

Editată cu minte, trup și suflet

REVISTA 4 ONLINE

Nr. 16, februarie 2023

INSPECTORATUL ȘCOLAR JUDEȚEAN GALAȚI

ASOCIAȚIA PROFESORILOR DE INFORMATICĂ
“TUDOR SORIN ”
DIN GALAȚI

REVISTA DE INFORMATICĂ

(iM)PULS ++

NR. 16, februarie 2023

4-ONLINE



Utilizarea platformei Open Roberta Lab în cadrul orelor de programare și de robotică de la gimnaziu

Profesor Penu Ionela

Open Roberta Lab este un mediu de programare online ce permite programarea roboților virtuali 2D și reali 3D. Platforma OPEN ROBERTA LAB, prin limbajul NEPO oferă posibilitatea de a programa atât roboți virtuali, folosind simulatorul integrat, cât și pe cei reali. Simulatorul integrat platformei permite programarea virtuală a unui model simplu de robot 2D pe un calculator cu o configurație simplă, conectat la Internet. Acest fapt face ca predarea roboticii educaționale să fie posibilă în toate școlile din România prin utilizarea resurselor actuale ale școlilor, fără costuri suplimentare, folosindu-se doar calculatoarele din dotarea laboratoarelor de informatică și conexiunea la Internet.

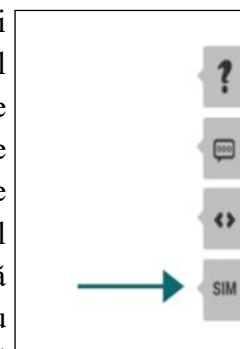
Este rezultatul unui proiect educațional inițiat de Ministerul Educației și Cercetării din Germania în colaborare cu institutul de cercetare Fraunhofer IAIS și dezvoltat împreună cu Google. org. Poate fi utilizată gratuit de orice elev, profesor, sau utilizator fără restricții. Proiectată de Beate Jost, Reihard Budde, Thorsten Leimbach, Achim Kapusta, platforma

Open Roberta a fost testată pentru prima dată în 2013, a apărut oficial în 2014, dar prima versiune stabilă, 2.0, a apărut în 11 octombrie 2016. (Oprea, 2018)

Open Roberta Lab se poate accesa folosind adresa <https://lab.open-roberta.org/>.

Mediul de simulare Open Roberta permite testarea programelor. Acest lucru este foarte util dacă nu există un robot real la îndemână. Programele OPEN Roberta pot fi rulate pe roboți reali sau în mediul de simulare. În simulare este disponibil un model simplu de robot 2D. Mediul de simulare pentru robotul 2D va fi afișat făcând clic pe butonul SIM din dreapta mediului de programare.

Același buton SIM poate fi utilizat pentru a închide mediul de simulare. În mediul de simulare există mai puține blocuri de programare executabile decât în mediul EV3. Acest lucru se datorează modelului robot simplu disponibil în prezent în mediul de simulare.



În panoul din stânga este afișată o parte a programului.

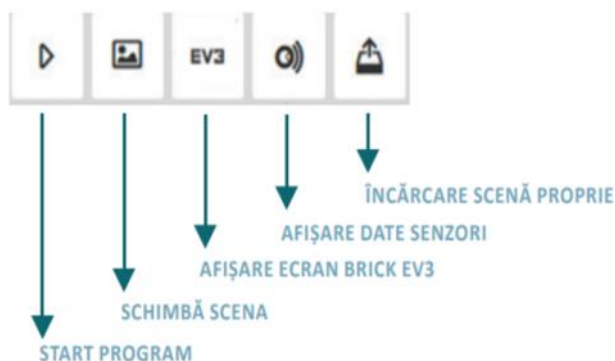
Panoul drept arată robotul 2D în mediul selectat în prezent.

The screenshot shows the Open Roberta Lab interface. On the left, there is a vertical menu of programming blocks: Action (orange), Sensors (green), Control (orange), Logic (blue), Math (blue), Text (green), Colours (yellow), and Variables (purple). The main workspace shows a 2D robot on a track. On the right, there is a 2D simulation of the robot on a track. At the bottom, there are several control buttons for simulation.

În colțul din stânga jos există câteva butoane pentru a controla simularea.



În colțul din stânga jos al ferestrei de simulare există câteva butoane pentru a controla simularea robotului într-un anumit mediu definit de o imagine numită scenă. Butoanele disponibile în mediul de simulare sunt diferite în funcție de tipul de robot selectat dintre cei prevăzuți în mediul Open Roberta Lab. Imaginea alăturată prezintă butoanele disponibile pentru modelul EV3.



Figură 2 - Butoane de control în Open Roberta Sim

Mediul de simulare poate fi schimbat. Există 6 medii standard prezentate în imaginile de mai jos, la care se pot adăuga medii personalizate adăugate de utilizator.

Mediul de simulare poate fi schimbat.

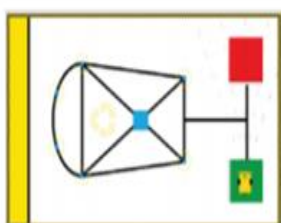
Există 6 medii standard prezentate în imaginile de mai jos, la care se pot adăuga medii personalizate adăugate de utilizator.

NEPO (New Easy Programming Online) este un limbaj de programare din cadrul platformei Open Roberta Lab care utilizează biblioteca Blockly la care au fost adăugate funcționalități suplimentare și îmbunătățiri, adaptate pentru Open Roberta Lab. Paradigma de programare a NEPO este inspirată de Scratch de la Massachusetts Institute of Technology.

Un bloc NEPO reprezintă și încorporează o anumită funcționalitate a robotului. Blocuri sunt organizate pe categorii, în funcție de acțiunile pe care le pot programa, de exemplu senzori. Blocurile sunt interconectate și vor fi executate de robot conform ordinii lor, începând cu blocul start.

Open Roberta Lab oferă două opțiuni pentru blocurile de acțiuni NEPO: Începător și Expert. Modul Începător este cel implicit. Această categorie include 8 blocuri care sunt cele mai importante pentru programarea unui robot. Modul Expert include în plus trei categorii de blocuri: "Liste", "Funcții" și "Mesaje".

WORLD ROBOT OLYMPIAD SCENE



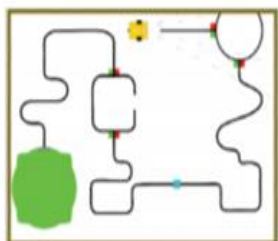
SIMPLE SCENE



DRAWING SCENE



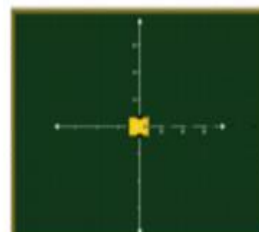
RESCUE SCENE



ROBERTA SCENE



MATH SCENE



Figură 3- Medii de simulare- scene



Profesorul virtual sau AI?

Profesor Penu Ionela

Care este primul lucru la care te gândești când auzi cuvântul “inteligenta artificială”?

Inteligenta Artificială are capacitatea de a învăța singură. Sunt mai multe moduri în care învață: învățare non-supervizată (în care învață prin experiențe proprii), învățarea prin recompense etc. Așa s-a ajuns ca un soft dotat cu Inteligență Artificială să învețe absolut toate schemele de șah performate până acum și să depășească omul în doar 4 ore. Pentru că Inteligența Artificială învață altfel. Pe baza unei scheme proprii, mult mai eficiente. Iar această învățare se transmite de la “părinți”, la “copii”. Google a creat Inteligența Artificială care dă naștere Inteligenței Artificiale (acum o perfecționează). Iar “copiii” își depășesc ”părinții”, pentru că știu deja CUM. Știu deja cum să învețe. Schema de învățare se transmite genetic. Ei nu mai învață să învețe, știu deja asta să facă și pornesc direct la lucru într-un mod foarte eficient și rapid.

