

# Proiectul unității de învățare

Clasa X

AN ȘCOLAR 2011 - 2012

Unitatea de învățare: **Subprogramele**

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>Definiția subprogramului;</li> <li>Necesitatea folosirii subprogramelor;</li> <li>Terminologia folosită pentru subprograme;</li> <li>Avantajele folosirii subprogramelor;</li> <li>Identificarea elementelor unui subprogram: prototipul, antetul, definiția subprogramului;</li> <li>Parametrii de comunicare;</li> <li>Clasificarea subprogramelor;</li> <li>Reguli pentru construirea subprogramelor C++;</li> <li><b>Evaluare;</b></li> <li>Reguli pentru construirea subprogramelor C++;</li> <li>Transferul de parametri între subprograme;</li> <li>Clasificarea variabilelor de memorie: durata de viață a variabilelor de memorie, domeniul de vizibilitate al identificatorilor;</li> </ul>	<ul style="list-style-type: none"> <li>Înșusirea regulilor pentru construirea subprograme- lor în limbajul C++;</li> <li>Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator;</li> <li>Construirea unor subpro- grame pentru rezolvarea subproblemelor unei probleme;</li> <li>Formarea deprinderilor de a defini și utiliza subprograme în rezolvarea problemelor;</li> <li>Aplicarea mecanismului recursivității prin crearea unor subprograme recursive (definite de utilizator);</li> <li>Compararea dintre implementarea recursivă și cea iterativă a aceluiasi algoritm;</li> <li>Recunoașterea situațiilor în care este necesară</li> </ul>	<ul style="list-style-type: none"> <li>Explicarea conceptelor referitoare la subprograme;</li> <li>Explicarea elementelor care alcatuiesc un subprograma: antetul subprogramului, corpul subprogramului, prototipul subprogramului, activarea subprogramului, parametrii de comunicare, utilizarea stivei de către subprograme;</li> <li>Descompunerea rezolvării unei probleme în subprobleme;</li> <li>Identificarea unor situații în care alegerea unui algoritm prezintă avantaje în raport cu altul;</li> <li>Descrierea notiunilor referitoare la durata de viață a variabilelor de memorie, domeniul de vizibilitate al identificatorilor;</li> <li>Prezentarea tehnicilor de utilizare a tablourilor in cadrul subprogramelor;</li> </ul>	<ul style="list-style-type: none"> <li>Manualul</li> <li>Culegeri de probleme</li> <li>Calcuatorul</li> <li>Mediul de programare Borland C++</li> </ul>	<ul style="list-style-type: none"> <li>Formativa</li> <li>Teste de evaluare</li> </ul>	

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>Alegerea modului de implementare a subprogramului;</li> <li>Tablourile de memorie și subprogramele;</li> <li>Subprogramele de sistem;</li> <li>Dezvoltarea programelor;</li> <li>Subprograme cu un număr variabil de parametri, supraîncărcarea funcțiilor;</li> <li><b>Evaluare;</b></li> </ul>	<ul style="list-style-type: none"> <li>utilizarea unor subprograme;</li> <li>Analiza problemei în scopul identificării subproblemelor acesteia;</li> </ul>	<ul style="list-style-type: none"> <li>Exersarea creării și aplicării subprogramelor pentru rezolvarea unor probleme întâlnite de elevi în studiul altor discipline școlare;</li> <li>Evidențierea greșelilor tipice în elaborarea algoritmilor;</li> <li>Exersarea definirii și apelării unor programe simple;</li> </ul>			

Unitatea de învățare: **Tipuri structurate de date - Șiruri de caractere**

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>Implementarea șirului de caractere în limbajul C++;</li> <li>Citirea și scrierea șirurilor de caractere;</li> <li>Algoritmi pentru prelucrarea șirurilor de caractere;</li> <li>Prelucrarea a două șirurilor de caractere;</li> <li>Prelucrarea unui șir de caractere;</li> <li>Prelucrarea subșirurilor de caractere;</li> <li>Conversii între tipul șir de caractere și tipuri numerice;</li> <li><b>Evaluare;</b></li> </ul>	<ul style="list-style-type: none"> <li>Însușire tehnicilor de lucru cu șiruri de caractere;</li> <li>Formarea deprinderilor de utilizare a funcțiilor pentru șiruri de caractere;</li> <li>Însușirea operațiilor și algoritmilor specifici șirurilor de caractere;</li> </ul>	<ul style="list-style-type: none"> <li>Explicarea noțiunii de șir de caractere;</li> <li>Descrierea operațiilor care se efectuează cu șiruri de caractere;</li> <li>Prezentarea și explicarea funcțiilor de bibliotecă care permit realizarea de operații cu șiruri de caractere;</li> <li>Combinarea unor prelucrări elementare pentru obținerea anumitor prelucrări complexe în funcție de scopul propus;</li> </ul>	<ul style="list-style-type: none"> <li>Manualul</li> <li>Culegeri de probleme</li> <li>Calcuatorul</li> <li>Mediul de programare Borland C++</li> </ul>	<ul style="list-style-type: none"> <li>Formativa</li> <li>Test de evaluare</li> </ul>	

Unitatea de învățare: **Tipuri structurate de date - Înregistrarea**

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>Implementarea înregistrării în limbajul C++;</li> </ul>	<ul style="list-style-type: none"> <li>Prelucrarea datelor structurate;</li> <li>Însușire tehnicilor de lucru cu înregistrări;</li> </ul>	<ul style="list-style-type: none"> <li>Explicarea noțiunii de înregistrare;</li> <li>Prezentarea tehnicilor de implementare a structurilor;</li> </ul>	<ul style="list-style-type: none"> <li>Manualul</li> <li>Culegeri de probleme</li> <li>Calcuatorul</li> </ul>	<ul style="list-style-type: none"> <li>Formativa</li> <li>Test de evaluare</li> </ul>	

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>▪ Declararea variabilei de tip înregistrare;</li> <li>▪ Accesul la câmpurile înregistrării;</li> <li>▪ Înregistrări imbricate;</li> <li>▪ Tablouri de înregistrări;</li> <li>▪ Înregistrări cu structură variabilă;</li> <li>▪ <b>Evaluare;</b></li> </ul>		<ul style="list-style-type: none"> <li>▪ Descrierea operațiilor care se pot efectua cu variabile de tip struct;</li> <li>▪ Combinarea unor prelucrări elementare pentru obținerea anumitor prelucrări complexe în funcție de scopul propus;</li> </ul>	<ul style="list-style-type: none"> <li>▪ Mediul de programare Borland C++</li> </ul>		

