmi (constante, variabile, expresii).            Operații asupra datelor (aritmetice, logice, relaționale).</p>
<b>Limbajul pseudocod</b>	<p>Reprezentarea algoritmilor. Pseudocod.            Principiile programării structurate. Structuri de bază:</p> <ul style="list-style-type: none"> <li>• structura liniară</li> <li>• structura alternativă</li> <li>• structura repetitivă</li> </ul>
<b>Algoritmi elementari</b>	<p>Prelucrarea numerelor :</p> <ul style="list-style-type: none"> <li>• prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, testarea proprietății de palindrom etc.)</li> <li>• probleme de divizibilitate (de exemplu, determinarea divizorilor unui număr, determinarea c.m.m.d.c./c.m.m.m.c., testare primalitate etc.)</li> </ul> <p>calculul unor expresii simple (sume, produse etc.)            Prelucrarea unor secvențe de valori</p> <ul style="list-style-type: none"> <li>• determinare minim/maxim</li> <li>• verificarea unei proprietăți (de exemplu, dacă toate elementele din secvență sunt numere perfecte etc.)</li> <li>• calculul unor expresii în care intervin valori din secvență (de exemplu: numărarea elementelor pare/impare etc)</li> <li>• generarea șirurilor recurente (de exemplu: șirul Fibonacci)</li> </ul>
<b>Aplicații interdisciplinare</b>	<p>Aplicații interdisciplinare (specifice profilului)  <i>Exemple:</i></p> <ul style="list-style-type: none"> <li>• Rezolvarea ecuației de gradul I și de gradul al II-lea</li> <li>• Simplificarea fracțiilor</li> <li>• Aplicații geometrice (distanța dintre două puncte, aria/perimetrul unui triunghi, volumul corpurilor regulate etc.)</li> </ul> <p>Analiza eficienței unui algoritm.            Exemplificări de modalități de implementare a algoritmilor studiați.</p>

*Clasa a X-a – Matematică informatică/ Științele naturii*

Unități de învățare	Conținuturi
<p><b>Elemente de bază și mediul de programare al limbajului C/C++</b></p>	<p><b>Mediul limbajului de programare studiat</b></p> <ul style="list-style-type: none"> <li>• Prezentare generală</li> <li>• Editarea programelor sursă</li> <li>• Compilare, rulare, depanare</li> </ul> <p><b>Noțiuni introductive</b></p> <ul style="list-style-type: none"> <li>• Structura programelor</li> <li>• Vocabularul limbajului</li> <li>• Tipuri simple de date (standard)</li> <li>• Constante, variabile, expresii</li> <li>• Citirea/scrierea datelor</li> </ul> <p><b>Structuri de control</b></p> <ul style="list-style-type: none"> <li>• Structura liniară</li> <li>• Structura alternativă</li> <li>• Structuri repetitive</li> </ul> <p><b>Implementarea unor algoritmi elementari cu aplicabilitate practică</b></p> <ul style="list-style-type: none"> <li>- Algoritmi structura alternativa</li> <li>- Algoritmi operatii pe cifre</li> <li>- Algoritmi divizorii unui numar</li> <li>- Descompunere in factori primi</li> <li>- Cmmdc, cmmmc</li> </ul>
<p><b>Tipuri structurate de date</b></p>	<p>Fișiere text</p> <ul style="list-style-type: none"> <li>• Definiție.</li> <li>• Operații specifice</li> </ul> <p>Tipul tablou. Tablouri unidimensionale și bidimensionale.</p>
<p><b>Algoritmi fundamentali de prelucrare a datelor structurate în tablouri</b></p>	<p>Căutare secvențială, căutare binară</p> <p>Sortare</p> <p>Interclasare</p> <p>Prelucrări specifice tablourilor bidimensionale</p>
<p><b>Aplicații interdisciplinare și analiza eficienței unui algoritm</b></p>	<p><b>Aplicații interdisciplinare</b></p> <p>Exemple orientative:</p> <ul style="list-style-type: none"> <li>• Calculul valorii unei expresii algebrice</li> <li>• Calcule combinatoriale</li> <li>• Analiza eficienței unui algoritm</li> </ul>