Care este scopul aceste învățări? Să ne fie nouă de folos. Mai exact, să ne îndeplinească dorințele înainte de a le exprima. Acum, când tastăm pe Google cuvântul Londra, platforma încearcă să vină în întâmpinarea dorințelor noastre înainte de a le exprima prin faptul că afișează în partea dreaptă, pe lângă obișnuitele linkuri la site-uri, starea vremii din Londra, numărul de locuitori ai capitalei Angliei etc. Se “gândește” că aceste informații ne-ar putea interesa. Ca să vină în întâmpinarea exactă a dorințelor noastre înainte de le exprima, Inteligența Artificială trebuie să ne cunoască foarte bine. Și atunci ne urmărește tot timpul. Pentru a ne învăța.

De curând, “cercetătorii de la universitățile Cambridge și Stanford au dezvoltat un algoritm de calculator care poate determina personalitatea cuiva cu mai mare precizie decât prietenii și familia acestuia, fără a utiliza nimic altceva decât activitatea pe Facebook a subiectului”. Este un

algoritm dotat cu Inteligența Artificială similar milioanei de algoritmi care ne urmăresc fiecare pas pe Facebook, 24h/24h, prin intermediul cookie-urilor implantate în calculatoarele și telefoanele noastre. Acest nou algoritm te urmărește pentru a te învăța și a te cunoaște în final...mai bine decât soția ta. Pe baza a 300 de like-uri. Citat: “Facebook te cunoaște mai bine decât soția ta [...] Acesta are nevoie de acces la doar 10 Like-uri pentru a fi mai bun la determinarea personalității ca un coleg de muncă, 70 de Like-uri pentru „a bate” un coleg de cameră, 150 pentru „a bate” un părinte sau frate și 300 de Like-uri pentru a întrece un soț / soție”.

De ce am ales să scriu despre acest subiect în primul articol dedicat Inteligenței Artificiale? Pentru că acesta este drumul spre care vom merge. Repede. Dependența față de telefon determină producătorii să-l transforme într-un dispozitiv din ce în ce mai aproape de noi, fizic, pentru a ne permite să-l urmărim tot timpul, mai exact, în timp ce desfășurăm activitățile noastre zilnice. Să nu mai ieșim niciodată de pe Facebook (Instagram etc). Pentru asta, Facebook (Instagram etc) trebuie să nu ne incomodeze în mișcări, cum face acum telefonul, să ne ia toată atenția, dar să ne permită, în același timp, să observăm realitatea din jur și să ne fie foarte de folos, pentru că altfel nu l-am dori, totuși, atât de aproape și permanent cu noi. Iar Inteligența Artificială este cheia: învață mereu despre noi și ne oferă totul într-un mod care ne face viața mai ușoară. Înlocuiește traducerea gândurilor în cuvinte, pentru că le aude și le traduce singură. Învață cum gândim și, în acest fel, rata de succes a traducerii este incredibil de mare: Alter Ego traduce cu o acuratețe de 92% mesajele gândite. Este sub procentul de 95% atins de Google’s voice transcription service prin intermediul unui microfon tradițional, dar sistemul se va îmbunătăți în scurt timp. Vi se pare un scenariu depărtat în timp?



Școala și materiile studiate sunt un subiect intens dezbătut în această perioadă. S-a vehiculat ideea că școala românească este perimată, profesorii depășiți și de aceea experții în educație caută soluții de schimbare a procesului de învățământ și a curriculei.

Cu ocazia Zilelor Colegiului Alexandru Ioan Cuza am organizat o serie de întâlniri on-line la care au participat foști absolvenți ai Colegiului și actualii elevi, din anii terminali dar și din alte clase de liceu. Unul dintre obiectivele acestor întâlniri a fost consilierea elevilor în ceea ce privește alegerea unui



parcurs educațional universitar și nu numai.

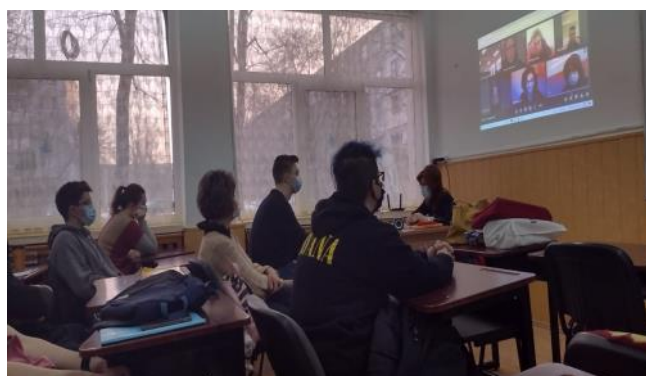
Absolvenții care au urmat facultăți din domeniul IT au făcut fiecare câte un scurt review al vieții de student. Pe lângă experiențele haidose împărtășite de ei, au fost descrise și disciplinele de studiu. Toți absolvenții, fără excepție, au subliniat faptul că materia informatică studiată în liceu i-a ajutat foarte mult pe tot parcursul facultății, că nu scapă nimeni de matematică, cel puțin în anul I, că facultatea te învață câte puțin din fiecare, urmând ca tu ca individ să aprofundezi ceea ce îți place mai mult.

Absolvenții de IT, chiar dacă în liceu nu înțelegeau utilitatea studierii algoritmicii, a limbajului C++ sau a limbajului SQL, studiat în clasa a XII-a, au descoperit în timpul facultății că toate acestea sunt puncte de plecare în studierea altor limbaje și pot fi chiar avantaje în găsirea unui internship. Absolvenții claselor de matematică informatică au povestit că anul I a fost unul foarte ușor deoarece ei aveau deja cunoștințele necesare promovării examenelor de la disciplinele legate de informatică.

Elevii au mai aflat de la absolvenții altor facultăți, cu profile diferite de IT, că programarea poate fi utilă și în alte domenii, dar că cel mai important lucru pe care trebuie să îl știi când termini o facultate este cum să lucrezi într-o echipă, cum să contribui cu cunoștințele tale la realizarea unui proiect în echipă, că degeaba ești foarte bun la programare, dacă nu poți comunica cu alții.

Noi ca profesori de informatică, am fost încă o dată plăcut surprinse de faptul că ceea ce facem cu elevii noștri la disciplina informatică este un punct de plecare solid spre o carieră în domeniul IT. Noțiunile de programare studiate la disciplina Informatică sunt uneori confundate cu noțiunile de operare studiate la disciplina TIC - tehnologia informațiilor și a comunicațiilor. Poate de aceea nu se consideră importantă disciplina informatică în lista disciplinelor de BAC și se vehiculează ideea de a o scoate dintre opțiunile de BAC ale elevilor de la clasele de matematică-informatică.

Împreună cu invitații noștri, am încurajat elevii pasionați de informatică să continue studiul acestei discipline, dar nu i-am neglijat nici pe ceilalți elevi cărora le-am recomandat să descopere ceea ce le place, să învețe să caute informații potrivite nevoilor lor, să experimenteze mai multe domenii de studiu, dacă nu sunt hotărâți încotro să meargă, să cunoască cât mai mulți oameni, să nu le fie frică să



ia decizii, să fie adaptabili.