Unitatea de învățare: **Tipuri structurate de date – lista, stiva, coada**

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>▪ Implementarea listelor în limbajul C++;</li> <li>▪ Implementarea și alocarea secvențială;</li> <li>▪ Implementarea prin alocarea înlănțuită;</li> <li>▪ Clasificarea listelor;</li> <li>▪ Algoritmi pentru prelucrarea listelor generale</li> <li>▪ Algoritmi pentru prelucrarea stivelor;</li> <li>▪ Algoritmi pentru prelucrarea cozilor;</li> <li>▪ <b>Evaluare;</b></li> </ul>	<ul style="list-style-type: none"> <li>▪ Înțelegere structurilor de tip de listă și a tehnicilor de alocare;</li> <li>▪ Însușirea operațiilor și algoritmilor specifici listelor ;</li> <li>▪ Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării;</li> <li>▪ Alegerea celui mai eficient algoritm de rezolvare a unei probleme;</li> </ul>	<ul style="list-style-type: none"> <li>▪ Explicarea noțiunii de lista liniară și clasificarea listelor liniare;</li> <li>▪ Descrierea operațiilor relative la liste liniare;</li> <li>▪ Prezentarea algoritmilor care implementează operațiile de la liste liniare: inițializarea listei, adăugarea primului nod, parcurgerea listei, căutarea unui nod în listă, adăugarea unui nod la listă, eliminarea unui nod din listă;</li> <li>▪ Combinarea unor prelucrări elementare pentru obținerea anumitor prelucrări complexe în funcție de scopul propus;</li> </ul>	<ul style="list-style-type: none"> <li>▪ Manualul</li> <li>▪ Culegeri de probleme</li> <li>▪ Calculatoarele</li> <li>▪ Mediul de programare Borland C++</li> </ul>	<ul style="list-style-type: none"> <li>▪ Formativa</li> <li>▪ Teste de evaluare</li> </ul>	

Unitatea de învățare: **Subprograme recursive**

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>▪ Definiția procesului recursiv;</li> <li>▪ Reguli pentru construirea unui subprogram recursiv;</li> <li>▪ Variabilele locale și subprogramele recursive;</li> </ul>	<ul style="list-style-type: none"> <li>▪ Aplicarea mecanismului recursivității prin crearea unor subprograme recursive (definite de utilizator);</li> </ul>	<ul style="list-style-type: none"> <li>▪ Explicarea conceptelor referitoare la subprograme recursive;</li> <li>▪ Descompunerea rezolvării unei probleme în subprobleme;</li> </ul>	<ul style="list-style-type: none"> <li>▪ Manualul</li> <li>▪ Culegeri de probleme</li> <li>▪ Calculatoarele</li> </ul>	<ul style="list-style-type: none"> <li>▪ Formativa</li> <li>▪ Test de evaluare</li> </ul>	

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>Implementarea recursivă a algoritmilor elementari: algoritmul pentru determinarea valorii minime (maxime), algoritmul pentru calculul c.m.m.d.c. a două numere întregi, algoritmi pentru prelucrarea cifrelor unui număr, algoritmul pentru testarea unui număr prim, algoritmul pentru determinarea divizorilor unui număr, algoritmi pentru conversia între baze de numerație;</li> <li>Implementarea recursivă a algoritmilor pentru prelucrarea tablourilor de memorie;</li> <li>Rekursivitatea în cascadă;</li> <li>Rekursivitatea directă și indirectă;</li> <li>Avantajele și dezavantajele recursivității;</li> <li>Evaluarea;</li> </ul>	<ul style="list-style-type: none"> <li>Formarea deprinderilor de a defini și utiliza subprograme recursive în rezolvarea problemelor;</li> <li>Compararea dintre implementarea recursivă și cea iterativă a aceluiași algoritm;</li> <li>Recunoașterea situațiilor în care este necesară utilizarea unor subprograme recursive;</li> <li>Analiza problemei în scopul identificării subproblemelor acesteia;</li> </ul>	<ul style="list-style-type: none"> <li>Identificarea unor situații în care alegerea unui algoritm recursiv prezintă avantaje în raport cu altul iterativ;</li> <li>Exersarea definirii și apelării unor subprograme recursive simple;</li> <li>Proiectarea/modelarea unor algoritmi și implementarea acestora cu ajutorul subprogramelor recursive;</li> <li>Prezentarea tehnicilor de utilizare a tablourilor în cadrul subprogramelor recursive;</li> <li>Exersarea creării și aplicării subprogramelor recursive pentru rezolvarea unor probleme întâlnite de elevi în studiul altor discipline școlare;</li> <li>Evidențierea greșelilor tipice în elaborarea algoritmilor recursivi;</li> </ul>	<ul style="list-style-type: none"> <li>Mediul de programare Borland C++</li> </ul>		

Unitatea de învățare: **Metoda de programare Divide et Impera**

Continuturi	Competente specifice	Activitati de invatare	Resurse	Evaluare	Observatii
<b>Metoda de programare Divide et Impera</b>	<i>Elaborarea algoritmilor de rezolvare a problemelor</i> ✓ Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei	TEHNICI DE PROGRAMARE	Manual Calculator	Aplicatii practice Lucrari scrise	

	<ul style="list-style-type: none"> <li>✓ Aplicarea creativă a metodelor de programare</li> <li>✓ Analiza comparativă a eficienței</li> </ul>				
--	--	--	--	--	--

Unitatea de învățare: **Structuri de date alocate dinamic**

Continuturi	Competente specifice	Activitati de invatare	Resurse	Evaluare	Observatii
<b>Structuri de date alocate dinamic</b> <ul style="list-style-type: none"> <li>✓ Liste simplu înlănțuite</li> <li>✓ Liste dublu înlănțuite</li> <li>✓ Liste circulare</li> </ul>	<b>Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora</b> <ul style="list-style-type: none"> <li>✓ Descrierea operațiilor specifice listelor înlănțuite și elaborarea unor subprograme care să implementeze aceste operații.</li> </ul>	IMPLEMENTAREA STRUCTURILOR DE DATE	Manual Calculator	Aplicatii practice Lucrari scrise	

Unitatea de învățare: **Aplicații interdisciplinare și din viața cotidiană**

Conținuturi	Competențe specifice	Activități de învățare	Resurse	Evaluare	Observații
<ul style="list-style-type: none"> <li>▪ Analiza problemei;</li> <li>▪ Elaborarea modului de rezolvare;</li> <li>▪ Transpunerea în limbajul de programare;</li> <li>▪ Testarea programului;</li> <li>▪ Elaborarea documentației;</li> <li>▪ Evaluare;</li> </ul>	<ul style="list-style-type: none"> <li>▪ Identificarea aplicațiilor informatice în viața socială;</li> <li>▪ Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene;</li> </ul>	<ul style="list-style-type: none"> <li>▪ Analiza, elaborarea și realizarea unor aplicații / proiecte de mai mare anvergură în care să fie implicați și antrenati grupuri de elevi;</li> <li>▪ Testarea și analizarea comportamentului programelor pentru diferite date de intrare;</li> <li>▪ Încurajarea discuțiilor purtate între elevi, exprimarea și ascultarea părerilor fiecăruia.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Manualul</li> <li>▪ Culegeri de probleme</li> <li>▪ Calculatorul</li> <li>▪ Mediul de programare Borland C++</li> </ul>	<ul style="list-style-type: none"> <li>▪ Formativa</li> <li>▪ Evaluarea proiectului realizat</li> </ul>	



DISCIPLINA: Informatică

PROFESOR: Costache Rebeca

CLASA: a X-a D (matematică-informatică), a X-a F (științe ale naturii)

