CONŢINUTURI PROGRAMĂ
RECOMANDATE PENTRU EXAMENE DE DIFERENŢĂ

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 învăţare	Conţinuturi
Algoritmi	Etapele rezolvării problemelor. Exemple Noţiunea de algoritm. Caracteristici. Exemple. Date cu care lucrează algoritmii (constante, variabile, expresii). Operaţii asupra datelor (aritmetice, logice, relaţionale).
Limbajul pseudocod	Reprezentarea algoritmilor. Pseudocod. Principiile programării structurate. Structuri de bază: • structura liniară • structura alternativă • structura repetitivă
Algoritmi elementari	Prelucrarea numerelor : • prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, testarea proprietăţii de palindrom etc.) • probleme de divizibilitate (de exemplu, determinarea divizorilor unui număr, determinarea c.m.m.d.c./c.m.m.m.c., testare primalitate etc.) calculul unor expresii simple (sume, produse etc.) Prelucrarea unor secvenţe de valori • determinare minim/maxim • verificarea unei proprietăţi (de exemplu, dacă toate elementele din secvenţă sunt numere perfecte etc.) • calculul unor expresii în care intervin valori din secvenţă (de exemplu: numărarea elementelor pare/impare etc) • generarea şirurilor recurente (de exemplu: şirul Fibonacci)
Aplicaţii interdisciplinare	Aplicaţii interdisciplinare (specifice profilului) *Exemple:* • Rezolvarea ecuaţiei de gradul I şi de gradul al II-lea • Simplificarea fracţiilor • Aplicaţii geometrice (distanţa dintre două puncte, aria/perimetrul unui triunghi, volumul corpurilor regulate etc.) Analiza eficienţei unui algoritm. Exemplificări de modalităţi de implementare a algoritmilor studiaţi.

Clasa a X-a – Matematică informatică/ Științele naturii

Unităţi de învăţare	Conţinuturi
Elemente de bază şi mediul de programare al limbajului C/C++	**Mediul limbajului de programare studiat** • Prezentare generală • Editarea programelor sursă • Compilare, rulare, depanare **Noţiuni introductive** • Structura programelor • Vocabularul limbajului • Tipuri simple de date (standard) • Constante, variabile, expresii • Citirea/scrierea datelor **Structuri de control** • Structura liniară • Structura alternativă • Structuri repetitive **Implementarea unor algoritmi elementari cu aplicabilitate practică** - **Algoritmi structura alternativa** - **Algoritmi operatii pe cifre** - **Algoritmi divizorii unui numar** - **Descompunere in factori primi** - **Cmmdc, cmmmc**
Tipuri structurate de date	Fişiere text • Definire. • Operaţii specifice Tipul tablou. Tablouri unidimensionale şi bidimensionale.
Algoritmi fundamentali de prelucrare a datelor structurate în tablouri	Căutare secvenţială, căutare binară Sortare Interclasare Prelucrări specifice tablourilor bidimensionale
Aplicaţii interdisciplinare şi analiza eficienţei unui algoritm	**Aplicaţii interdisciplinare** Exemple orientative: • Calculul valorii unei expresii algebrice • Calcule combinatoriale • **Analiza eficienţei unui algoritm**

Clasa a XI-a – Matematică informatică

Unitatea de învăţare	Conţinuturi
Şiruri de caractere	Particularităţi de memorare a şirurilor de caractere: Subprograme predefinite de prelucrare a şirurilor de caractere.
Structuri de date neomogene (struct/record)	Rezolvarea unor probleme cu caracter practic: • declararea unei variabile de tip structură neomogenă • exemplificarea cazurilor în care se utilizează structuri de date neomogenă • accesarea valorilor din câmpurile unei înregistrări • prelucrarea unei înregistrări/structuri la nivel de câmp şi la nivel de structură
Subprograme	Structura şi a modul de definire al subprogramelor Declararea şi apelul subprogramelor Variabile locale şi globale Transferul parametrilor la apel (prin valoare şi referinţă) Returnarea valorilor de către subprograme Aplicaţii folosind subprograme
Recursivitate	Definire. Exemplificare Mecanisme de implementare Aplicaţii cu subprograme recursive
Metoda de programare „Divide et Impera”	Prezentare generală Aplicaţii
Metoda de programare “Backtracking”	Prezentare generală Implementarea iterativă sau recursivă a algoritmilor de generarea produsului cartezian, permutărilor, combinărilor, aranjamentelor, submulţimilor unei mulţimi
Grafuri	Terminologie şi proprietăţi (graf orientat şi neorientat, adiacenţă, incidenţă, grad; lanţ, lanţ elementar, drum, drum elementar, ciclu,ciclu elementar, circuit, circuit elementar; subgraf, graf parţial; conexitate, tare conexitate, arbore, arbore parţial, etc) Reprezentarea în memorie a grafurilor (matrice de adiacenţă, liste de adiacenţă, lista muchiilor/ arcelor) Parcurgerea grafurilor. Aplicaţii.

