

Prefață

Atunci când am pornit la drum în proiectarea conținutului acestei cărți, mai întâi a trebuit să răspundem la o mulțime de întrebări: cui se adresează această carte, care sunt obiectivele pe care le urmărim, atât din punct de vedere al conținutului cât și al aspectelor didactice...

Este important ca de la început să clarificăm un aspect important: nu este vorba de un manual obișnuit care, pe baza unei programe școlare, oferă cunoștințe acoperitoare, intră în detalii care sunt necesare elevilor pentru a înțelege și a dobândi concepte fundamentale. Această carte se adresează acelor copii care sunt doritori de cunoștințe suplimentare, doresc să învețe mai repede și să atingă performanțe în informatică, cu precădere în algoritmică și programare. În consecință, cunoștințele teoretice sunt prezentate sumar, sunt „orientate” spre scopul clar de a crea deprinderile necesare pentru a „ataca” cât mai repede problemele propuse spre rezolvare din domeniu.

Învățământul de informatică din România a cunoscut multe succese. Absolvenții fostelor licee de informatică (înființate în 1972) azi sunt profesori universitari, cercetători, conducători de firme, cu un cuvânt: oameni de succes. Acest lucru se datorează și calității învățământului din aceste școli, dar și faptului că informatica îi atrăgea pe cei mai dotați tineri. Este la fel și azi, dar învățământul de informatică, pe de o parte s-a extins (în sensul numărului de copii care învață informatică în România), pe de altă parte s-a redus (a scăzut numărul de ore pe săptămână, alocate disciplinei de la 8 la 3). Este clar că nu se poate lucra „performanță” în 3 ore cu toți copiii, așa că a apărut necesitatea de a oferi celor pasionați și doritori de performanță un cadru suplimentar, unde se pot ocupa cu ceea ce îi interesează.

Elevii din România cărora le place programarea pot participa la o serie de concursuri, simpozioane, tabere unde pot învăța și pot arăta ceea ce știu. Olimpiadele de informatică sunt cu precădere de algoritmică și programare. Prima condiție pe care un elev trebuie să o îndeplinească pentru a avea succes la aceste olimpiade este să fie în stare să se gândească singur, să se concentreze și să aibă curaj pentru implementarea unei soluții originale. Fiecare proiect de algoritm, fiecare program este o creație nouă și trebuie să îndeplinească toate cerințele și restricțiile impuse în specificațiile proble-

melor de rezolvat. Pentru a ajunge în situația respectivă, este nevoie de multe exerciții, de multe antrenamente. Aceste abilități se pot forma doar rezolvând multe, foarte multe probleme.

În concluzie, cartea este mai mult o culegere de probleme decât un manual. În concepția autorilor, ea totuși trebuie parcursă în succesiunea capitolelor, așa cum sunt ele înșiruite în carte. Recomandăm acest lucru deoarece, cartea nefiind o versiune îmbogățită a unui manual elaborat conform programei școlare, cunoștințele din carte depășesc conținutul acestei programe, este nevoie de o abordare treptată a problemelor. Accentuăm, de asemenea, că dacă unii elevi au deja cunoștințe de programare (din gimnaziu, de la Clubul Copiilor etc.), ei nu trebuie constrânși, de exemplu, să nu folosească subprograme anterior capitolului în care se prezintă acest concept. Altfel spus, orice pas „înainte” este permis, chiar binevenit, cu condiția ca rezolvitorii să respecte specificațiile problemei.

Profesorul este sfătuit să lucreze cu elevii ținând cont de întrebările și de răspunsurile lor, să-și modeleze activitatea în funcție de nivelul grupului. Dacă activitatea decurge fără probleme, evident se poate avansa repede și se pot studia și probleme din alte surse sau chiar din manualele care constituie continuarea celui de față. Se știe că la olimpiade „nu prea” se ține cont de programa școlară, iar la olimpiadele internaționale există o singură categorie de vârstă: 14-18 ani.

Rezolvările problemelor propuse sunt prezentate sub formă de explicații și eventual pseudocodul algoritmului. Textele sursă le veți putea consulta în software-ul care însoțește aceste manuale. Nu avem pretenția că întotdeauna am propus cea mai elegantă și cea mai eficientă rezolvare. Dacă ne contactați și veți fi de acord ca soluțiile interesante și originale ale dumneavoastră să devină publice, acestea vor fi înglobate în software și vor fi adăugate noilor ediții ale manualelor.

Vă dorim succes!