*Clasa a XI-a – Matematică informatică*

Unitatea de învățare	Conținuturi
<b>Șiruri de caractere</b>	Particularități de memorare a șirurilor de caractere: Subprograme predefinite de prelucrare a șirurilor de caractere.
<b>Structuri de date neomogene (struct/record)</b>	Rezolvarea unor probleme cu caracter practic: <ul style="list-style-type: none"> <li>• declararea unei variabile de tip structură neomogenă</li> <li>• exemplificarea cazurilor în care se utilizează structuri de date neomogenă</li> <li>• accesarea valorilor din câmpurile unei înregistrări</li> <li>• prelucrarea unei înregistrări/structuri la nivel de câmp și la nivel de structură</li> </ul>
<b>Subprograme</b>	Structura și a modul de definire al subprogramelor Declararea și apelul subprogramelor Variabile locale și globale Transferul parametrilor la apel (prin valoare și referință) Returnarea valorilor de către subprograme Aplicații folosind subprograme
<b>Recursivitate</b>	Definire. Exemplificare Mecanisme de implementare Aplicații cu subprograme recursive
<b>Metoda de programare „Divide et Impera”</b>	Prezentare generală Aplicații
<b>Metoda de programare “Backtracking”</b>	Prezentare generală Implementarea iterativă sau recursivă a algoritmilor de generarea produsului cartezian, permutărilor, combinațiilor, aranjamentelor, submulțimilor unei mulțimi
<b>Grafuri</b>	Terminologie și proprietăți (graf orientat și neorientat, adiacență, incidență, grad; lanț, lanț elementar, drum, drum elementar, ciclu, ciclu elementar, circuit, circuit elementar; subgraf, graf parțial; conexitate, tare conexitate, arbore, arbore parțial, etc) Reprezentarea în memorie a grafurilor (matrice de adiacență, liste de adiacență, lista muchiilor/ arcelor) Parcurgerea grafurilor. Aplicații.

*Clasa a IX-a – Matematică informatică intensiv informatică*

Unități de învățare	Conținuturi
Elaborarea algoritmilor de rezolvare a problemelor și implementarea lor într-un limbaj de programare	<p>Etapele rezolvării problemelor.                      Noțiunea de algoritm. Caracteristici.                      Reprezentarea algoritmilor în pseudocod                      Principiile programării structurate.                      Structuri de bază: structura liniară, structura alternativă, structura repetitivă.                      Algoritmi elementari.                      1. Prelucrarea numerelor:</p> <ul style="list-style-type: none"> <li>• prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, inversul unui număr, testarea proprietății de palindrom, etc.)</li> <li>• probleme de divizibilitate (de exemplu, determinarea divizorilor unui număr, determinarea c.m.m.d.c./c.m.m.m.c., testare primalitate, descompunere în factori primi, etc.)</li> <li>• calculul unor expresii simple (sume, produse, etc.)</li> </ul> <p>2. Prelucrarea unor secvențe de valori:</p> <ul style="list-style-type: none"> <li>• determinare minim/maxim</li> <li>• verificarea unei proprietăți (de exemplu, dacă toate elementele din secvență sunt numere perfecte, etc.)</li> <li>• calculul unor expresii în care intervin valori din secvență (de exemplu: numărarea elementelor pare/impare, etc.)</li> <li>• generarea șirurilor recurente (de exemplu: șirul Fibonacci, progresii aritmetice și geometrice)</li> </ul>
	<p><b>Elementele de bază ale limbajului de programare</b>  <b>Noțiuni introductive</b>                      Structura programelor                      Vocabularul limbajului                      Tipuri simple de date (standard)                      Constante, variabile, expresii                      Citirea/scrierea datelor                      Reprezentarea algoritmilor într-un limbaj de programare                      Structuri de control implementate în limbajul de programare.</p>
Fișiere text	<p>Definire, operații specifice</p> <ul style="list-style-type: none"> <li>• citirea și afișarea datelor folosind fișiere text</li> </ul>
Tablouri unidimensionale	<p>Algoritmi fundamentali de prelucrare a datelor structurate în tablouri</p> <ul style="list-style-type: none"> <li>• parcurgerea tablourilor unidimensionale</li> <li>• interschimbarea, deplasarea, ștergerea și inserarea de elemente</li> <li>• operații cu mulțimi</li> <li>• căutare secvențială, căutare binară</li> <li>• sortare</li> <li>• interclasare</li> <li>• secvențe și subșiruri</li> </ul>
Tablouri bidimensionale	<ul style="list-style-type: none"> <li>• parcurgerea tablourilor bidimensionale pe linii/coloane</li> <li>• tablouri bidimensionale pătratice, diagonale</li> </ul>