Magdalena Grigore, Daniela Ilie
prof. informatică Colegiul Național
"A.I.Cuza", Galați



Integrarea noilor tehnologii în procesul educativ

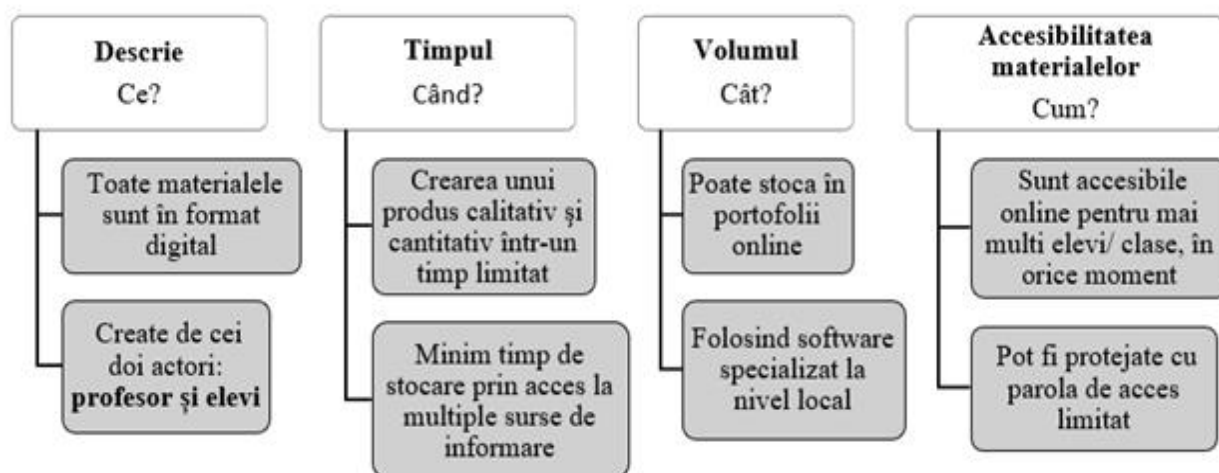
Prof. Humelnicu-Ungureanu Mihaela

Școala Gimnazială nr. 17 Galați

Folosirea tehnologiei în domeniul educației le oferă elevilor posibilități nelimitate de învățare și evoluție. Prin integrarea tehnologiei elevii sunt implicați în rezolvarea problemelor propuse. Noile tehnologii desemnează:

- ◆ noi forme de comunicare și transmitere a informației
- ◆ instrumente de prezentare multimedia care permit un grad ridicat de interactivitate.

O creștere a calității instruirii și motivației pentru învățare se observă atunci când metodele tradiționale se completează cu metode moderne, care implică noile tehnologii.



Modelul bazat pe tehnologie este marcat de câteva dimensiuni fundamentale:

- * standardizare a nivelului de informații transmise elevului
- * caracter practic, prin reducerea atât a costurilor, cât și a timpilor de deplasare
- * performanță mare de învățare
- * eficiență.



Aceste activități generează motivația intrinsecă și prin deprinderile pe care le dobândesc elevii își construiesc o gândire științifică corectă.



APLICAȚII	DESCRIERE	Exemple
FLIPSNACK	Este una dintre cele mai cunoscute platforme de creare de materiale digitale interactive: reviste, broșuri, ebook digital. Puteți accesa aplicația: www.flipsnack.com .	https://www.flipsnack.com/ungureanumi/abilitati-de-tastare-rapida.html https://www.flipsnack.com/ungureanumi/posta-electronica.html https://www.flipsnack.com/ungureanumi/6_efecte-de-tranzitie-tp54pc14ot.html
JIGSAWPLANET	Este o aplicație care ne ajută să creăm puzzle-uri. Avem posibilitatea: de a încărca o fotografie, de a seta dificultatea puzzle-ului (numărul de elemente) timpul de rezolvare. Puteți accesa aplicația: https://www.jigsawplanet.com	https://www.jigsawplanet.com/?rc=play&pid=35d8bbc1da22 https://www.jigsawplanet.com/?rc=play&pid=3e634dbaa0e8 https://www.jigsawplanet.com/?rc=play&pid=394af3bd6ebe
WORDWALL	Este utilizată pentru a crea activități interactive. Sunt disponibile mai multe șabloane într-o versiune interactivă. Puteți accesa aplicația: https://wordwall.net/	https://wordwall.net/ro/resource/24251636/dsipozitive-periferice https://wordwall.net/ro/resource/24251114/dsipozitive-periferice https://wordwall.net/ro/resource/24250951/dsipozitive-periferice https://wordwall.net/ro/resource/24250549/identificare-dsipozitive-periferice https://wordwall.net/ro/resource/24251337/identificati-dsipozitivele-de-intrare
Liveworksheets	Permite transformarea fișș elor de lucru tradiționale tipărite: .pdf sau .jpg în foi de lucru interactive. Puteți accesa aplicația: https://www.liveworksheets.com	https://www.liveworksheets.com/4-ix976676oz https://www.liveworksheets.com/4-1f698131sc https://www.liveworksheets.com/4-ik698410rq https://www.liveworksheets.com/4-uc689032ep https://www.liveworksheets.com/4-so699155uk https://www.liveworksheets.com/4-qk698048sj https://www.liveworksheets.com/4-jf698897kb https://www.liveworksheets.com/4-bx698525zk
Google Forms	Ne permite crearea testelor cu ajutorul formularelor. Puteți accesa aplicația: https://www.google.ro/intl/ro/forms/about/	https://docs.google.com/forms/d/e/1FAIpQLSdnMUOxvc1gpDYBXoFNp4AM_ch4LOsA-a7ZkaIfd3LcCDzhmQ/viewform https://docs.google.com/forms/d/e/1FAIpQLSc2N5V_goqgRRGo9aT_gLbzQgkJ-2PQLzLK9h-kWbIwLAJvA/viewform https://docs.google.com/forms/d/e/1FAIpQLSc7VVSyG35Sp_kn0a060KQhII50F6QgZ-a-Yp0limv2tC9vEg/viewform https://docs.google.com/forms/d/e/1FAIpQLSceJiZwjyPliTM_Xxge2UEeZXCASEm98BSQH9LePRd1jfhFvQ/viewform https://docs.google.com/forms/d/e/1FAIpQLSe1eGpMRIGzN8IUaBRxfCW-GyVLFHQHKnPD0kj0BIX7F6oOKQ/viewform



APLICAȚII	DESCRIERE	Exemple
WORDART	Este o aplicație care realizează ușor un „nor de cuvinte” – un mod de reprezentare vizuală a unui text sau a unui șir de cuvinte. Puteți accesa aplicația: https://wordart.com/	https://wordart.com/8xuadzwwtads/virusi-informatici https://wordart.com/6huzkp1y0lin/email
Kahoot!	Este o platformă de învățare bazată pe joc și tehnologie educațională. Pentru crearea unui test se alege comanda Quiz . Se trece la scrierea întrebărilor, apăsând, pentru fiecare întrebare nouă opțiunea Add question . Varianta corectă de răspuns, este marcată cu bifă . Puteți accesa aplicația: https://kahoot.com/	https://kahoot.it/challenge/02839459?challenge-id=8e9b8342-5023-43a2-b933-bc66c5b9e7c3_1643471963885 https://kahoot.it/challenge/07938526?challenge-id=8e9b8342-5023-43a2-b933-bc66c5b9e7c3_1643472187250
PADLET	Reprezintă un instrument de lucru colaborativ oferind un spațiu virtual pentru întrebări și răspunsuri. Puteți accesa aplicația: https://padlet.com/	https://padlet.com/ungureanumi/c5rh7kfyf7yeqzol
JAMBOARD	Este o tablă digitală interactivă dezvoltată de Google, cu ajutorul căreia utilizatori aflați la distanță pot să schițeze idei și să le salveze în Cloud. Puteți accesa aplicația: https://jamboard.google.com	https://jamboard.google.com/d/1SQ2yd-4rau2730CrgP_UHBJLIIdwe5AQyDN4BuRcaBo/viewer?f=0 https://jamboard.google.com/d/1AEjqIKxZqHgWjJqPoEIZ7urWSRzSYn1CzTwuhoO7BQ4/viewer?f=0

Bibliografie



Crenguța-Lăcrămioara Oprea – *Strategii didactice interactive*, Editura Didactică și pedagogică, 2007
<https://sites.google.com/view/ungureanu-mihaela/pagina-de-pornire?authuser=0>



WIKI

Profesor Visterneanu Claudia-Dora

Liceul de Turism și Alimentație “Dumitru Moțoc”

Ce este wiki?