Autorii

Programa școlară

1. Generalități

- *Conținutul următoarelor programe (dezvoltate pentru doritorii de cunoștințe solide în programare) nu precizează limbajul de programare. În funcție de cerințele elevilor și disponibilitățile profesorilor toate temele se pot trata realizând aplicații în limbajul **Pascal** și/sau **C++**. De asemenea, se vor face aplicații și în **FreePascal**, respectiv **gnu C**. Evident, nu sunt excluse nici mediile vizuale (Delphi, Visual C++, VisualBasic sau Java).*
- *Conținutul programei este grupat în 32 de săptămâni pe parcursul cărora aproximativ din patru în patru săptămâni se vor organiza „miniconcursuri” din domeniul (domeniile) de cunoștințe abordate deja. Elevii vor lucra în condiții similare viitoarelor concursuri la care vor participa. Evaluarea se va face automat, utilizând software-ul de evaluare care va fi folosit și în celelalte săptămâni pentru a verifica lucrările elevilor și a evalua modul în care ei avansează.*

2. Obiective generale

Elevii participanți vor fi capabili:

- să analizeze enunțuri de probleme care depășesc nivelul programei școlare;
- să aleagă tipul datelor, să creeze structuri de date adecvate în scopul modelării datelor de intrare și de ieșire;
- să proiecteze algoritmi de rezolvare pentru probleme;
- să implementeze algoritmi proiectați utilizând un limbaj de programare;
- să prelucreze date structurate la nivel de componente și la nivel de structură;
- să lucreze cu date organizate în fișiere;
- să lucreze cu subprograme;
- să aleagă algoritmul de ordonare, respectiv de căutare adecvat;
- să folosească operatorii pe biți în vederea optimizării programelor;
- să prelucreze numere de orice dimensiune;
- să lucreze recursiv în situația în care acest lucru este necesar și nu afectează timpul de executare a programelor;
- să rezolve probleme aplicând metoda backtracking.

3. Gruparea temelor pe săptămâni

Săptămânile 1-2: Recapitularea cunoștințelor acumulate în gimnaziu

Obiective specifice:

Acei elevi care au participat la activitățile centrului de excelență la informatică în gimnaziu sau au fost atrași spre concursuri de programare vor efectua o recapitulare sub conducerea profesorului. Dar întotdeauna există și elevi care iau contact cu algoritmi și cu programarea doar în clasa a 9-a. Activitatea se va desfășura în așa manieră încât începătorii să-i ajungă din urmă pe cei avansați cât se poate de repede. În acest scop, pentru acești elevi, conținutul capitolului de recapitulare se va eșalona pe trei-patru săptămâni. Elevii capabili vor rezolva problemele propuse în acest capitol pentru a fi evaluați și grupați în subgrupe în funcție de nivelul de cunoștințe.

- Structura programelor Pascal, Vocabularul limbajului
- Constante, tipuri standard de date, variabile, expresii
- Citirea și scrierea datelor
- Instrucțiuni
 1. Probleme pentru începători: Învățarea alfabetului, Ghiocel și viorele, Serviciul pe clasă, tăierea copacilor, La scăldat, Drumul spre școală.
 2. Avansați: Numere prime... cu divizori, Matematica... pe calculator, La pescuit.
 3. Excelenți: Serviciul de gardă, Plasarea oglinzilor, Palindrom, Numere apropiate.

Săptămâna 3: Fișiere

Obiective specifice:

Elevii se vor familiariza cu date stocate pe suport electronic. Se știe că în timpul realizării programelor și în timpul verificării lor, acestea se testează în mod repetat, pentru diverse date de intrare care, de regulă, sunt furnizate în fișiere. De asemenea, datele de ieșire (rezultatele) trebuie păstrate în format electronic. La marea majoritate a concursurilor de programare datele de intrare sunt citite din fișiere, iar cele de ieșire sunt scrise în fișiere, motiv pentru care elevii trebuie să se familiarizeze cu acest aspect încă de la început.