Unitatea de învățare	Conținuturi
<b>Limbaj C++</b>  <b>Subprograme</b>	<ul style="list-style-type: none"> <li>• Declararea, definierea și apelul subprogramelor</li> <li>• Transferul parametrilor la apel</li> <li>• Returnarea valorilor de către subprograme</li> <li>• Variabile locale și globale</li> <li>• Subprograme de tip funcție</li> <li>• Subprograme de tip procedură</li> <li>• Transferul parametrilor prin valoare și referință</li> <li>• Modularizarea unui program prin intermediul subprogramelor</li> </ul>
<b>Tipuri structurate de date – șiruri de caractere</b>	<ul style="list-style-type: none"> <li>• Șir de caractere</li> <li>• Funcții standard la nivel de caracter și la nivel de structură</li> <li>• Algoritmi fundamentali: prelucrarea unui șir de caractere la nivel de caracter și la nivel de structură, utilizând funcții specifice</li> <li>• Aplicații la parcurgere și prelucrare a șirurilor de caractere</li> <li>• Prelucrarea unui șir de caractere utilizând funcții predefinite</li> </ul>
<b>Tipuri structurate de date – tipul înregistrare</b>	<ul style="list-style-type: none"> <li>• Tipuri structurate de date                             <ul style="list-style-type: none"> <li>- Înregistrare (structură)</li> </ul> </li> <li>• Algoritmi fundamentali                             <ul style="list-style-type: none"> <li>- prelucrarea unei înregistrări/ structuri la nivel de câmp și la nivel de structură</li> </ul> </li> </ul>
<b>Tipuri structurate de date – lista, stiva, coada</b>	<ul style="list-style-type: none"> <li>• Tipuri structurate de date:                             <ul style="list-style-type: none"> <li>- lista, stiva, coada</li> <li>- operații specifice</li> </ul> </li> <li>• Aplicații care implementează liste liniare cu alocare statică</li> <li>• Implementarea operațiilor specifice stivei și cozii</li> </ul>
<b>Subprograme recursive</b>	<ul style="list-style-type: none"> <li>• Mecanismul de realizare a recursivității</li> <li>• Compararea implementării recursive, a unui algoritm, cu cea iterativă, avantaje și dezavantaje ale celor două tipuri de implementări.</li> <li>• Algoritmi elementari implementați recursiv</li> </ul>
<b>Metoda de programare Divide et Impera</b>	<ul style="list-style-type: none"> <li>• Algoritmi elementari implementați folosind metoda Divide et Impera: minim, maxim, cmmdc</li> <li>• Căutarea eficientă a unui element într-o mulțime ordonată aplicând metoda Divide et Impera (căutarea binară)</li> <li>• Aplicații de căutare eficientă a unui element într-o mulțime ordonată</li> <li>• Sortarea eficientă a unei mulțimi de valori aplicând metoda Divide et Impera (sortarea rapidă, sortarea prin interclasare). Implementarea algoritmilor de sortare. Aplicații care utilizează sortarea rapidă, sortarea prin interclasare</li> </ul>