Este un site proiectat pentru ca grupuri de persoane să capteze și să împărtășească rapid idei, un site web de colaborare online care poate fi setat pentru a fi editat de oricine sau doar de anumiți utilizatori. Acest site se poate realiza prin crearea de pagini simple și legarea acestora.

O organizație, o școală, un grup de elevi, pot utiliza un wiki pentru o varietate de proiecte disciplinare și interdisciplinare.

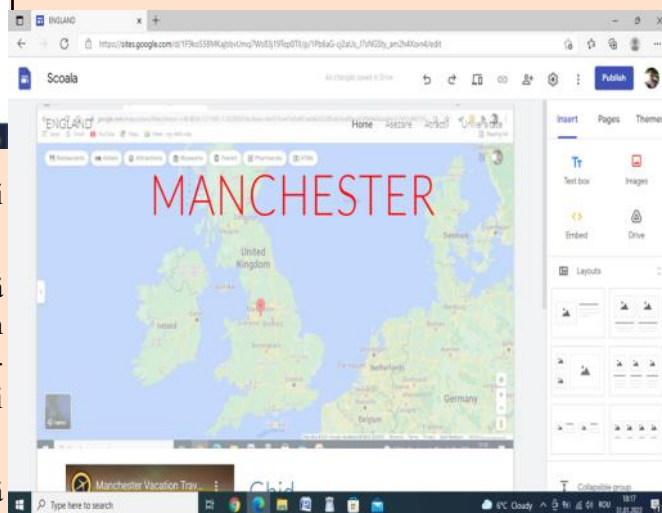
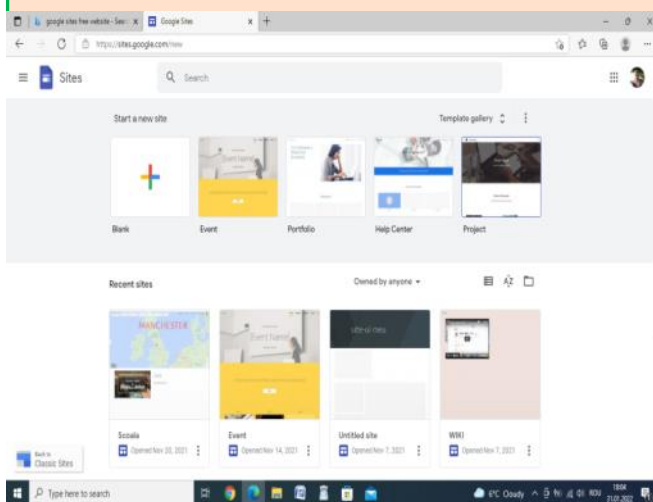
Un wiki se poate utiliza atât la scară largă cât și la scară mică. La scară largă, se pot partaja volume mari de informații. La scară mică, se poate utiliza un site de echipă pentru a colecta și a împărtăși rapid idei pentru un anumit proiect. Wiki

este creată în biblioteca Pagini de site.

Cum se crează un Wiki: persoana trebuie să se înscrie pentru un cont în Google înainte de a putea crea un Wiki cu **Google Sites**. Odată creat contul Google, se **accesază Google Sites** și se face clic pe butonul „Creați site” pentru a începe crearea propriului Wiki.

Google Sites conține diferite șabloane, care se pot alege pentru site-ul care urmează a fi creat. Se selectează „Project Wiki” și astfel începe crearea wiki-ului. Există opțiunea de a partaja în mod public wiki-ul sau de a partaja cu un grup de persoane. Se pot adăuga colaboratori și membri în wiki prin selectarea „Partajați acest site” din meniul drop-down pentru mai multe acțiuni și se introduce adresa de e-mail a colaboratorilor. Se pot invita alte persoane să colaboreze la wiki, prin invitații trimise pe e-mail și acorda permisiuni diferite fiecărei persoane. Se pot face personalizări cu meniul „Gestionați site-ul”. Astfel se pot schimba culoarea, fonturile și tema. Se pot insera diverse elemente cum ar fi: Picasa, Foaie de calcul, Document, Prezentare, etc.

Mai jos avem exemplul unui site Școală, un site de prezentare a orașului Manchester din Marea Britanie. Acest site este format din 4 pagini: Home, Așezare, Atracții și Universitate, cum se vede în



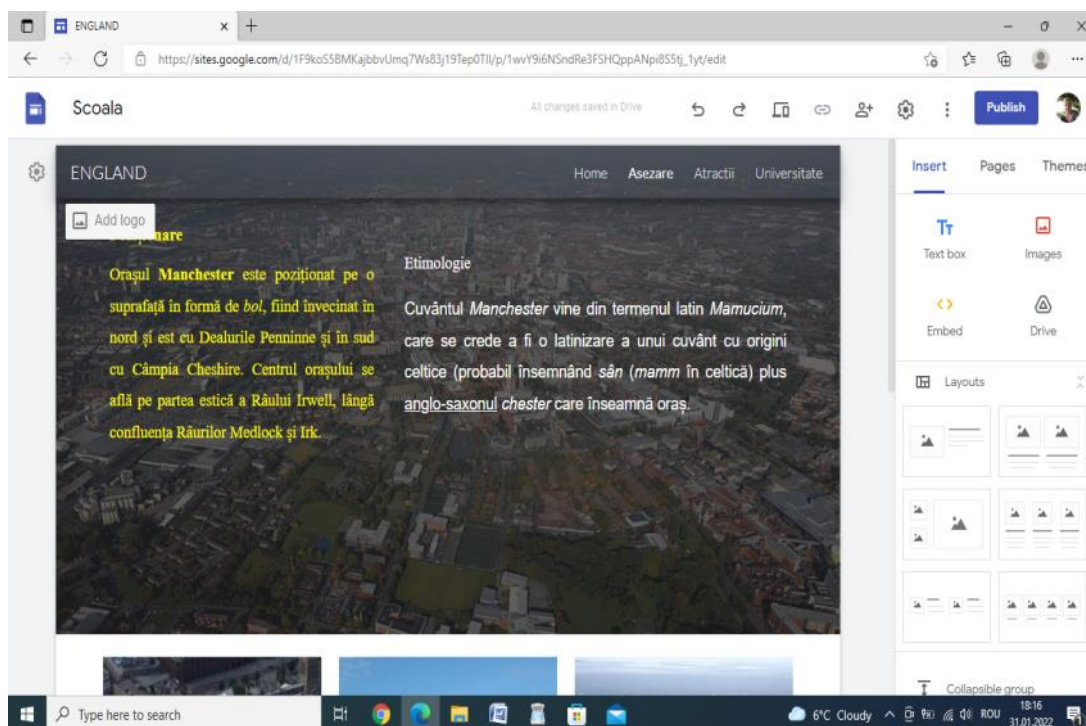
reprezintă o modalitate excelentă de a comunica și de a lucra la un proiect cu alte persoane.

Website-ul Wiki permite vizitatorilor să adauge, să ștergă, să editeze și să modifice un conținut disponibil. Autorul unui wiki poate fi notificat cu privire la toate modificările și poate urmări și monitoriza dezvoltarea conținutului site-ului.