- Detalii generale despre fișiere;
- Asignare, deschidere pentru scriere/citire, închidere.
 4. citirea unui număr dintr-un fișier și scrierea sa în alt fișier;
 5. citirea tuturor numerelor dintr-un fișier care conține câte un număr pe o linie și scrierea lor în alt fișier, câte unul pe o linie;
 6. citirea datelor dintr-un fișier care conține numere, mai multe pe o linie, și scrierea lor într-un alt fișier în aceeași formă;

7. citirea unei secvențe de numere dintr-un fișier și scrierea ei în alt fișier;
8. citirea unor nume (șiruri de caractere) dintr-un fișier și scrierea lor într-un alt fișier;
9. citirea unui număr natural N dintr-un fișier și scrierea într-un alt fișier a tuturor numerelor naturale cuprinse între 0 și N , câte unul pe o linie;
10. citirea unui număr natural N dintr-un fișier și a N nume (aflate pe următoarele linii ale fișierului) și scrierea celor N nume într-un alt fișier;
11. copierea integrală a conținutului unui fișier într-un alt fișier.

Săptămâna 4: Calcule efectuate pe șiruri de date fără păstrarea lor sub formă de tablouri

Obiective specifice:

Elevii vor fi capabili să recunoască clasa de probleme în care, având la dispoziție o secvență de date (care trebuie prelucrate liniar) nu este necesară păstrarea lor sub forma structurilor de date în memorie. De asemenea, vor învăța să aplice „teorema de programare” care prelucrează secvențe de date, care transmite rezultatul unor decizii.

- Inițializarea sumei (produsului) cu elementul neutru al operației;
- Efectuarea unor operații în paralel cu citirea datelor în cazul problemelor care nu necesită păstrarea șirului de date în memorie;
- Realizarea protecției explicite din interiorul programului în cazul unor posibile depășiri, și a unor date care nu corespund specificațiilor din enunț.
 1. citirea unor numere dintr-un fișier și scrierea sumei lor într-un alt fișier;
 2. citirea unor numere dintr-un fișier și scrierea produsului lor într-un alt fișier;
 3. citirea unor numere dintr-un fișier și scrierea și scrierea mediei aritmetice într-un alt fișier;
 4. citirea unor caractere dintr-un fișier (unul pe linie) și scrierea șirului de caractere într-un alt fișier.

Săptămâna 5: Determinarea minimului, respectiv a maximului

Obiective specifice:

Elevii familiarizați cu date stocate pe suport electronic vor învăța cum se redeschide un fișier pentru citire, respectiv pentru scrierea în continuare a unor rezultate în acesta. De asemenea, ei vor fi capabili să rezolve probleme de maxim și de minim, furnizând, dacă problema cere și poziția (pozițiile) valorilor egale cu acestea. Cu această ocazie se familiarizează cu selecția, selectarea tuturor elementelor având o proprietate dată, și cu realizarea optimă a acestora.

- Redeschiderea fișierelor atunci când este necesar;
- Determinarea minimului/maximului dintr-un șir de date;
- Inițializarea minimului/maximului (cu elementul neutru sau cu primul element).

1. citirea unor numere dintr-un fișier și scrierea într-un fișier a minimului lor precum și a primei poziții pe care apare acest minim;
2. citirea unor numere dintr-un fișier și scrierea într-un fișier a minimului lor precum și a ultimei poziții pe care apare acest minim;
3. citirea unor numere dintr-un fișier și scrierea într-un fișier a minimului lor precum și a tuturor pozițiilor pe care apare acest minim;
4. citirea unor numere dintr-un fișier și scrierea într-un fișier a minimului lor precum și a tuturor pozițiilor pe care apare acest minim, urmat de maximul lor și a tuturor pozițiilor pe care apare acest maxim;
5. citirea unor nume dintr-un fișier și afișarea tuturor numelor care, din punct de vedere lexicografic, sunt „mai mari” decât ultimul nume din fișier.

Săptămâna 6: Concurs (I)

Obiective specifice:

Se va testa capacitatea elevilor de a utiliza fișierele pentru citirea și scrierea datelor. Vor fi trei probleme care vor fi rezolvate în două ore.

Săptămâna 7: Numere (I)

Obiective specifice:

Elevii vor fi capabili să prelucreze numere pe baza cifrelor acestora, date sub formă de caractere sau numere și vor învăța să aplice schema lui Horner.

- Descompunere/compunere numere în/din cifre.
 1. afișarea cifrelor unui număr;
 2. determinarea numărului de cifre ale unui număr;
 3. determinarea celei mai mari (mici) cifre a unui număr;
 4. determinarea celei mai mari (mici) cifre a unui număr și a pozițiilor pe care apare aceasta;
 5. determinarea inversului unui număr;
 6. generarea unui număr dintr-un șir de caractere care sunt cifre (protecție în caz de depășire și de caractere care nu sunt cifre);
 7. verificarea faptului că un număr este palindrom;
 8. determinarea tuturor numerelor palindroame dintr-un interval dat;
 9. citirea unui număr real de forma $x.y$ și afișarea numărului $y.x$.