## PLANIFICARE CALENDARISTICĂ

anul școlar 2023 - 2024

*Conform cu programa școlară aprobată prin ordinul ministrului educației, cercetării și inovării 5099 / 09.09.2009*

Unități de învățare	Competențe specifice	Conținuturi	Nr. ore	Data	Observații
<b>Modulul I (7 săptămâni, S1-S7, 11.09.2023 – 27.10.2023)</b>					
		<ul style="list-style-type: none"><li>• Prelucrarea regulamentului de protecția muncii și conduită, specifice laboratorului de informatică</li><li>• Recapitulare și evaluare inițială</li></ul>	1	S1	

Unități de învățare	Competențe specifice	Conținuturi	Nr. ore	Data	Observații
Elemente de bază și mediul de programare al limbajului C++	1.1. Implementarea algoritmilor reprezentați din pseudocod în limbaj de programare	<b>Elementele de bază ale limbajului de programare</b> Noțiuni introductive: <ul style="list-style-type: none"> <li>• Structura programelor</li> <li>• Vocabularul limbajului</li> <li>• Tipuri simple de date (standard)</li> <li>• Constante, variabile</li> <li>• Expresii</li> <li>• Citirea/scrierea datelor</li> </ul>	2	S2-S3	
	1.1. Implementarea algoritmilor reprezentați din pseudocod în limbaj de programare	Mediul limbajului de programare studiat: <ul style="list-style-type: none"> <li>• Prezentare generală</li> <li>• Editarea programelor sursă</li> <li>• Compilare, rulare, depanare</li> </ul>	1	S4	
	1.2. Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)	Structuri de control: <ul style="list-style-type: none"> <li>• Structura liniară</li> </ul>	2	S5-S6	
		• <b>Recapitulare și Evaluare sumativă</b>	1	S7	
<b>Vacanță 28.10.2023 – 05.11.2023</b>					
<b>Modulul al II-lea (6+1 săptămâni, S8-S14, 06.11.2023 – 22.12.2023)</b>					
Elemente de bază și mediul de programare al limbajului C++	1.1. Implementarea algoritmilor reprezentați din pseudocod în limbaj de programare 1.2. Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)	Structuri de control: <ul style="list-style-type: none"> <li>• Structura alternativă</li> <li>• Structuri repetitive</li> </ul>	3	S8-S10	
		<b>Implementarea unor algoritmi elementari cu aplicabilitate practică</b>	2	S11-S12	
		• <b>Recapitulare și Evaluare sumativă</b>	1	S13	
		<b>Școala altfel! (18.12.2023 - 22.12.2023)</b>	1	S14	
<b>Vacanță 23.12.2023 – 07.01.2024</b>					
<b>Modulul al III-lea (6 săptămâni, S15-S20, 08.01.2024 – 16.02.2024)</b>					



Unități de învățare	Competențe specifice	Conținuturi	Nr. ore	Data	Observații
Tipuri structurate de date	2.1. Identificarea necesității structurării datelor în tablouri 2.2. Prelucrarea datelor structurate în tablouri 2.3. Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor	Fișiere text • Definire • Operații specifice	2	S15-S16	
		Tipuri structurate de date. Tipul tablou. Tablouri unidimensionale și bidimensionale	4	S17-S20	
Vacanță 17.02.2024 – 25.02.2024					
Modulul al IV-lea (8+1 săptămâni, S21-S29, 26.02.2024 – 26.04.2024)					
Tipuri structurate de date	3.1. Prelucrarea datelor structurate în tablouri	Algoritmi fundamentali de prelucrare a datelor structurate în tablouri: • sortare • căutare secvențială, căutare binară • interclasare • prelucrări specifice tablourilor bidimensionale	7	S21-S27	
		• Recapitulare și Evaluare sumativă	1	S28	
		Săptămâna verde (22.04.2024 - 26.04.2024)	1	S29	
Vacanță 27.04.2024 – 07.05.2024					
Modulul al V-lea (7 săptămâni, S30-S36, 08.05.2024 – 21.06.2024)					
Aplicații interdisciplinare și din viața cotidiană	4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 4.2. Alegerea unui algoritm eficient de rezolvare a unei probleme	Aplicații interdisciplinare Exemple orientative: • Prelucrări statistice ale unei serii de valori • Operații cu polinoame • Calcule combinatoriale • Determinarea unor mărimi fizice dintr-un circuit electric serie/paralel • Aplicații din genetică (legea creșterilor organice, etc.)	2	S30-S31	
		Analizarea eficienței unui algoritm	1	S32	

Unități de învățare	Competențe specifice	Conținuturi	Nr. ore	Data	Observații
	5.1. Identificarea aplicațiilor informaticii în viața socială 5.2. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene	<b>Aplicații din viața cotidiană</b> Exemple orientative: <ul style="list-style-type: none"> <li>• Determinarea situației școlare a unui elev (medii semestriale, medii generale, numărul de absențe, etc.)</li> <li>• Balanța de cheltuieli ale unei familii</li> <li>• Determinarea salariului unei persoane</li> <li>• Evidența operațiilor într-un cont bancar</li> </ul>	2	S33-S34	
		• <b>Recapitulare și Evaluare sumativă</b>	1	S35	
		• <b>Încheierea situației școlare</b>	1	S36	
<b>Vacanță 22.06.2024 – 31.08.2024</b>					

DISCIPLINA: Informatică  
PROFESOR: Costache Rebeca  
CLASA: a X-a D (matematică-informatică), a X-a F (științe ale naturii)

PLANIFICARE ANUALĂ  
anul școlar 2023 - 2024

DISCIPLINA	Clasa		Nr. de ore săptămână	Nr. de ore anual	Observații
Informatică	X	Nr. de ore din trunchiul comun	1	36	
		Nr. de ore de aprofundare	—	—	
		Nr. de ore de extindere	—	—	
		TOTAL	1	36	

Resurse umane - elevi	Resurse materiale
Nivel de cunoștințe, motivația învățării, atitudini, aptitudini:	Dotarea clasei (laboratorului): - Laborator 1 Informatică&TIC: 27 PC elevi + 1 PC profesor, videoproiector, imprimantă, Whiteboard - Laborator 2 Informatică&TIC: 27 PC elevi + 1 PC profesor, videoproiector, copiator, Whiteboard

Competențe generale	Competențe specifice	Unități de învățare	Modul	Nr. ore	Instrumente de evaluare	Repartizarea orelor pe tipuri de lecție			
						comunicare	formare deprinderi	recapitulare	verificare și evaluare
		Recapitulare și evaluare inițială	I	1	Teste scrise cu itemi obiectivi și semiobiectivi	0	0	0	1
1. Implementarea algoritmilor într-un limbaj de programare	1.1. Implementarea algoritmilor reprezentați din pseudocod în limbaj de programare 1.2. Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)	Elemente de bază și mediul de programare al limbajului C++	I II	12	Teste scrise cu itemi obiectivi și semiobiectivi; Probe practice	3	6	1	2
2. Identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea	2.1. Identificarea necesității structurării datelor în tablouri 2.2. Prelucrarea datelor structurate în tablouri 2.3. Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor	Tipuri structurate de date	III IV	14	Teste scrise cu itemi obiectivi și semiobiectivi; Probe practice	3	9	1	1