Clasa a IX-a – Matematică informatică intensiv informatică

Unităţi de învăţare	Conţinuturi
Elaborarea algoritmilor de rezolvare a problemelor şi implementarea lor într-un limbaj de programare	Etapele rezolvării problemelor. Noţiunea de algoritm. Caracteristici. Reprezentarea algoritmilor în pseudocod Principiile programării structurate. Structuri de bază: structura liniară, structura alternativă, structura repetitivă. Algoritmi elementari. 1. Prelucrarea numerelor: • prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, inversul unui număr, testarea proprietăţii de palindrom, etc.) • 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.) • calculul unor expresii simple (sume, produse, etc.) 2. Prelucrarea unor secvenţe de valori: • determinare minim/maxim • verificarea unei proprietăţi (de exemplu, dacă toate elementele din secvenţă sunt numere perfecte, etc.) • calculul unor expresii în care intervin valori din secvenţă (de exemplu: numărarea elementelor pare/impare, etc.) • generarea şirurilor recurente (de exemplu: şirul Fibonacci, progresii aritmetice şi geometrice)
	Elementele de bază ale limbajului de programare **Noţiuni introductive** 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.
Fişiere text	Definire, operaţii specifice • citirea şi afişarea datelor folosind fişiere text
Tablouri unidimensionale	Algoritmi fundamentali de prelucrare a datelor structurate în tablouri • parcurgerea tablourilor unidimensionale • interschimbarea, deplasarea, ştergerea şi inserarea de elemente • operaţii cu mulţimi • căutare secvenţială, căutare binară • sortare • interclasare • secvenţe şi subşiruri
Tablouri bidimensionale	• parcurgerea tablourilor bidimensionale pe linii/coloane • tablouri bidimensionale pătratice, diagonale

Clasa a X-a – Matematică informatică intensiv informatică

Unitatea de învăţare	Conţinuturi
Limbaj C++ **Subprograme**	• Declararea, definirea şi apelul subprogramelor • Transferul parametrilor la apel • Returnarea valorilor de către subprograme • Variabile locale şi globale • Subprograme de tip funcţie • Subprograme de tip procedură • Transferul parametrilor prin valoare şi referinţă • Modularizarea unui program prin intermediul subprogramelor
Tipuri structurate de date – şiruri de caractere	• Şir de caractere • Funcţii standard la nivel de caracter şi la nivel de structură • Algoritmi fundamentali:prelucrarea unui şir de caractere la nivel de caracter şi la nivel de structură, utilizând funcţii specifice • Aplicaţii la parcurgere şi prelucrare a şirurilor de caractere • Prelucrarea unui şir de caractere utilizând funcţii predefinite
Tipuri structurate de date – tipul inregistrare	• Tipuri structurate de date - Înregistrare (structură) • Algoritmi fundamentali - prelucrarea unei înregistrări/ structuri la nivel de câmp şi la nivel de structură
Tipuri structurate de date – lista, stiva, coada	• Tipuri structurate de date: - lista, stiva, coada - operaţii specifice • Aplicaţii care implementează liste liniare cu alocare statică • Implementarea operaţiilor specifice stivei şi cozii
Subprograme recursive	• Mecanismul de realizare a recursivităţii • Compararea implementării recursive, a unui algoritm, cu cea iterativă, avantaje şi dezavantaje ale celor două tipuri de implementări. • Algoritmi elementari implementaţi recursiv
Metoda de programare Divide et Impera	• Algoritmi elementari implementaţi folosind metoda Divide et Impera: minim, maxim, cmmdc • Căutarea eficientă a unui element într-o mulţime ordonată aplicând metoda Divide et Impera (căutarea binară) • Aplicaţii de căutare eficientă a unui element într-o mulţime ordonată • Sortarea eficientă a unei mulţimi de valori aplicând metoda Divide et Impera (sortarea rapidă, sortarea prin interclasare). Implementarea algoritmilor de sortare. Aplicaţii care utilizează sortarea rapidă, sortarea prin interclasare