Săptămâna 8: Numere (II)

Obiective specifice:

Elevii vor fi capabili să recunoască problemele care cer stabilirea unei proprietăți al unui număr dat. Răspunsurile date în cazul acestor probleme este bivalent (Da/Nu), din cauza aceasta rezolvările lor se vor testa pe „pachete” de teste. De asemenea, în cazul în care se cere generarea unor numere, având o proprietate cerută, vor exersa

„valorificarea” lor imediată (eventual scriindu-le imediat într-un fișier) fără păstrarea acestora în memorie.

- Verificarea unei proprietăți date a unui număr.
 1. verificarea dacă un număr este par;
 2. numărarea divizorilor unui număr;
 3. descompunerea unui număr în factori primi;
 4. cel mai mare divizor comun a două numere întregi;
 5. verificarea primalității unui număr;
 6. determinarea celui mai mic număr prim mai mare decât un număr dat;
 7. determinarea celui mai mare număr prim mai mic decât un număr dat;
 8. determinarea prefixelor prime ale unui număr;
 9. determinarea sufixelor prime ale unui număr;
 10. verificarea aspectului de „vale” a unui număr;
 11. verificarea aspectului de „munte” a unui număr;
 12. verificarea aspectului de „vale” a unui cuvânt;
 13. verificarea aspectului de „munte” a unui cuvânt;
 14. determinarea subșirului având aspectul „vale”, respectiv „munte” dintr-un șir dat.

Săptămâna 9: Numere (III)

Obiective specifice:

Elevii vor fi capabili să genereze secvențe de numere, având o proprietate dată, fără păstrarea lor în memorie.

- Generarea numerelor având o proprietate cerută
 1. determinarea primelor N numere prime;
 2. determinarea tuturor numerelor prime dintr-un interval;
 3. determinarea tuturor numerelor de tip „vale” dintr-un interval;
 4. determinarea tuturor numerelor de tip „munte” dintr-un interval;
 5. determinarea tuturor numerelor de tip „vale” și a tuturor de tip „munte” dintr-un interval;
 6. generarea termenilor șirului lui Fibonacci;
 7. numere prietene;
 8. alte numere speciale.

Săptămâna 10: Numere (IV)

Obiective specifice:

Multe probleme cer (explicit sau implicit) ca un număr sau numerele date să se convertească într-o altă bază. Aceste operații se vor realiza fără utilizare de tablouri.

- Conversii de numere dintr-o bază în alta
 1. conversie din baza 10 în baza 2;

2. conversie din baza 2 în baza 10;
3. conversie din baza 2 în baza 8 (16);
4. conversie din baza 8 (16) în baza 2;
5. conversie din baza 10 în baza $b < 10$;
6. conversie din baza $b < 10$ în baza 10;
7. conversie din baza $a < 10$ în baza $b < 10$;
8. conversie din baza 16 în baza 10;
9. conversie din baza 10 în baza 16;
10. conversie din baza $a < 36$ în baza $b < 36$;
11. conversie din baza 10 în baza b (oricât de mare);
12. conversie din baza b (oricât de mare) în baza 10;
13. conversie din baza a (oricât de mare) în baza b (oricât de mare).

Săptămâna 11: Concurs (II)

Obiective specifice:

Se va testa capacitatea elevilor de a manipula cifrele unui număr, de a verifica anumite proprietăți ale acestora și de a genera numere cu anumite proprietăți. Vor fi cinci probleme simple care se vor rezolva în două ore.

Săptămâna 12: Șiruri (I)

Obiective specifice:

Elevii vor fi capabili să recunoască acele probleme care necesită declararea unor tablouri unidimensionale, în consecință păstrarea datelor în memorie pe parcursul executării programului. Ei se vor obișnui cu prelucrarea lor. Se vor rezolva probleme care, pe lângă verificarea proprietății cer și returnarea (afișarea) poziției pe care se află acest (aceste) elemente.