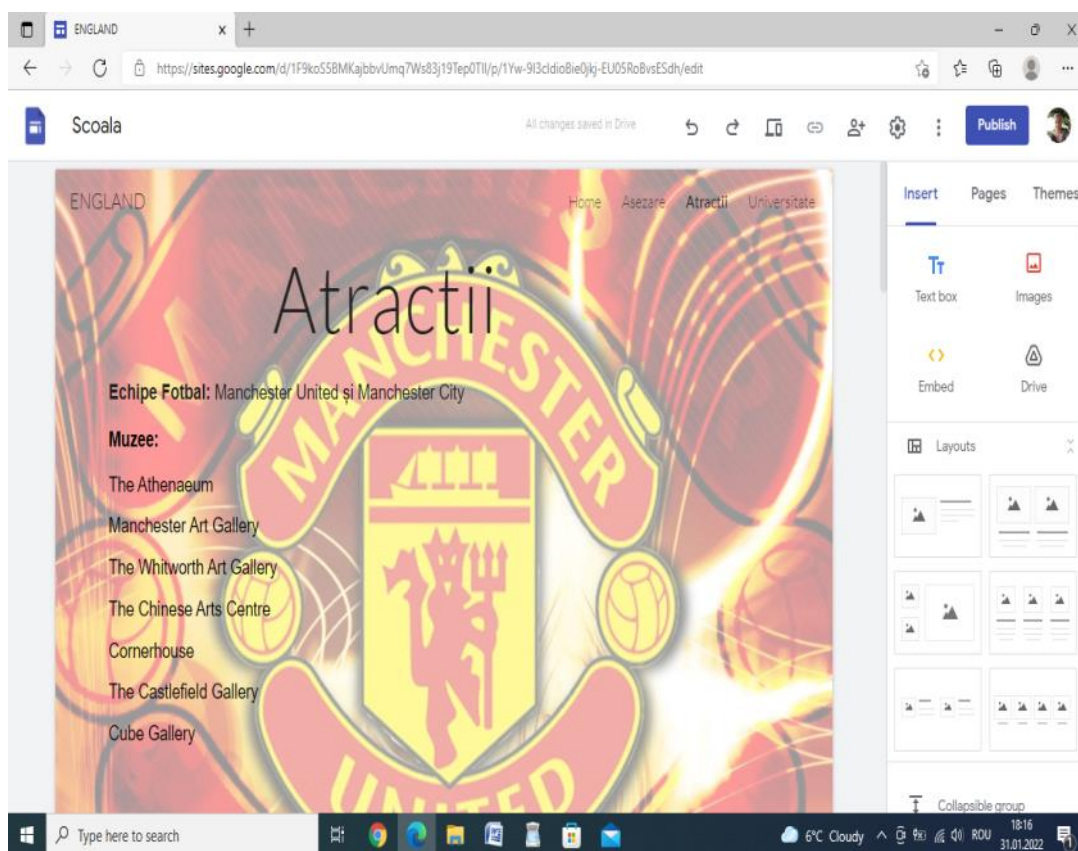
Într-un site de echipă, fiecare pagină nouă



Site-urile Google facilitează realizarea unui Wiki fără a avea nici un fel de cunoștințe tehnice. Are o interfață ușor de utilizat, care permite crearea unui Wiki atrăgător din punct de vedere vizual și ușor de utilizat.

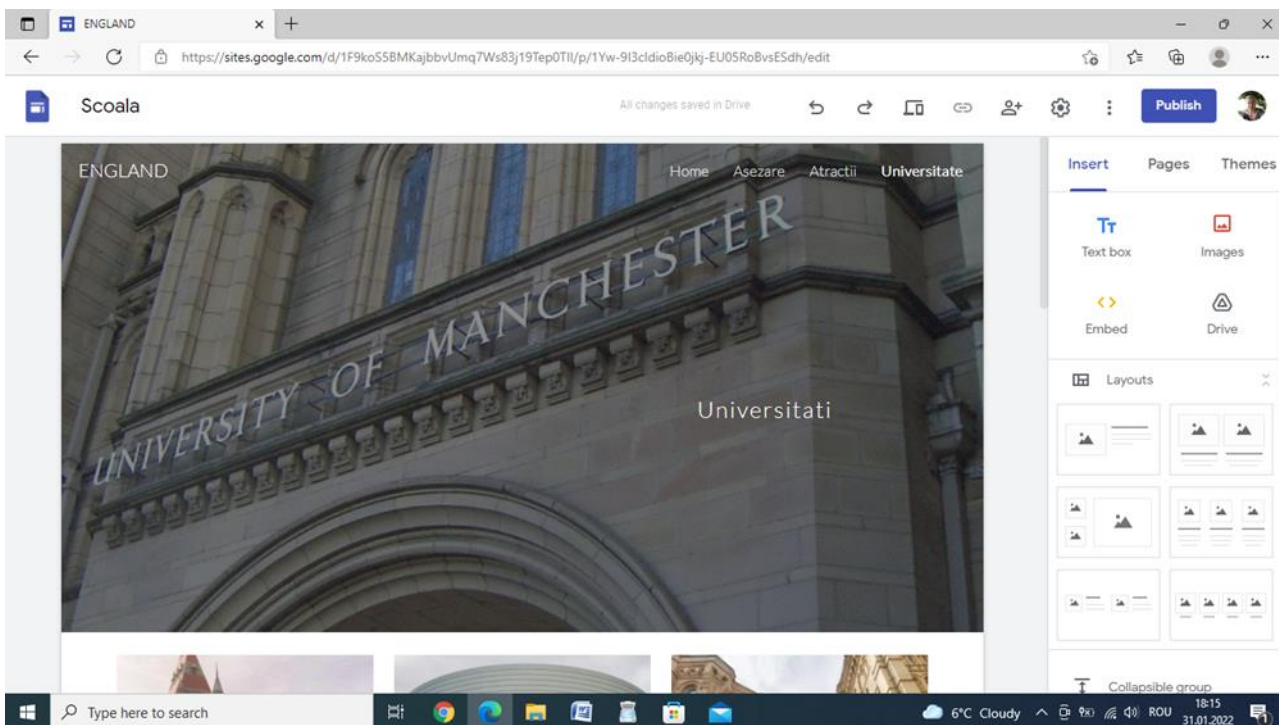


Site-urile Wiki sunt ușor de realizat, sunt atractive pentru elevii de gimnaziu cât și de liceu.



Aceste site-uri se pot folosi atât pentru orele de TIC cât și pentru alte discipline separate sau chiar interdisciplinar, prin crearea unor proiecte pe diverse subiecte.

atât pentru orele de TIC, pentru alte discipline cât și interdisciplinar, prin crearea unor proiecte pe diverse subiecte.



Manchester (*/ˈmʌntʃəstər/*) este unul dintre marile orașe [engleze](#), organizat din punct de vedere administrativ sub forma unui [Burg Metropolitan](#) în cadrul [Comitatului Metropolitan Greater Manchester](#) în regiunea [North West England](#). Orașul are o populație propriu zisă de 441.200 locuitori și se află în centrul unei conurbații cu o populație de 2.240.230 locuitori, fiind cea de a 3-a conurbație engleză după [Londra](#) și [West Midlands](#).

Manchester este unul dintre primele orașe industrializate ale lumii, fiind cunoscut rolul important pe care l-a avut în timpul [Revoluției Industriale](#). A fost centrul dominant internațional al manufacturii materialelor textile, în secolul XIX fiind numit și *Cottonopolis* (cotton = bumbac; polis = cetate, oraș).