3. Elaborarea algoritmilor de rezolvare a problemelor	3.1. Prelucrarea datelor structurate în tablouri								
4. Aplicarea algoritmilor fundamentali în prelucrarea datelor	4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 4.2. Alegerea unui algoritm eficient de rezolvare a unei probleme	Aplicații interdisciplinare și din viața cotidiană	V	6	Teste scrise cu itemi obiectivi și semiobiectivi; Probe practice	1	4	0	1
5. Identificarea conexiunilor dintre informatică și societate	5.1. Identificarea aplicațiilor informaticii în viața socială 5.2. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene								
		Școala altfel!	II	1		0	1	0	0
		Săptămâna verde	IV	1		0	1	0	0
		Încheierea situației școlare	V	1		0	0	0	1

## Proiect didactic

**Disciplina :** Tehnologia informației și a comunicațiilor

**Clasa:** a IX-a A matematica informatica

**Profesor** Costache Rebeca

**Unitatea de învățare:** Procesorul de texte Microsoft Word

**Tema:** Tabele, formatarea și operații în tabele

**Tipul lecției:** Predare de noi cunoștințe

**Locul de desfășurare:** laborator multimedia

**Competențe generale:** Formarea și dezvoltarea deprinderilor de utilizare a unui procesor de texte

**Competențe specifice:** Utilizarea avansată a editorului de texte

**Competențe operaționale:** Elevii vor ști:

- O1: să insereze tabele;
- O2: să formateze tabele;
- O3: să facă operații cu celule rânduri și coloane;
- O4: să transforme din text în tabel și viceversa ;

### Strategii didactice

#### Principii didactice:

- Principiul participării și învățării active
- Principiul conexiunii inverse
- Principiul asigurării progresului gradat al performanței

**Metode de învățământ:** Metoda mozaic, problematizarea, învățarea prin descoperire, conversația, exercitiul de consolidare, munca independentă

Timp: 1 ora

**Materiale:** fișe de lucru, calculatoare

#### Forme de organizare a activității

- ✓ Frontală
- ✓ Individuală / pe grupe

**Metode de evaluare:** Verificare prin lucru individual

## Desfășurarea activității

### Structura lecției pe secvențe de instruire

Etape ale lecției	Activitatea desfășurată de profesor	Activitatea elevilor	Metode didactice	Timpul alocat
<b>Momentul organizatoric</b>	Se stabilește prezența și se verifică dacă sunt asigurate condițiile didactico-materiale utile desfășurării lecției.	Raportează absenții	Frontală	2 min.
<b>Captarea atenției clasei</b>	Profesorul explică scopul activităților care urmează să fie realizate, anunță obiectivele operaționale și comunică modul de desfășurare a activității.	Ascultă profesorul	Frontală	1 min.
<b>Recapitularea cunoștințelor care urmează să fie folosite în cadrul lecției</b>	Profesorul adresează clasei întrebări: <ol style="list-style-type: none"> <li>1. Ce caracteristici pot fi stabilite prin inițializarea (formatarea) paginii? Cum stabilim aceste caracteristici?</li> <li>2. Care sunt operațiile de editare? Enumerați și explicați care este efectul fiecărei opțiuni?</li> <li>3. Cum folosesc și cum setez tabulatorii?</li> <li>4. Cum aliniez textul?</li> </ol>	Răspund la întrebări	Conversația	7 min
<b>Predarea cunoștințelor noi. (Metoda mozaic)</b>	<p>Profesorul anunță titlul lecției.</p> <p>Def. Tabelul este un obiect Word în care informația este prezentată într-o formă organizată pe linii și coloane.</p> <p>Elementul constructiv al tabelului este celula.</p> <p>Inserarea unui tabel se face prin <b>Table→Insert Table→Se introduce numărul de linii și coloane→OK</b></p> <p>Profesorul stabilește tema de studiu și o împarte în 5 <i>sub-teme</i> (Anexa 1).</p> <ol style="list-style-type: none"> <li>1. Inserarea/stergerea unui rand/unei coloane</li> <li>2. Imbinarea/scindarea celulelor</li> <li>3. Sortarea datelor din tabele</li> <li>4. Formatarea tabelelor</li> <li>5. Formatarea textului din cadrul unui tabel</li> </ol> <p><b><i>Elevii clasei sunt impartiti in echipe de învățare</i></b> de câte 5 elevi (în funcție de numărul lor în clasă).</p> <ul style="list-style-type: none"> <li>• Fiecare elev din echipă, primește un număr de la 1 la 5 și are ca sarcină să studieze în mod independent, sub-tema corespunzătoare numărului său.</li> <li>• El trebuie să devină expert în problema dată. Astfel, elevii cu numărul 1 din toate echipele de învățare formate, vor aprofunda sub-tema cu numărul 1. Cei cu numărul 2 vor studia sub-tema numărul 2, și așa mai departe.</li> </ul> <p><b><i>Constituirea grupurilor de experți:</i></b></p> <p>După ce au parcurs faza de lucru independent, experții cu același număr se reunesc, constituind <i>grupe de experți</i> pentru a dezbate problema împreună. Astfel, elevii cu numărul 1, părăsesc echipele de învățare inițiale și se adună la o masă pentru a aprofunda sub-</p>	<p>Efectuează sarcinile de lucru sub îndrumarea profesorului.</p> <p>Elevii sunt atenți la aprecieri și la recomandările făcute de profesor.</p> <p>Fiecare elev își primește sub-tema și aprofundează subiectul corespunzător sub-temei.</p> <p>Dupa ce a aprofundat tema, fiecare elev se alatura grupului de experți corespunzător sub-temei sale. Aici au loc discutii și clarificari pe anumite aspecte</p> <p>Fiecare elev se întoarce la grupa din care a făcut parte inițial și prezintă colegilor săi conținutul sub-temei sale. De asemenea, fiecare elev își însușește conținuturile sub-temelor prezentate de colegii lui de grupa.</p>	Exercițiul	30 min