- Definirea șirurilor cu ajutorul tablourilor unidimensionale și rezolvarea problemelor definite pentru săptămânile 4 și 5.
 1. determinarea sumei elementelor unui șir;
 2. determinarea minimului și maximului unui șir;
 3. determinarea proprietății unui element dintr-un șir;
 4. căutarea liniară a unui element într-un șir;
 5. căutarea liniară a unui element într-un șir și determinarea tuturor aparițiilor;
 6. căutarea liniară a unui element într-un șir și eliminarea lui;
 7. căutarea liniară tuturor elementelor dintr-un șir și eliminarea lor;
 8. determinarea dublurilor într-un șir și eliminarea lor;
 9. citirea a două șiruri (cu număr diferit de elemente) și determinarea șirului care are suma elementelor maximă/minimă;
 10. citirea a N șiruri (cu număr diferit de elemente) și determinarea șirului care are suma elementelor maximă/minimă;

11. citirea a N șiruri (cu număr diferit de elemente) și identificarea șirului care are cele mai multe elemente.

Săptămâna 13: Șiruri (II) – Ordonări

Obiective specifice:

Ordonarea este una dintre operațiile cele mai frecvente în aplicații. Elevii vor fi capabili, ca în funcție de proprietățile datelor să aleagă metoda de ordonare cea mai potrivită. Metode de ordonare mai performante se vor învăța deodată cu metoda divide et impera, respectiv structuri de date avansate. De asemenea, elevii vor face distincție între verificarea proprietății unui element din șir și proprietatea globală a șirului.

- Ordonarea șirurilor
 1. verificarea faptului că un șir este ordonat;
 2. verificarea faptului că elementele unui șir sunt în progresie aritmetică (geometrică);
 3. ordonare cu metoda bulelor;
 4. sortare prin selecție;
 5. sortare prin inserție;
 6. sortare prin numărare.

Săptămâna 14: Șiruri (III) – Șiruri ordonate

Obiective specifice:

Căutarea este operația care poate fi efectuată cu algoritmi mult mai eficienți dacă datele sunt ordonate sau sunt păstrate în structuri de date de căutare. Elevii vor învăța algoritmul de căutare binară fără a învăța în detalii metoda divide et impera și se vor familiariza cu rezolvările problemelor care cer determinarea celei mai lungi secvențe de date având elemente consecutive de o proprietate dată, folosind algoritmi liniari.

- Căutarea în șiruri ordonate.
 1. căutare binară;
 2. interclasare a două șiruri;
 3. căutare binară cu determinarea extremităților aparițiilor;
 4. inserarea unui element într-un șir ordonat cu păstrarea ordinii;
 5. eliminarea unui element într-un șir ordonat cu păstrarea ordinii;
 6. determinarea dintr-un șir a două subșiruri pe baza unei proprietăți date (creând două subșiruri noi, creând un singur subșir nou, „pe loc”);
 7. determinarea dintr-un șir a mai multor subșiruri pe baza a mai multor proprietăți date;
 8. determinarea celei mai lungi secvențe formată din elemente având o proprietate dată;

9. determinarea celei mai lungi secvențe care este formată din elemente având suma maximă.

Săptămâna 15: Polinoame

Obiective specifice:

Utilizarea tablourilor nu este absolut necesară pentru a prelucra polinoame. Elevii vor fi capabili să implementeze operații pe polinoame.

- Calcule cu polinoame
 1. citirea unui polinom, dat prin grad și coeficienți;
 2. citirea unui polinom, dat prin monoame;
 3. determinarea valorii unui polinom pentru un argument dat;
 4. adunarea a două polinoame;
 5. înmulțirea a două polinoame;
 6. împărțirea a două polinoame;
 7. determinarea celui polinom dintre N polinoame care pentru un argument dat are valoare maximă.

Săptămâna 16: Concurs (III)

Obiective specifice:

Se va testa capacitatea elevilor de a lucra cu șiruri. De asemenea, se va testa capacitatea de a alege algoritmul de sortare, respectiv căutare adecvat, precum și metoda adecvată de stocare a unui polinom. Vor fi trei probleme care vor fi rezolvate în două ore.

Săptămâna 17: Tablouri multidimensionale (I)

Obiective specifice:

Teoria matricelor se învață în clasa a XI-a, deci nu este nevoie ca elevii să fie introduși în capitolul respectiv al matematicii. Tablourile de numere sau de caractere se pot prelucra fără aceste cunoștințe. Elevii se vor obișnui să definească tablouri bidimensionale, să prelucreze elementele acestor tablouri și să acceseze elementele tablourilor bidimensionale, folosind expresii de indici.