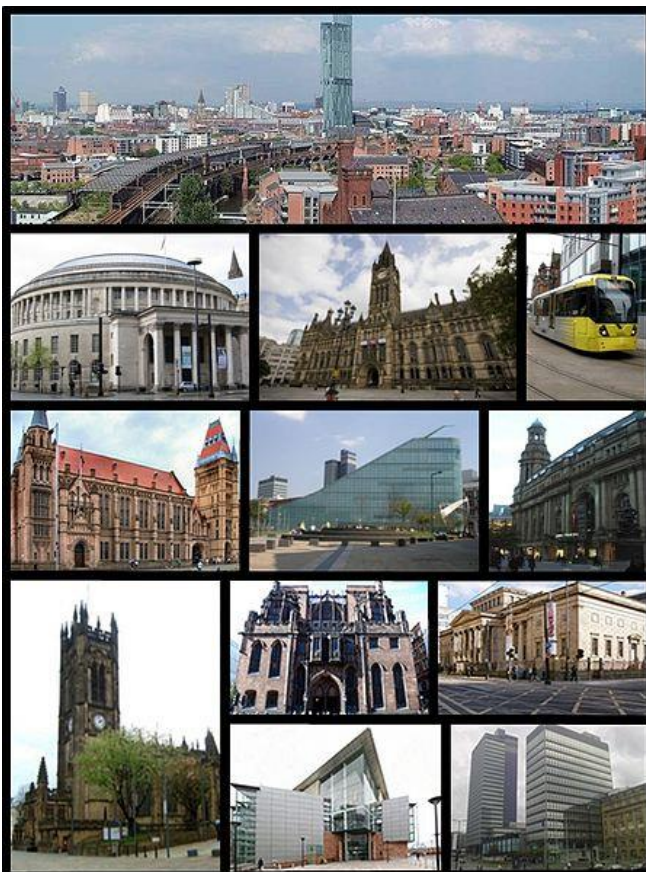
Poziționare[[modificare](#) | [modificare sursă](#)]

Orașul **Manchester** este poziționat pe o suprafață în formă de *bol*, fiind învecinat în nord și est cu Dealurile Pennine și în sud cu Câmpia Cheshire. Centrul orașului se află pe partea estică a Râului Irwell, lângă confluența Râurilor Medlock și Irk.

Climă[[modificare](#) | [modificare sursă](#)]

Manchester are reputația unui oraș *plouos*, în ciuda unei medii anuale a precipitațiilor de 809 mm, care este mult mai mică decât în orașe ca [Plymouth](#), [Cardiff](#) sau chiar [New York](#) care are o medie anuală de 1200 mm. Precipitațiile sunt în general ușoare astfel încât o ploaie cu un volum mic de apă poate dura și o oră. Căderile de zăpadă sunt destul de rare în oraș, fiind mai frecvente spre vest.

<https://ro.wikipedia.org/wiki/Manchester>



Utilizarea resurselor Web 2.0 în procesul de predare-învățare a noțiunilor de algoritmică și programare la gimnaziu

Penu Ionela
Profesor de informatică și TIC
Școala Gimnazială „Miron Costin”, Galați

Motto:

„Să nu-i educăm pe copii pentru lumea de azi. Această lume nu va mai exista când ei vor fi mari. Și nimic nu ne permite să știm cum va fi lumea lor. Atunci să-i învățăm să se adapteze.”
(*Maria Montessori*)

Trăim într-o epocă confruntată cu schimbări profunde, cu un ritm accelerat de dezvoltare a tehnologiilor informaționale fapt care a condus la înregistrarea unei adevărate revoluții în domeniul educației. Sistemul de educație este direct implicat în evoluția societății informaționale prin formarea și dezvoltarea competențelor digitale.

Nucleul sistemului de educație este elevul iar, profesorii, au obligația, să îl pregătească pentru o lume computerizată, o lume a tehnologiei. Elevii trebuie încurajați să perceapă tehnologia ca fiind parte din viața lor iar omul este cel care pune în mișcare tehnologia. Elevii trebuie motivați să își dezvolte competențele digitale pentru a deveni creatori de produse informatice și nu doar utilizatori.

Conform planului-cadru pentru învățământul gimnazial, aprobat prin OMENCS nr. 3590/2016, disciplina Informatică și TIC se predă în clasele a V-a, a VI-a, a VII-a și a VIII-a, ca disciplină de trunchi comun, cu o alocare de o oră/ săptămână. Implementarea acestui plan-cadru a început cu anul școlar 2017-2018. Astăzi, în momentul conceperii acestui material putem spune că suntem la a doua generație de elevi care au studiat informatică pe durata a 4 ani de studiu.

În acest context, acest articol își propune să trateze un subiect de actualitate în procesul de învățământ, și anume dezvoltarea de strategii didactice care utilizează resurse Web 2.0 pentru predarea noțiunilor de algoritmică și programare elevilor de gimnaziu, conducând astfel la formarea și dezvoltarea competenței de crearea a produselor digitale.

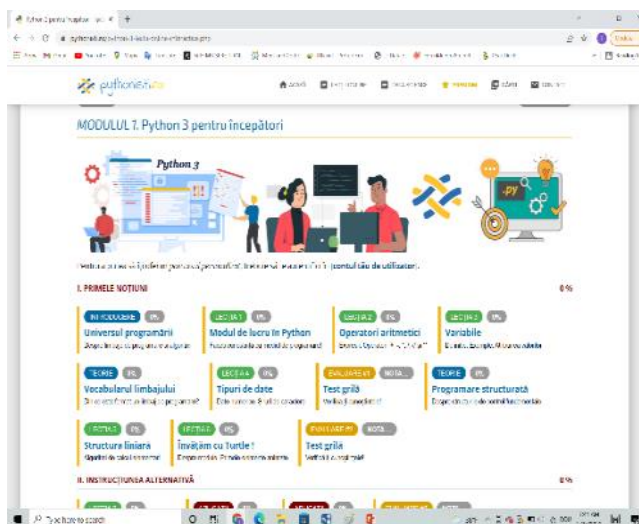
Resursele Web 2.0 sunt considerate drept mecanism de sprijin pentru pregătirea și realizarea materialelor didactice, evaluarea, analizarea evo-

luției elevilor, realizarea prezentărilor formative și informative, realizarea orarului și a calendarului de activități, dezvoltarea de proiecte în colaborare, cea mai mare parte dintre acestea fiind centrate pe elev. Web 2.0 se bazează pe încredere, dorința oamenilor de a se dezvolta prin dobândirea cunoștințelor și pe aplicațiile Open Source.

În continuare vor fi prezentate câteva instrumente Web cu destinații diverse în contextual lecțiilor de algoritmică și programare.

Platforma *pythonisti.ro*

Un instrument Web 2.0 care poate fi folosit în predarea noțiunilor de programare este platforma *pythonisti.ro* care pune la dispoziție lecții online pentru studierea limbajului de programare Python. Este un proiect dezvoltat de infoBits Academy și susținut de Uniunea Profesorilor de Informatică din România.



Pe lângă lecții, pe această platformă există și o comunitate puternică de elevi, părinți și profesori care au ca scop principal educația de calitate a viitorilor programatori și specialiști IT.

Lecțiile online sunt interactive, respectă programa școlară aprobată M.E.N. și pot fi folosit în cadrul oricărui ciclu de învățământ ca o introducere în limbajul de programare Python 3. Informațiile prezentate sunt optimizate ca dimensiune și se pune accentul pe interacțiunea frecventă cu utilizatorul.

Evaluarea exercițiilor și problemelor propuse în acest curs este în timp real și fiecare conține feedback interactiv.



Probleme ATESTAT Informatică 2021-2022
***Subiecte și rezolvări PROGRAMARE* ISJ Galați**

Subiectul nr 1

Se citesc de la tastatură mai multe numere naturale până când se întâlnește valoarea 0. Să se afișeze pe ecran câte dintre aceste numere sunt egale cu suma cuburilor cifrelor lor.

Exemplu: 153 10 1 34 5 0 se afișează 2 (sunt 2 numere: 153 1)

```
#include <iostream>
using namespace std;
int s,x,k;
int suma_cuburilor_cifrelor(int x)
{
    int s=0;
    while(x)
    {
        int u=x%10;
```

```
        s+=u*u*u;
        x/=10;
    }
    return s;
}
int main()
{
    cin>>x;
    while(x)
    {
        if(x==suma_cuburilor_cifrelor(x))
            k++;
        cin>>x;
    }
    cout<<k;
    return 0;
}
```

Subiectul nr 2

Se citește un număr de la tastatură de cel mult 9 cifre. Să se afișeze la ecran numărul obținut prin micșorarea fiecărei cifre nenule.