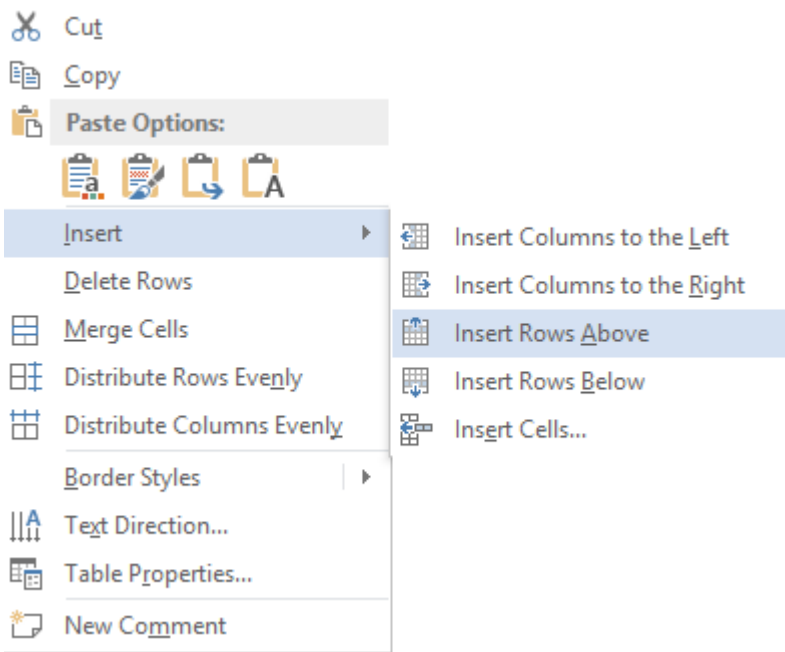
Etape ale lecției	Activitatea desfășurată de profesor	Activitatea elevilor	Metode didactice	Timpul alocat
	<p>tema cu numărul 1. La fel procedează și ceilalți elevi cu numerele 2, 3, 4 sau 5.</p> <p><b>Faza discuțiilor în grupul de experți:</b></p> <p>Elevii prezintă un <i>raport individual</i> asupra a ceea ce au studiat independent. Au loc discuții pe baza datelor și a materialelor avute la dispoziție, se adaugă elemente noi și se stabilește modalitatea în care noile cunoștințe vor fi transmise și celorlalți membrii din echipa inițială.</p> <p>Fiecare elev este membru într-un grup de experți și face parte dintr-o echipă de învățare.</p> <p><b>Reîntoarcerea în echipa inițială de învățare.</b></p> <p><i>Faza raportului de echipă:</i></p> <p>Experții transmit cunoștințele asimilate, reținând la rândul lor cunoștințele pe care le transmit colegii lor, experți în alte sub-teme. Modalitatea de transmitere este scurtă, concisă, atractivă, lucrându-se pe date concrete.</p>			
<b>Asigurarea feedback-ului</b>	<p><b>Evaluarea</b></p> <p>Faza demonstrației</p> <p>În această fază fiecare elev va primi câte o fișă de lucru ce va trebui rezolvată individual.</p>	Fiecare elev rezolvă individual fișa propusă.	exercitiul	10 min



## Teme de studiu

### Sub-tema1. Inserarea/stergerea unui rand/unei coloane

Inserarea unui rand/coloana intr-un tabel se face selectand Inserare rand/coloana



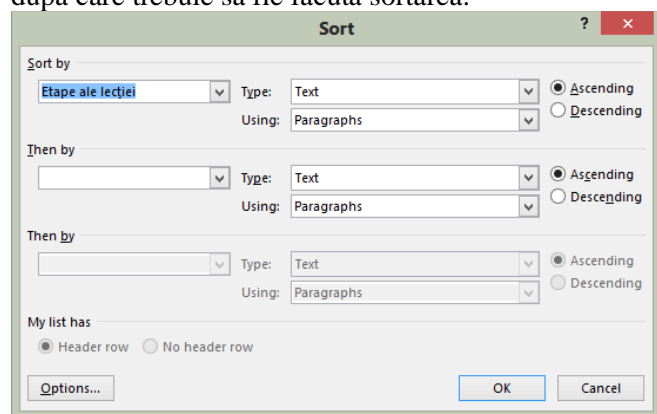
### Sub-tema2. Imbinarea/scindarea celulelor

Pentru imbinarea mai multor celule se selecteaza celulele dorite si se alege Imbinare celule.

Pentru scindarea unei celule in mai multe randuri/coloane se selecteaza celula dorita si se alege Scindare celule. In fereastra noua aparuta se precizeaza numarul de linii/coloane in care se doreste a se face scindarea celulei.

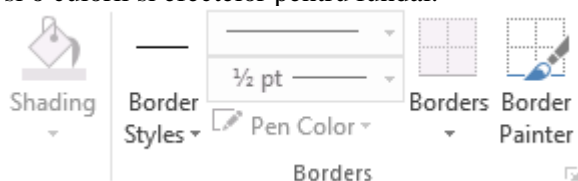
### Sub-tema3. Sortarea datelor din tabele

Pentru sortarea datelor dintr-un tabel se selecteaza optiunea Sortare date din meniul Data dupa care se completeaza criteriile dupa care trebuie sa fie facuta sortarea.



### Sub-tema4. Formatarea tabelelor

Formatarea tabelului se refera la stabilirea tipului de linie pentru contur si la stabilirea tipului de fundal pentru celule. Aceasta formatare se face prin optiunea formatare tabel dupa care se trece la stabilirea culorii, stilului si dimensiunii liniei precum si o culorii si efectelor pentru fundal.






### Sub-tema5. Formatarea textului din cadrul unui tabel


Formatarea textului din cadrul unui tabel consta in stabilirea culorii, fontului, dimensiunii, stilului respectiv orientarii textului.




Times New Ro ▾




12 ▾




A<sup>↑</sup> A<sup>↓</sup> | Aa ▾ | 




B I U ▾ abc x<sub>2</sub> x<sup>2</sup> |  ▾  ▾

Font 










Text



Cell

Direction Margins

Alignment

## Fisa de lucru

Realizati tabelul de mai jos iar apoi ordonati descrescator dupa note elevii din tabel.

Nr crt	Numele si prenumele	Matematica		Punctaj total
		Punctaj proba 1	Punctaj proba 2	
1.	Micu Florin	78	90	168
2.	Crisan Tudor	80	100	180
3.	Duma Daniela	70	40	110
4.	Vasiu Monica	90	50	140
5.	Zdrenghea Ioan	50	20	70
6.	Duvlea Valentin	30	70	100

## Fisa de lucru

Realizati tabelul de mai jos iar apoi ordonati descrescator dupa note elevii din tabel.

Nr crt	Numele si prenumele	Matematica		Punctaj total
		Punctaj proba 1	Punctaj proba 2	
1.	Micu Florin	78	90	168
2.	Crisan Tudor	80	100	180
3.	Duma Daniela	70	40	110
4.	Vasiu Monica	90	50	140
5.	Zdrenghea Ioan	50	20	70
6.	Duvlea Valentin	30	70	100

## Fisa de lucru

Realizati tabelul de mai jos iar apoi ordonati descrescator dupa note elevii din tabel.