- Calcule simple cu elementele unei matrice
 1. suma elementelor unei matrice;
 2. identificarea liniei și coloanei pe care se află minimul/maximul;
 3. suma elementelor de pe diagonala principală;
 4. suma elementelor de pe diagonala secundară;
 5. suma elementelor de deasupra (de dedesubtul) diagonalei principale;
 6. suma elementelor de deasupra (de dedesubtul) diagonalei secundare;
 7. determinarea sumei celor opt vecini ai unui element;

8. determinarea sumelor elementelor aflate în triunghiul de N, S, E și V delimitat de diagonalele matricei.

Săptămâna 18: Tablouri multidimensionale (II)

Obiective specifice:

Tablourile se pot parcurge în multe feluri. Elevii se familiariza cu aceste parcurgeri, formându-și dexteritatea de a se „plimba” într-o matrice bidimensională conform cerințelor.

- Prelucrarea tablourilor bidimensionale.
 1. rotirea cu 90 de grade a unui tablou bidimensional;
 2. determinarea matricei simetrice față de diagonala principală (secundară);
 3. înlocuirea elementelor din triunghiul de nord cu cele din triunghiul de vest (și variante);
 4. parcurgerea pe diagonale a unei matrice;
 5. parcurgerea în spirală a unei matrice;
 6. determinarea poziției punctului „șă” într-o matrice;
 7. calcularea sumelor de-a lungul laturilor matricelor pătratice;
 8. verificarea unei proprietăți globale a unui tablou bidimensional.

Săptămâna 19: Tablouri multidimensionale (III)

Obiective specifice:

Tablourile bidimensionale se compun de șiruri de șiruri. Elevii se vor obișnui să stabilească când anume este avantajos să prelucrez tablouri bidimensionale folosind această proprietate, adică manevrând șiruri.

- Operații cu matrice
 1. produsul a două matrice;
 2. produsul a N matrice;
 3. inversarea a două linii/coloane;
 4. ordonarea liniilor astfel încât elementele de pe diagonala principală/secundară să fie în ordine crescătoare/decrescătoare;
 5. interclasarea a N șiruri ordonate;
 6. verificarea triunghiularității unei matrice (inferior, superior triunghiulară).

Săptămâna 20: Tablouri multidimensionale (IV)

Obiective specifice:

În probleme pot să apară și tablouri cu mai multe dimensiuni decât 2. Elevii își vor forma abilități de a prelucra asemenea tablouri.

- Prelucrarea tablourilor având dimensiune mai mare decât 2
 1. suma elementelor unui tablou n -dimensional;

2. identificarea coordonatelor unui element dat (toate aparițiile) într-un tablou n -dimensional;
3. parcurgerea unui tablou tridimensional dinspre exterior înspre interior;
4. determinarea matricei dintr-un tablou tridimensional care are suma elementelor maximă;

Săptămâna 21: Concurs (IV)

Obiective specifice:

Se va testa capacitatea elevilor de a lucra cu tablouri multidimensionale. Vor fi patru probleme care vor fi rezolvate în două ore.

Săptămâna 22: Articole

Obiective specifice:

Acele date care se compun din mai multe componente având tipuri diferite de multe ori se structurează în articole. Elevii vor învăța cum se declară eficient și cum se prelucreză asemenea date. Ei vor rezolva probleme asemănătoare cu cele din viața cu zi de zi, aplicând facilitățile oferite de mediul de programare ales.

- Definirea unor structuri de date de tip articol (inclusiv cu variante) și prelucrarea lor.
 1. crearea unui catalog pentru o clasă;
 2. afișarea în ordine descrescătoare după medie a elevilor admiși și alfabetică a elevilor respinși la un examen;
 3. afișarea tuturor elevilor care au domiciliul într-un anumit oraș;
 4. ordonarea elevilor în funcție de înălțime și, pentru înălțimi egale, în ordinea greutății;
 5. crearea unui fișier care să conțină articole privind cărțile dintr-o bibliotecă;
 6. crearea unui fișier care să conțină articole privind cărțile dintr-o bibliotecă astfel încât în cazul manualelor să se specifice disciplina, clasa pentru care este valabil manualul, autorii etc., iar pentru cărțile care nu sunt manuale numărul autorilor, autorii, titlul, editura etc. (se va lucra cu înregistrări cu variante);
 7. crearea unui fișier cu articole privind rezultatele participanților la olimpiada internațională de informatică.

Săptămâna 23: Mulțimi și simularea lor

Obiective specifice:

Multe probleme necesită efectuarea operațiilor din teoria mulțimilor. În funcție de mediul de programare și de proprietățile concrete ale datelor, acestea pot fi organizate în mulțimi, sau se pot organiza în alte structuri de date (tablouri) simulând operațiile respective.