Exemplu: Dacă se citește numărul 1048 se va afișa 37

```
#include<iostream>
using namespace std;
int x, y, p;
int main()
{
```

```
    cin>>x;
    p=1;
    while(x!=0)
    {
        if(x%10!=0)
            y=y+(x%10-1)*p;
        else y=y+(x%10)*p;
        p=p*10;
        x=x/10;
    }
    cout<<y;
    return 0;
}
```

Subiectul nr 3

Se citește de la tastatură un text de maxim 50 caractere litere mici ale alfabetului englez. Să se afișeze textul obținut prin ștergerea primelor k vocale (k citit de la tastatură). Dacă în text nu există k vocale se vor șterge toate vocalele.

Exemplu: Dacă textul citit este *voi lua nota maxima la bacalaureat* și k=11 se va afișa *v l nt mxm l bcalaureat*

```
#include <iostream>
#include<cstring>
using namespace std;
char s[51], voc[]="aeiou";
int i, k;
int main()
{
    cin.get(s, 51);
```

```
    cin>>k;
    for(i=0;i<strlen(s) && k!=0;i++)
    if(strchr(voc,s[i])!=0)
    {
        strcpy(s+i, s+i+1);
        i--;
        k--;
    }
    cout<<s;
    return 0;
}
```



Subiectul nr 4

Se citesc de la tastatură trei numere naturale a, b și c cu maxim 9 cifre fiecare. Să se verifice dacă numerele a, b și c au exact aceiași factori în descompunerea lor în factori primi (exponenții lor putând să difere). Se va afișa pe ecran "DA" sau "NU".

Exemplu:

dacă a=300, b=1500, c=30 se afișează DA

dacă a=300, b=700, c=140 se afișează NU

```
#include <iostream>
using namespace std;
int fa[100],fb[100],fc
[100],a,b,c,nra,nrb,nrc;
void descompunere(int x,int v
[100],int &nr)
{ int b=2,e=0;
while(x!=1)
{ while(x%b==0)
```

```
{ x/=b; e++; }
if(e)
v[++nr]=b;
b++; e=0;
}
}
int main()
{ cin>>a>>b>>c;
descompunere(a,fa,nra);
descompunere(b,fb,nrb);
descompunere(c,fc,nrc);
int ok=1;
for(int i=1; i<=max(max
(nra,nrb),nrc); i++)
if(!(fa[i]==fb[i]&&fa[i]
==fc[i])) ok=0;
if(ok==1) cout<<"DA";
else cout<<"NU";
return 0; }
```

Subiectul nr 5

Prin asocierea fiecărei litere mici din alfabetul englez cu un număr egal cu poziția literei în alfabet, se poate codifica orice secvență de litere mici cu o secvență de numere. Se citește din fișierul text cod.txt un număr n ($0 < n < 30000$) și de pe rândul următor, o secvență de n litere mici. Scrieți un program care va codifica secvența de caractere citită și va scrie pe ecran numerele asociate, separate prin spații.

Exemplu: pentru fișierul cod.txt: 4; info

se vor afișa pe ecran, în ordine, numerele: 9 14 6 15

```
#include <iostream>
#include <fstream>
```

```
#include <cstring>
using namespace std;
ifstream f("cod.txt");
char s[201];
int n;
int main()
{
f>>n;
f.get();
f.get(s,201);
for(int i=0; i<strlen(s); i++)
cout<<int(s[i]-'a'+1)<<"
";
return 0;
}
```

Subiectul nr 6

Se citesc de la tastatură un număr n ($n \leq 100$) și apoi n elemente ale unui tablou unidimensional v. Să se afișeze pe ecran cea mai lungă secvență de elemente citite consecutiv cu proprietatea că fiecare componentă a succesiunii are același număr de cifre distincte în scrierea sa.

Exemplu: n=9 și elementele 55, 8, 37, 949, 26, 2213, 157, 38, 2 se afișează 37, 949, 26.

```
#include <iostream>
using namespace std;
int n,v[101],fr[10],fr2
[101],poz,poz2,poz1f,poz2f,maxi;
int main()
{ cin>>n;
for(int i=1; i<=n; i++)
{ cin>>v[i];
int aux=v[i];
for(int j=0; j<=9; j++) fr[j]
=0;
while(aux)
{ fr[aux%10]++;
```

```
aux/=10;
}
for(int j=0; j<=9; j++)
if(fr[j]!=0) fr2[i]+=1;
}
for(int i=1; i<n; i++)
{
if(fr2[i]==fr2[i+1])
{ poz=i;
i++;
while(fr2[i]==fr2[i+1])
poz2=i;
if(poz2-poz+1>maxi)
{ maxi=poz2-poz+1;
poz1f=poz;
poz2f=poz2;
}
}
}
for(int i=poz1f;
i<=poz2f; i++)
cout<<v[i]<<" ";
return 0;
```

Subiectul nr 7

Se citește un număr $n < 100$. Să se afișeze primele n numere cu următoarea proprietate: pătratul răsturnatului este egal cu răsturnatul pătratului.
Exemplu: dacă $n=12$ se va afișa 1 2 3 11 12 13 21 22 31 101 102 103.

```
#include <iostream>
using namespace std;
int n,x,k,inv, invp, pinv,
aux;
int main()
{
    cin>>n;
    x=1;
    while(k<n)
    {
        aux=x;
        inv=0;
        while(aux!=0)
        {
```

```
            inv=inv*10+aux%10;
            aux=aux/10;
        }
        pinv=inv*inv;
        invp=0;
        while(pinv!=0)
        {
            invp=invp*10+pinv%
10;
            pinv=pinv/10;
        }
        if(invp==x*x)
        {
            cout<<x<<" ";
            k++;
        }
        x++;
    }
    return 0;
}
```

Subiectul nr 8

Se citește din fișierul *date.in* o matrice A cu m linii și n coloane ($m \leq 20, n \leq 20$).

Să se construiască și să se afișeze în fișierul *date.out* matricea B obținută prin rotirea cu 90° în sens trigonometric a matricei A.

Exemplu: dacă $m=3$ și $n=5$ și matricea A este

$$\begin{pmatrix} 7 & 4 & 2 & 3 & 7 \\ 9 & 6 & 5 & 1 & 6 \\ 4 & 7 & 2 & 5 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 7 & 6 & 9 \\ 3 & 1 & 5 \\ 2 & 5 & 2 \\ 4 & 6 & 7 \\ 7 & 9 & 4 \end{pmatrix}$$

matricea B obținută va fi

```
#include <fstream>
using namespace std;
ifstream f("date.in");
```

```
ofstream g("date.out");
int A[21][21], B[21][21], m, n, i, j;
int main()
{
    f>>m>>n;
    for(i=1; i<=m; i++)
        for(j=1; j<=n; j++)
            f>>A[i][j];
    for(i=1; i<=n; i++)
        for(j=1; j<=m; j++)
            B[i][j]=A[j][n+1-i];
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=m; j++)
            g<<B[i][j]<<" ";
        g<<endl;
    }
    f.close();
    g.close();
    return 0;
}
```



Subiectul nr 9

Se citește un număr $n < 100$. Să se afișeze primele n numere prime palindrom.

Un număr este palindrom dacă citit de la stânga la dreapta sau de la dreapta la stânga este același.