Clasa a XI-a – Matematică informatică intensiv informatică

Unitatea de învăţare	Conţinuturi
TEHNICI DE PROGRAMARE- Backtracking	• Descrierea metodei, implementarea metodei, aplicaţii: problema celor n dame, problema colorării hărţilor. • Aplicaţii ale metodei în combinatorică: generarea permutărilor, combinărilor, aranjamentelor, funcţiilor surjective, partiţiilor unui număr, produsul cartezian a n mulţimi, submulţimile unei mulţimi, generarea tuturor partiţiilor unei mulţimi. • Generarea tuturor posibilităţilor de a ieşi dintr-un labirint, problema bilei, săritura calului pe table de şah.
TEHNICI DE PROGRAMARE – Metoda Greedy	• Descrierea metodei, implementarea metodei, aplicaţii. • Probleme pentru care metoda Greedy conduce la soluţia optimă: suma maximă, problema planificării spectacolelor, problema rucsacului (cazul continuu). • Greedy euristic: plata unei sume cu număr minim de bancnote, săritura calului, problema comis-voiajorului.
TEHNICI DE PROGRAMARE – Metoda programării dinamice	• Descrierea metodei, implementare, aplicaţii: o problema sumei în triunghi; o subşir crescător de lungime maximă; o subşir comun maximal; o problema rucsacului (cazul discret);
EFICIENŢA ALGORITMILOR	• Analiza complexităţii unui algoritm • Tipuri de complexitate : O(n), O(n*logn), O(n2), O(2n). • Compararea metodelor de rezolvare a unei probleme din punctul de vedere a eficienţei. • Rezolvarea unor probleme cu caracter practic folosind metoda cea mai eficientă.
Noţiuni teoretice - GRAFURI NEORIENTATE	• Graf neorientat, adiacenţă, incidenţă, grad al unui nod. • Reprezentarea în memorie a grafurilor neorientate (matrice de adiacenţă, liste de adiacenţă, lista muchiilor, matricea costurilor). • Graf parţial şi subgraf, lanţ şi ciclu, componente conexe. • Tipuri speciale de grafuri (graf complet, graf hamiltonian, graf eulerian, graf bipartit).
Noţiuni teoretice - GRAFURI ORIENTATE	• Graf orientat, adiacenţă, incidenţă, grad interior, grad exterior. • Reprezentarea în memorie a grafurilor orientate (matrice de adiacenţă, liste de adiacenţă, lista arcelor). • Drumuri şi circuite, componente tare conexe în grafuri orientate. • Graf turneu.
ALGORITMI DE PRELUCRARE A GRAFURILOR	• Parcurgerea grafurilor în lăţime şi în adâncime. • Determinarea componentelor conexe ale unui graf neorientat. • Determinarea componentelor tare conexe ale unui graf orientat. • Determinarea matricei lanţurilor/drumurilor. • Determinarea drumurilor de cost minim într-un graf (algoritmul lui Dijkstra, algoritmul Roy-Floyd). • Arbori parţiali de cost minim (algoritmul lui Kruskal sau algoritmul lui Prim).
STRUCTURI DE DATE	• Arbori cu rădăcină (definiţie, proprietăţi, reprezentare cu referinţe ascendente, reprezentare cu referinţe descendente)

Unitatea de învăţare	Conținuturi
ARBORESCENTE	• Arbori binari (definiţie, proprietăţi specifice; reprezentarea arborilor binari cu referinţe descendente; operaţii specifice)
TIPURI SPECIALE DE ARBORI BINARI	• Arbore binar complet – definiţie, proprietăţi, reprezentare secvenţială; • Arbore binar de căutare – definiţie, proprietăţi, operaţii specifice; • Heap-uri – definiţie, proprietăţi, operaţii specifice.
ELEMENTE DE PROGRAMARE ORIENTATĂ PE OBIECTE	• Principiile programării orientate pe obiecte • Clase şi obiecte (definire, utilizare, operaţii specifice) • Moştenire şi polimorfism.