- Definirea unor structuri de date de tip mulțime (cu set în Pascal, și fără) și prelucrarea lor
 1. afișarea reuniunii a două mulțimi de numere/litere;
 2. afișarea intersecției a două mulțimi de numere/litere;
 3. afișarea diferenței a două mulțimi de numere/litere;
 4. verificarea apartenenței unui număr (unei litere) la o mulțime dată;
 5. generarea claselor de echivalență pe baza perechilor de numere/caractere care se află într-o relație dată.

Săptămâna 24: Aritmetica numerelor mari

Obiective specifice:

Dacă un număr de un anumit tip depășește cel mai mare număr posibil de reprezentat cu toate cifrele exacte, dar problema necesită un rezultat exact, operațiile aritmetice care trebuie efectuate se vor simula, folosind șiruri de caractere. Aceste prelucrări necesită cunoașterea subprogramelor definite în unit-ul standard al mediului pe string-uri. Elevii vor rezolva probleme în care este necesară implementarea operațiilor aritmetice pe numere mari.

- Aritmetica numerelor mari
 1. implementarea operațiilor cu numere mari (adunare, scădere, înmulțire, împărțire, restul împărțirii întregi, ridicare la putere, radical);
 2. generarea primelor N ($N > 60$) termeni ai șirului lui Fibonacci;
 3. generarea valorii $N!$ ($N > 15$);
 4. calcularea valorii a^b .

Săptămâna 25: Operații pe biți

Obiective specifice:

Elevii se vor familiariza cu utilizarea operatorilor care permit prelucrarea datelor pe biți. În rezolvarea de probleme, utilizarea acestora mărește performanțele programelor.

- Utilizarea operațiilor pe biți în scopul accelerării anumitor operații din algoritmi.
 1. schimbarea semnului unui număr;
 2. schimbarea valorii unui bit dintr-un număr;
 3. codificarea și decodificarea unui număr pe baza unei chei;
 4. interschimbarea a două numere fără variabilă auxiliară, folosind operația **xor**;
 5. înmulțirea și împărțirea cu puteri ale lui 2;
 6. numărarea biților având valoarea 1 (sau 0) din reprezentarea unui număr în baza 2;
 7. descompunerea unui număr întreg în octetul din stânga și octetul din dreapta;
 8. obținerea reprezentării în baza 2 a unui număr întreg, folosind operații pe biți;
 9. rotirea cifrelor binare ale unui număr întreg;

10. schimbarea cifrelor binare egale cu 1 în 0 și invers;
11. generarea tuturor submulțimilor mulțimii $\{1, 2, \dots, N\}$;
12. generarea tuturor șirurilor care conțin numai valorile 0 și 1.

Săptămâna 26: Concurs (V)

Obiective specifice:

Se va testa capacitatea elevilor de a lucra cu tipurile articol și mulțime, precum și capacitatea de a utiliza numerele mari și operațiile pe biți. Vor fi patru probleme care vor fi rezolvate în două ore.

Săptămâna 27: Funcții și proceduri

Obiective specifice:

Elevii au folosit deja multe subprograme predefinite, dar ei trebuie să fie capabili să implementeze rezolvarea unor subprobleme sub formă de subprograme proprii. Ei vor implementa algoritmi de rezolvare sub formă generalizată, astfel încât acestea să se poată reutiliza în rezolvarea mai multor probleme. Va crește lizibilitatea programelor și se va ușura depanarea.

- Utilizarea funcțiilor și procedurilor (predefinite și utilizator)
 1. rezolvarea majorității problemelor tratate, folosind subprograme;
 2. rezolvarea unor probleme în care problema inițială se descompune în subprobleme;
 3. rezolvarea unor probleme în care un același subprogram se apelează în mod repetat pentru date diferite;
 4. realizarea unui unit propriu care conține toate problemele rezolvate cu subprograme și scrierea unor programe care folosesc acest unit.

Săptămâna 28: Recursivitate

Obiective specifice:

Recursivitatea este un concept des întâlnit în programare. Ea totuși trebuie aplicată cu grijă, fără ca folosind-o să se mărească inutil complexitatea algoritmilor. Subprogramele recursive sunt lizibile, compacte și se aplică mai ales asupra structurilor de date definite recursiv sau în cazul problemelor ale căror model matematic este definit în termeni recursivi. Se va reveni la recursivitate în clasa a X-a, când se vor învăța liste și arbori.