Nr crt	Numele si prenumele	Matematica		Punctaj total
		Punctaj proba 1	Punctaj proba 2	
1.	Micu Florin	78	90	168
2.	Crisan Tudor	80	100	180
3.	Duma Daniela	70	40	110
4.	Vasiu Monica	90	50	140
5.	Zdrenghea Ioan	50	20	70
6.	Duvlea Valentin	30	70	100

# PROIECT DIDACTIC

**DATA:** 19.05.2024

**TIMP ALOCAT:** 50 min.

**CLASA:** a X-a

**LOCUL DESFĂȘURĂRII:** sala de clasă

**PROFESOR:** Costache Rebeca

**DISCIPLINA:** Informatică

**UNITATEA DE ÎNVĂȚARE:** Subprograme

**TEMA:** Transmiterea parametrilor prin valoarea în limbajul C++

**TIPUL LECȚIEI:** Lecție de predare-învățare

**SCOPUL LECȚIEI:** însușirea noțiunilor de bază despre transmiterea parametrilor prin valoare în cadrul subprogramelor C++

## COMPETENȚE GENERALE

- identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora;
- elaborarea algoritmilor de rezolvare a problemelor
- implementarea algoritmilor într-un limbaj de programare

## COMPETENȚE SPECIFICE

- Descompunerea problemelor în subprobleme și identificarea modului de transmitere a informațiilor între acestea
- Construirea unor subprograme pentru rezolvarea subproblemelor unei probleme
- Identificarea mecanismului corect de transmitere a parametrilor între subprogramele definite

## OBIECTIVE OPERAȚIONALE:

Elevii trebuie să fie capabili:

- să reproducă și să explice noțiunile de bază despre subprograme
- să evalueze corect evoluția parametrilor transmiși prin valoare
- să rezolve aplicații ce presupun utilizarea noțiunilor dobândite

## OBIECTIVE EDUCAȚIONALE

### OBIECTIVE COGNITIVE:

- să definească corect noțiunile teoretice însușite;
- să aplice corect noțiunile însușite în aplicații concrete.

### OBIECTIVE AFECTIVE:

- să argumenteze anumite situații create în etapele de rezolvare a unei aplicații;
- să manifeste interes față de problemele puse și dorința de învățare prin descoperire proprie a adevărului științific;
- să studieze individual și în echipă, în colaborare și în competiție, cunoscând scopul învățării temei date;
- să aprecieze corect soluțiile oferite de ceilalți colegi.

### OBIECTIVE PSIHOMOTORII:

- să dezvolte gândirea algoritmică, logică, flexibilă, creatoare;
- să-și dezvolte atenția concentrată și spiritul de observație;
- să utilizeze corect noțiunile teoretice însușite;
- să conceapă programe pentru aplicațiile propuse.

## NIVELUL ÎNȚĂL AL CLASEI:

Colectiv eterogen

## STRATEGII DIDACTICE

### ◆ PRINCIPII DIDACTICE:

- principiul participării și învățării active;
- principiul asigurării progresului gradat al performanței;
- principiul conexiunii inverse.

### ◆ METODE DE ÎNVĂȚARE:

- metode de comunicare orală: conversația, explicația;
- metode activ participative: învățarea prin descoperire, problematizarea, exercițiul.

◆ **PROCEDEE DE INSTRUIRE:**

- explicația în etapa de comunicare;
- învățarea prin descoperire;
- problematizarea prin crearea situațiilor problemă;
- conversația de consolidare în etapa de fixare a cunoștințelor.

◆ **FORME DE ORGANIZARE:**

- frontală și individuală

◆ **FORME DE DIRIJARE A ÎNVĂȚĂRII:**

- dirijată de profesor sau prin materiale didactice;
- independentă.

◆ **RESURSE MATERIALE:**

- materiale bibliografice ( manualul, culegeri, prezentare PowerPoint);
- fișe de lucru;
- set de aplicații;

◆ **METODE DE EVALUARE:**

- observarea și aprecierea verbală;
- chestionare orală;
- set de aplicații.

◆ **DESFĂȘURAREA ACTIVITĂȚII:**

**A. Moment organizatoric :**

▪ **Pregătirea lecției**

- întocmirea proiectului didactic;
- pregătirea setului de aplicații.

▪ **Organizarea și pregătirea clasei (2 min.):**

- verificarea prezenței;
- verificarea temei prin sondaj;
- verificarea existenței resurselor materiale.

▪ **Captarea atenției clasei (2 min.):**

- anunțarea subiectului pentru tema respectivă;
- anunțarea obiectivelor urmărite;
- anunțarea modului de desfășurare a activității

**B. Comunicarea noilor cunoștințe (30 min.):**

**Transmiterea prin valoare**

Se utilizează atunci când suntem interesați ca subprogramul să lucreze cu acea valoare, dar să nu poată modifica parametrul efectiv corespunzător din blocul apelator.

Se pot transmite prin valoare:

1. **Valorile reținute de variabile.** În acest caz parametrii efectivi trebuie să fie numele variabilelor.

Exemplu:

```
#include<iostream.h>
void test(int n)
{
    n++;
    cout<<n<<endl; // tipărește n=8
}
void main()
{
    int n=7;
    test(n);
    cout<<n<<endl; // tipărește n=7
}
```

Parametrul **n** este transmis prin valoare. În funcția **main()** acest parametru este inițializat cu valoarea

7. Când apelăm funcția **test()**, se rezervă spațiu pe stivă, spațiu care are numele parametrului formal (în acest caz, tot **n**) și care este inițializat cu valoarea memorată de variabila **n** a programului principal.

Altfel spus, pe stivă se **copie** valoarea parametrului efectiv de apel. În funcție, variabila **n** (care este

locală acestei funcții) este incrementată și devine 8, valoare care va fi tipărită. La ieșirea din funcție, variabila **n** din stivă se pierde, adică nu mai are spațiu alocat, prin urmare valoarea 8 este pierdută. În **main()** se tipărește valoarea variabilei **n** (locală acesteia) care are valoarea 7.

Pentru exemplul anterior, conținutul stivei, în momentul apelului și după execuția funcției **test()**, este următorul:

<b>n=7</b> , parametru valoare de apel, local funcției <b>test</b>	
<b>n=7</b> , var. locală funcției <b>main()</b>	<b>n=7</b> , var. locală funcției <b>main()</b>
adresa de revenire din funcția <b>test</b>	adresa de revenire din funcția <b>test</b>

#### după execuția funcției **test**

Se observă că, în momentul apelului funcției **test()**, pe stivă sunt alocate două variabile cu același nume **n**. Prima variabilă este variabila locală funcției **main()** care se salvează pe stivă în momentul apelului pentru a putea reface contextul funcției **main()** după încheierea apelului. A doua variabilă este parametrul formal tip valoare **n**, vizibil numai în funcția **test()** și inițializat în momentul apelului cu valoarea 7. Indiferent ce valori primește acest **n** în corpul funcției **test()**, după încheierea execuției acestei funcții, spațiul său este de alocat din stivă, adică variabila respectivă este distrusă. Din acest motiv, după execuția funcției **test()**, conținutul stivei este cel din dreapta. Se reface contextul din care s-a lansat apelul funcției **test()**, adică se recuperează din stivă valoarea variabilei locale **n=7** și adresa de revenire, adică adresa instrucțiunii **cout**.

1. **Expresii.** În acest caz, parametrii efectivi sunt expresii, care pot conține și funcții și care mai întâi se evaluează. Exemplu:

```
#include<iostream.h>
#include<math.h>
void test(int n)
{
    cout<<n<<endl;
}
void main()
{
    test(5);           // se va tipări 5
    test(7+(int)sqrt(45)); // se va tipări 13
}
```

În funcție se creează o variabilă numită **n**, reținută pe stivă, care la primul apel va primi valoarea 5 și la al doilea apel valoarea 13. La ieșirea din funcție conținutul acestei variabile se pierde.

**Transmiterea parametrilor prin valoare** se utilizează când nu dorim ca subprogramul apelat să poată modifica parametrii efectivi de apel. Acesta este modul implicit de transmitere a parametrilor în limbajul C. Dacă nu ar exista decât transmiterea prin valoare, ar fi imposibil să modificăm valoarea anumitor valori care sunt declarate în blocul apelator.