Unitatea de învățare	Conținuturi
<p><b>TEHNICI DE PROGRAMARE- Backtracking</b></p>	<ul style="list-style-type: none"> <li>• Descrierea metodei, implementarea metodei, aplicații: problema celor <math>n</math> dame, problema colorării hărților.</li> <li>• Aplicații ale metodei în combinatorică: generarea permutărilor, combinărilor, aranjamentelor, funcțiilor surjective, partițiilor unui număr, produsul cartezian a <math>n</math> mulțimi, submulțimile unei mulțimi, generarea tuturor partițiilor unei mulțimi.</li> <li>• Generarea tuturor posibilităților de a ieși dintr-un labirint, problema bilei, săritura calului pe table de șah.</li> </ul>
<p><b>TEHNICI DE PROGRAMARE – Metoda Greedy</b></p>	<ul style="list-style-type: none"> <li>• Descrierea metodei, implementarea metodei, aplicații.</li> <li>• Probleme pentru care metoda Greedy conduce la soluția optimă: suma maximă, problema planificării spectacolelor, problema rucsacului (cazul continuu).</li> <li>• Greedy euristic: plata unei sume cu număr minim de bancnote, săritura calului, problema comis-voiajorului.</li> </ul>
<p><b>TEHNICI DE PROGRAMARE – Metoda programării dinamice</b></p>	<ul style="list-style-type: none"> <li>• Descrierea metodei, implementare, aplicații: <ul style="list-style-type: none"> <li>○ problema sumei în triunghi;</li> <li>○ subșir crescător de lungime maximă;</li> <li>○ subșir comun maximal;</li> <li>○ problema rucsacului (cazul discret);</li> </ul> </li> </ul>
<p><b>EFICIENȚA ALGORITMILOR</b></p>	<ul style="list-style-type: none"> <li>• Analiza complexității unui algoritm</li> <li>• Tipuri de complexitate : <math>O(n)</math>, <math>O(n \cdot \log n)</math>, <math>O(n^2)</math>, <math>O(2n)</math>.</li> <li>• Compararea metodelor de rezolvare a unei probleme din punctul de vedere a eficienței.</li> <li>• Rezolvarea unor probleme cu caracter practic folosind metoda cea mai eficientă.</li> </ul>
<p><b>Noțiuni teoretice - GRAFURI NEORIENTATE</b></p>	<ul style="list-style-type: none"> <li>• Graf neorientat, adiacență, incidență, grad al unui nod.</li> <li>• Reprezentarea în memorie a grafurilor neorientate (matrice de adiacență, liste de adiacență, lista muchiilor, matricea costurilor).</li> <li>• Graf parțial și subgraf, lanț și ciclu, componente conexe.</li> <li>• Tipuri speciale de grafuri (graf complet, graf hamiltonian, graf eulerian, graf bipartit).</li> </ul>
<p><b>Noțiuni teoretice - GRAFURI ORIENTATE</b></p>	<ul style="list-style-type: none"> <li>• Graf orientat, adiacență, incidență, grad interior, grad exterior.</li> <li>• Reprezentarea în memorie a grafurilor orientate (matrice de adiacență, liste de adiacență, lista arcelor).</li> <li>• Drumuri și circuite, componente tare conexe în grafuri orientate.</li> <li>• Graf turneu.</li> </ul>
<p><b>ALGORITMI DE PRELUCRARE A GRAFURILOR</b></p>	<ul style="list-style-type: none"> <li>• Parcurgerea grafurilor în lățime și în adâncime.</li> <li>• Determinarea componentelor conexe ale unui graf neorientat.</li> <li>• Determinarea componentelor tare conexe ale unui graf orientat.</li> <li>• Determinarea matricei lanțurilor/drumurilor.</li> <li>• Determinarea drumurilor de cost minim într-un graf (algoritmul lui Dijkstra, algoritmul Roy-Floyd).</li> <li>• Arbori parțiali de cost minim (algoritmul lui Kruskal sau algoritmul lui Prim).</li> </ul>

Unitatea de învățare	Conținuturi
<b>STRUCTURI DE DATE ARBORESCENTE</b>	<ul style="list-style-type: none"> <li>• Arbori cu rădăcină (definiție, proprietăți, reprezentare cu referințe ascendente, reprezentare cu referințe descendente)</li> <li>• Arbori binari (definiție, proprietăți specifice; reprezentarea arborilor binari cu referințe descendente; operații specifice)</li> </ul>
<b>TIPURI SPECIALE DE ARBORI BINARI</b>	<ul style="list-style-type: none"> <li>• Arbore binar complet – definiție, proprietăți, reprezentare secvențială;</li> <li>• Arbore binar de căutare – definiție, proprietăți, operații specifice;</li> <li>• Heap-uri – definiție, proprietăți, operații specifice.</li> </ul>
<b>ELEMENTE DE PROGRAMARE ORIENTATĂ PE OBIECTE</b>	<ul style="list-style-type: none"> <li>• Principiile programării orientate pe obiecte</li> <li>• Clase și obiecte (definire, utilizare, operații specifice)</li> <li>• Moștenire și polimorfism.</li> </ul>