- Definirea conceptului și rezolvarea unor probleme enunțate în termeni recursiv.
 1. inversarea cifrelor din configurația unui număr dat (fără string-uri);
 2. inversarea numerelor dintr-un șir dat (fără tablouri);
 3. calculul recursiv al factorialului;
 4. calculul recursiv al celui mai mare divizor comun a două numere întregi;

5. calculul recursiv al termenilor unui șir pe baza unei relații de recurență;
6. generarea submulțimilor mulțimii $\{1, 2, \dots, N\}$;
7. determinarea obiectelor din care se compune o fotografie.

Săptămâna 29: Backtracking (I)

Obiective specifice:

Utilitatea metodei backtracking este „contestată” în literatura de specialitate, totuși, având în vedere că există multe probleme pentru rezolvarea cărora nu există altă metodă mai bună, elevii o vor cunoaște și vor învăța s-o aplice dacă este cazul sau dacă nu găsesc nici o alternativă viabilă. Ei vor deveni capabili să identifice această clasă de probleme și să implementeze rezolvări dezvoltate pe baza metodei. La început se vor trata probleme care prelucrează șiruri și cer furnizarea tuturor soluțiilor.

- Definirea metodei și identificarea clasei de probleme care se rezolvă cu metoda backtracking.
 1. generarea tuturor permutărilor șirului de numere $1, 2, \dots, N$;
 2. așezarea a 8 regine pe tabla de șah, fără ca acestea să se atace;
 3. generarea produsului scalar a două mulțimi;
 4. generarea tuturor aranjamentelor de N luate câte M ;
 5. generarea tuturor perechilor de paranteze care se închid corect;
 6. generarea tuturor funcțiilor injective cunoscând mulțimea de argumente și cea a valorilor funcțiilor;
 7. generarea tuturor funcțiilor surjective cunoscând mulțimea de argumente și cea a valorilor funcțiilor;
 8. generarea tuturor partițiilor unui număr dat;
 9. generarea partițiilor mulțimii $\{1, 2, \dots, N\}$;
 10. generarea tuturor turnurilor de cuburi în funcție de anumite proprietăți date.

Săptămâna 30: Backtracking (II)

Obiective specifice:

Elevii vor fi capabili să identifice clasa de probleme care se rezolvă cu metoda backtracking pe date definite în două dimensiuni.

- Definirea metodei și identificarea clasei de probleme care se rezolvă cu metoda backtracking în două dimensiuni
 1. ieșirea din labirint;
 2. labirint cu pereții celulelor păstrați pe biți;
 3. aranjarea unor piese de *tetris* astfel încât se completează integral un dreptunghi de dimensiuni date.

Săptămâna 31: Backtracking (III)

Obiective specifice:

În cazurile în care s-a hotărât aplicarea metodei pentru probleme care necesită determinarea unui minim, programul poate fi optimizat. Elevii vor fi capabili să identifice clasa de probleme în care asemenea optimizări sunt posibile și vor învăța să le implementeze. De asemenea, ei vor învăța modalități prin care pot să măsoare timpul de executare al propriului program și să-l oprească în momentul în care ar urma ca acesta să ruleze mai mult decât timpul permis. Vor învăța tehnici de optimizare, cum ar fi scrierea în fișier a unor „pachete de rezultate”, generarea unor rezultate ale problemei în timpul lucrului și introducerea acestora în tablouri de constante etc.

- Definirea metodei și identificarea clasei de probleme în care rezolvarea cu metoda backtracking necesită (și permite) optimizări.
 1. determinarea turnului de cuburi de înălțime minimă;
 2. determinarea drumului pe care broasca ajunge la lăcustă și se întoarce la locul de pândă pe suprafața unui lac înghețat unde gheața se rupe în urma efectuării săriturilor;
 3. vizitarea unei matrice de dimensiuni $M \cdot N$ astfel încât se creează un șir A ale cărui elemente sunt numerele din matrice; două elemente aflate pe poziții consecutive în șirul A trebuie să fie vecine în matrice. Suma $1 \cdot A_1 + 2 \cdot A_2 + \dots + M \cdot N \cdot A_{M \cdot N}$ trebuie să fie minimă.

Săptămâna 32: Concurs (VI)

Obiective specifice:

Se va testa capacitatea elevilor de a folosi tehnici recursive și de a rezolva probleme prin metoda backtracking. Vor fi două probleme care vor fi rezolvate în două ore.