#### C. Fixarea noilor cunoștințe și realizarea feed-back-ului (15 min.):

1. Precizați care dintre următoarele linii de program reprezintă corect, din punct de vedere sintactic, lista de parametri și valoarea returnată de o funcție cu numele **test**, dacă cerem ca parametrii formali să fie două variabile de tip întreg, **x** și **y**, și o variabilă de tip real **z**, și să returneze un rezultat întreg.
  - a) `int test(int x,y,float z)`
  - b) `int test(int x,y;float z)`
  - c) `int test(int x,int y,float z)`
  - d) `test(int x;int y;float z) int`
  - e) `test(int x,int y,float z) int`
2. Care dintre afirmațiile de mai jos sunt adevărate?
  - a) la apelul unei funcții, se produce înlocuirea parametrilor formali cu parametrii actuali
  - b) tipul parametrilor dați la apelul unei funcții trebuie să coincidă sau să fie compatibil cu tipul celor definiți în antetul funcției
  - c) la apelul unei funcții, se salvează pe stivă adresa de revenire, precum și variabilele locale și parametrii modulului apelat
  - d) orice funcție trebuie să aibă întotdeauna cel puțin un parametru
  - e) nici una dintre afirmațiile de mai sus
3. Considerăm o funcție `demo`, de tip `void` pentru care se definesc ca parametri trei variabile întregi. Cum realizăm apelul funcției, astfel încât la apel să dăm ca parametri variabilele întregi `a`, `b` și `c`?
  - a) `demo(int a,int b,int c);`
  - b) `demo(int a;int b;int c);`
  - c) `demo(a,b,c);`
  - d) `demo(a;b;c);`
  - e) `demo(int a,b,c);`
4. Scrieți o funcție `D` care primește ca parametru un număr întreg `a` și returnează valoarea lui `a+2`.
  - a) `int D(int a);{D=(a+2);}`
  - b) `int D(int a){D=a+2;}`
  - c) `int D(int a){return(a+2);}`
  - d) `int D(int a){return a+2;}`
  - e) nici una dintre variantele anterioare
5. Care dintre afirmațiile de mai jos sunt adevărate?
  - a) parametrii definiți în antetul unei funcții se numesc actuali, iar cei care apar la apelul funcției se numesc formali
  - b) valoarea returnată de către o funcție poate fi transmisă ca parametru altei funcții
  - c) variabilele de tip tablou nu se pot transmite ca parametri funcțiilor
  - d) variabilele globale sunt cunoscute pe tot parcursul programului în care au fost declarate în toate modulele care urmează declarației
  - e) corpul unei funcții trebuie cuprins între “{” și “}”, numai dacă este alcătuit din cel puțin două instrucțiuni distincte
6. Avantajele utilizării funcțiilor într-un program sunt:
  - a) se poate obține o economisire a spațiului de memorie rezervat variabilelor folosite în cadrul programului
  - b) viteză mai mare în execuția programului
  - c) posibilitatea de a executa de mai multe ori instrucțiunile cuprinse într-o funcție
  - d) un program care conține funcții poate fi urmărit și corectat mai ușor
  - e) nici unul dintre avantajele de mai sus
7. Scrieți o funcție cu numele `divizori` care primește prin parametrul `n` un număr natural nenul cu maxim 9 cifre și returnează numărul divizorilor proprii ai numărului `n`.
8. Deduceți șirul de valori care se afișează în urma execuției programului de mai jos:

```
#include<iostream.h>
int i,j,k;
int test(int x,int y)
```

```

{
    return (x-y);
}
void calcul(int p,int q)
{
    int u,v;
    u=p-i; v=q+j;
    i=test(u,q);
    j=test(v,p);
}
void main()
{
    i=2; j=3;
    calcul(i,j);
    cout<<i<<" "<<j<<endl;
    calcul(j,i);
    cout<<i<<" "<<j<<endl;
}

```

a) 2 3 2 3

b) 3 2 3 2

c) 2 3 -3 4

d) -3 4 4 -3

e) -3 4 10 -3



9. Fie programul:

```
#include<iostream.h>
int j,x;
int test(int,int);
void main()
{
    x=2;
    cout<<test(x,x+3);//(1)
    cout<<x;        //(2)
}
int test(int a,int b)
{
    int x=a+b;
    for(j=1;j<=x;j++)
        if(j>a && j<b) return j;
    return 0;
}
```

Precizați care dintre afirmațiile de mai jos sunt adevărate:

- a) instrucțiunea (1) afișează valoarea 3
- b) instrucțiunea (2) afișează valoarea 7
- c) funcția **test** returnează 0, indiferent care ar fi parametrii dați la apel
- d) corpul funcției **test** este eronat, deoarece conține două instrucțiuni return
- e) programul este eronat, deoarece variabila **x** a fost declarată de două ori

10. Determinați valorile pe care le afișează programul de mai jos:

```
#include<iostream.h>
int x,y;
int T(int m,int n)
{
    m=n+x; n+=1;
    return(n+y+m);
}
```

```
void main()
{
    y=10; x=12;
    cout<<T(x,y);
    cout<<x<<" "<<y;
}
```

- a) 43,22,10      b) 43,12,10      c) 47,10,12      d) 44,22,11      e) 44,12,11

**D. Temă pentru acasă (1 min.)**

Aplicații: - exercițiile 1, 2, 3, 5, 6 din manual, pagina 165.

Nume:.....

## LECȚIE DE EVALUARE SUMATIVĂ

### Test de verificare

#### Subprograme

Clasa a XI-a, matematica informatica

#### Subiecte

1. Explicați noțiunile de parametri locali, parametri globali, parametri formali transmiși prin valoare și parametri formali transmiși prin referință. (2p)
2. Se consideră un șir de n numere reale. Să se afișeze șirul format prin inversarea elementelor de pe pozițiile consecutive în șirul inițial. (3p)

Ex: șir inițial 3 2 5 8 9 10

șir final 2 3 8 5 10 9

3. Ce realizează următorul program:(1p)

```
#include<iostream.h>
int n,x,k;
int v[10];
void aparitii(int &k)
{
    int k=0;
    for (int i=0;i<n;i++)
        if(v[i]= =x) k++;
    n++;
}
void main()
```

```
{
    cin>>n;
    for (int i=0;i<n;i++)
        cin>>v[i];
    cout<<"x=";
    cin>>x;
    aparitii(k);
    cout<<k<<n;
}
```

4. Corelați prin intermediul săgeților componentele programului C++ prezentat cu definițiile din partea dreaptă: (2p)

Definiția subprogramului

Corpul subprogramului

Antetul subprogramului

Directive de procesare

5. Completați spațiile libere (2p)

În exemplul de mai jos, am definit funcția....., de tip....., cu parametrii.....de tip..... După cum observăm funcția returnează o valoare....., adică de același tip cu tipul..... Funcția se apelează

---

prin.....are doua argumente .....de același tip cu  
parametrii funcției.

La apelul funcției valoarea returnată de funcție se depune în variabila.....

În programul următor se citesc .....  
calculând .....

```
#include <iostream.h>
#include <conio.h>
int suma(int a, int b);
void main()
{
    int x,y, z;
    cout<<"x=";<<cin>>x;
    cout<<"y=";<<cin>>y;
    z=suma(x,y);
    cout<<"suma="<<z;
    getch();
}

int suma(int a,int b);
{
    int c;
    c=a+b;
    return c;
}
```

Timp de lucru 50 min  
1p oficiu

Unitatea de învățare: **Subprograme**

## implementate în C++ Aplicații de laborator



« Cu cât este mai **diffică** problema,  
cu atât mai **mulțumit** vei fi când o vei rezolva »

**Realizați în C++ următoarele aplicații:**

### Aplicatia nr.1.

Timp de lucru alocat: 10 min

Sa se scrie o functie care sa rezolve o ecuatie de gradul II, unde coeficientii ecuatiei sunt dati ca parametri. Functia principala va citi coeficientii a doua ecuatii si va afisa solutiile pentru fiecare ecuatie in parte.

### Aplicatia nr.2.

Timp de lucru alocat: 15 min

Se citește o literă și un număr întreg n. Să se calculeze:

- pentru litera a  $S1=1^2+3^2+5^2+\dots+(2n-1)^2$
- pentru litera b  $S2=1+1*2+1*2*3+\dots+1*2*3\dots n$

**Indicație:** Definim funcția s1:

```
long s1(int n) // funcția s1 are un parametru transmis prin valoare
{
    int i; long s=0;
    for(i=1; i<=n; i++)
        s+=pow(2*i-1,2);
    return s; // funcția s1 returnează valoarea calculată
}
```

### Aplicatia nr.3.

Timp de lucru alocat: 15 min

Se citește un număr natural n. Să se verifice și să se afișeze dacă este un număr prim, astfel să se afișeze descompunerea în factori primi.

**Indicație:**

- Definim funcția **prim**, care primește ca parametru transmis prin valoare numărul n și returnează 1 dacă numărul este prim și 0 dacă numărul nu este prim.
- Definim funcția **factori**, funcție fără tip, care primește ca parametru numărul n și afișează factorii primi și ordinul lor de multiplicitate.

### Aplicatia nr.4.

Timp de lucru alocat: 10 min

Sa se scrie o functie care sa verifice daca un numar intreg este sau nu palindrom (este egal cu rasturnatul sau).

Timp de lucru alocat: 10 min

Sa se scrie o functie care sa verifice daca un doua numere intregi sunt prime gemene (sunt prime si diferenta in modul este egala cu 2).



## Proiect didactic

**Unitatea școlară:**

**Data:** 19.05.2024

**Profesor:** Costache Rebeca

**Clasa:** a XI-a (matematică - informatică)

**Disciplina:** Informatică

**Locul de desfășurare:** sala de clasă

**Unitatea de învățare:** *Subprograme*

**Tema:** *Aplicatii care folosesc subprograme*

**Tipul lecției:** ”consolidarea cunoștințelor”

**Competențe generale:**

1. Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora
2. Elaborarea algoritmilor de rezolvare a problemelor
3. Implementarea algoritmilor într-un limbaj de programare

**Competențe specifice:**

1. Recunoașterea situațiilor în care este necesară utilizarea unor subprograme
2. Analiza problemei în scopul identificării subproblemelor acesteia
3. Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator
4. Implementarea unor algoritmi de prelucrare a tablourilor bidimensionale, a șirurilor de caractere și a structurilor neomogene

**Resurse:**

*Timp:* 50 minute

*Materiale:* cretă, tablă, manual, calculator, fișa de lucru

*Procedurale:*

- ✓ Metode de comunicare orală

- ✎ Expunerea
- ✎ Conversația
- ✓ Metode de acțiune
  - ✎ Exercițiul
  - ✎ Analiza problemei
  - ✎ Învățarea prin descoperire
- ✓ Procedee de instruire
  - ✎ Explicația în etapa de comunicare
  - ✎ Învățarea prin descoperire, prin rezolvarea de aplicații
  - ✎ Conversația în etapa de fixare a cunoștințelor
- ✓ Forme de organizare a activității
  - ✎ Frontală
- ✓ Metode de evaluare
  - ✎ Probe orale si scrise(pe calculator)

#### **Material bibliografic de specialitate:**

- ☞ Manualul „Informatică – varianta C++” pentru clasa a XI-a, Tudor S., Editura L&S Infomat, 2000;
- ☞ “Didactica predării informaticii”, Masalagiu C., Asiminoaei I., Editura Polirom, București, 2004;
- ☞ “Evaluarea progresului școlar – de la teorie la practică”, Stoica I., Editura Humanitas Educational, 2003;
- ☞ “Metode de învățământ”, Cerghit I., Editura Polirom, 2006.
- ☞ C++ Manual Complet Herbert Schildt Ed. Teora, 2000
- ☞ C++ fără mistere , Jeff Kent, Ed. Rosetti Educational, 2007

**Desfășurarea activității**  
**Structura lecției pe secvențe de instruire**

Nr. crt	Etape	Timp	Activitate desfasurata de profesor	Activitate desfasurata de elevi	Metode de învățământ	Evaluare
1.	Moment organizatoric	2'	Se stabilește prezența și se verifică dacă sunt asigurate condițiile didactico-materiale utile desfășurării lecției	Pregatesc materialele necesare .	Conversatia Analiza problemei	Probe orale si scrise(pe calculator)
2.	Verificarea temei	3'	Control frontal Intrebari privind realizarea temei.	Pregatesc caietele pentru control.		
3.	Recapitularea unor cunostinte despre un suprogram.	5'	a) De ce folosim un subprogram? b)Care sunt partile unui subprogram? c)Ce inseamna a declara si ce inseamna a defini un subprogram in C++? d)Ce intelegem prin variabile locale si prin variabile globale?	Raspund la intrebarile puse de profesor		
4.	Fixarea si consolidarea cunostintelor	35'	a)Distribuie fisele de lucru b)Propune rezolvarea problemelor din fisa. c)Analizeaza impreuna cu elevii proiectarea rezolvarii problemelor Solicita elevilor sa introduca in calculator programele realizate la tabla. d)Acorda ajutorul necesar compilarii si executiei programelor.	* Se concentreaza la intrebarile di fisa. * Se anunta pentru a rezolva la tabla problemele propuse. * Introduc in calculator programele.		



5.	Evaluarea cunostintelor(aprecieri)	2'	Apreciaza felul in care elevii au raspuns la intrebari si au rezolvat problemele din fisa .	Compara modul de lucru cu rezultatele celorlalti.		
6.	Notarea elevilor	1'	Se noteaza elevii care au avut o participare concludenta.	Prezinta carnetele de note.		
7.	Comunicarea temei	1'	Se precizeaza problemele din tema. Acestea vor fi problemele din FISA de lucru care un au fost rezolvate in clasă	Noteaza in caiete tema.		
8.	Observatii si concluzii	1'	Face o prezentare a modului de desfasurare a orei.	Pot pune intrebari cu eventuale nelamuriri.		