Exemplu: dacă $n=12$ se va afișa 2, 3, 5, 7, 11, 101, 131, 151, 181, 191, 313, 353.

```
#include <iostream>
using namespace std;
int n, x, k, ok, d, aux, inv;
int main()
{
    cin >> n;
    x = 2;
    while (k < n)
    {
        ok = 1;
```

```
for (d = 2; d <= x / 2; d++)
    if (x % d == 0)
        ok = 0;
aux = x;
inv = 0;
while (aux != 0)
{
    inv = inv * 10 + aux % 10;
    aux = aux / 10;
}
if (ok == 1 && inv == x)
{
    k++;
    cout << x << " ";
}
x++;
}
return 0;
}
```

Subiectul nr 10

Se citește o matrice pătratică A de dimensiune n ($n \leq 20$).

Să se afișeze suma elementelor prime de pe diagonala secundară.

Exemplu: dacă matricea A este $n=5$ și

$$\begin{pmatrix} 2741 & 1199 & 27 & 13 & 5 \\ 22 & 6 & 7008 & 1 & 1999 \\ 45 & 46 & 202 & 201 & 33 \\ 1024 & 29 & 111 & 469 & 199 \\ 96 & 213 & 21 & 4096 & 16 \end{pmatrix}$$

se va

afișa 34

```
#include <iostream>
#include <cmath>
using namespace std;
int A[21][21], i, j, s, n, ok, d;
int main()
{
    cin >> n;
```

```
for (i = 1; i <= n; i++)
    for (j = 1; j <= n; j++)
    {
        cin >> A[i][j];
        if (i + j == n + 1)
        {
            ok = 1;
            for (d = 2; d <= sqrt(A[i][j]); d++)
                if (A[i][j] % d == 0)
                    ok = 0;
            if (A[i][j] == 0 || A[i][j] == 1)
                ok = 0;
            if (ok == 1)
                s = s + A[i][j];
        }
    }
    cout << s;
    return 0;
}
```

Subiectul nr 11

Se citește un număr $n < 25$. Să se afișeze primele n numere cu următoarea proprietate: suma divizorilor fiecărui număr este un număr prim.

Exemplu: dacă $n=12$ se va afișa 1 2 4 9 16 25 64 289 729 1681 2401 3481.

```
#include <iostream>
using namespace std;
int n, x, k, suma, ok, d;
int main()
{
    cin >> n;
    x = 1;
    while (k < n)
    {
        suma = 0;
        for (d = 1; d <= x; d++)
```

```
if (x % d == 0)
    suma = suma + d;
ok = 1;
for (d = 2; d <= suma / 2; d++)
    if (suma % d == 0)
        ok = 0;
if (ok == 1)
{
    cout << x << " ";
    k++;
}
x++;
}
return 0;
}
```



<p>Subiectul nr 12</p> <p>Se citesc un număr n ($n \leq 100$) și apoi n elemente ale unui tablou unidimensional v. Să se elimine din tablou elementele care au suma cifrelor număr prim. Să se afișeze pe ecran tabloul rezultat. Exemplu: dacă $n=12$ și elementele 34 1109 56 875 285 531 1190 11 98 287 444 41 se va afișa 875 285 531 444.</p> <pre>#include <iostream> using namespace std; int n, v[101], i, j, s, aux, ok, d; int main() { cin>>n; for(i=1; i<=n; i++) cin>>v[i]; for(i=1; i<=n; i++) { s=0; aux=v[i];</pre>	<pre>while(aux!=0) { s=s+aux%10; aux=aux/10; } ok=1; for(d=2; d<=s/2; d++) if(s%d==0) ok=0; if(s==1) ok=0; if(ok==1) { for(j=i; j<n; j++) v[j]=v[j+1]; n--; i--; } for(i=1; i<=n; i++)</pre>
<p>Subiectul nr 13</p> <p>Se citesc un număr n ($n \leq 100$) și apoi n elemente ale unui tablou unidimensional v. Să se ordoneze descrescător după suma cifrelor elementele tabloului v și apoi să se afișeze tabloul ordonat. Exemplu: dacă $n=12$ și elementele 34 1109 56 879 285 531 1190 17 98 287 444 41 se va afișa 879 98 287 285 1190 1109 56 531 441 17 34 41.</p> <pre>#include <iostream> using namespace std; int n, v[101], auxi, auxj, aux, i, j, si, sj; int main() { cin>>n; for(i=1; i<=n; i++) cin>>v[i];</pre>	<pre>for(i=1; i<n; i++) for(j=i+1; j<=n; j++) { auxi=v[i]; auxj=v[j]; si=0; sj=0; while(auxi!=0) { si=si+auxi%10; auxi=auxi/10; } while(auxj!=0) { sj=sj+auxj%10; auxj=auxj/10; } if(si<sj)</pre>
<p>Subiectul nr 14</p> <p>Se citește de la tastatură un număr natural nenul n, cu maximum 8 cifre. Să se construiască în memorie și să se afișeze apoi pe ecran o matrice pătratică, având numărul de linii și de coloane egal cu numărul de cifre ale numărului dat, completată cu cifrele numărului citit. Astfel, elementele de pe ultima coloana a matricei vor fi toate egale cu cifra unităților numărului dat, elementele de pe penultima coloană a matricei vor fi toate egale cu cifra zecilor numărului dat, și așa mai departe, ca în exemplu. Exemplu: Dacă se citește 28463, matricea construită va fi următoarea:</p> <pre> 2 8 4 6 3 2 8 4 6 3 2 8 4 6 3 2 8 4 6 3 2 8 4 6 3 2 8 4 6 3 2 8 4 6 3 2 8 4 6 3</pre>	<pre> 2 8 4 6 3 2 8 4 6 3 #include <iostream> using namespace std; int a[9][9]; int main() { int n, c=0; cin>>n; int aux=n; while(aux) { aux/=10; c++; } for(int i=1; i<=c; i++) {</pre>



Subiectul nr 15

Se citesc din fișierul fib.in un număr natural n (n<=100) și apoi un șir format din n numere ordonate crescător.

Scrieți programul care determină și afișează în fișierul fib.out numărul de termeni ai șirului dat care au proprietatea că aparțin șirului lui Fibonacci.

Exemplu: pentru n=6 și numerele 2, 5, 10, 13, 13, 25

se afișează valoarea 4, deoarece termenii 2, 5, 13 și 13 fac parte din șirul lui Fibonacci.

```
#include <fstream>
using namespace std;
ifstream f("fib.in");
ofstream g("fib.out");
int n,a,b,c,nr;
int main()
{
    f>>n;
    a=0; b=1; c=1;
```

```
while(f>>n)
{
    if(n==a||n==c)
        nr++;
    else if((n>c))
    {
        while(n>c)
        {
            a=b;
            b=c;
            c=a+b;
        }
        if(n==c)
            nr++;
    }
}
g<<nr;
return 0;
}
```

Subiectul nr 16

Se citesc două numere a și b (a, b cu maxim 9 cifre). Să se afișeze toate numerele cu proprietatea că au numărul de cifre pare egal cu numărul de cifre impare.

Exemplu: dacă a=94 și b=1014

se va afișa 94 96 98 1001 1003 1005 1007 1009 1010 1012 1014.

```
#include <iostream>
using namespace std;
int a, b, i, aux, nrp, nri;
int main()
{
    cin>>a>>b;
    for(i=a; i<=b; i++)
    {
```

```
        aux=i;
        nrp=0;
        nri=0;
        while(aux!=0)
        {
            if(aux%2==0)
                nrp++;
            else nri++;
            aux=aux/10;
        }
        if(nrp==nri)
            cout<<i<<" ";
    }
    return 0;
}
```

Subiectul nr 17

Se citește un număr de la tastatură de cel mult 9 cifre. Să se afișeze la ecran numărul obținut prin dublarea fiecărei cifre mai mică decât 5.

Exemplu:

dacă se citește numărul 275043 se va afișa 475086.

```
#include<iostream>
using namespace std;
int x, y, p;
int main()
```

```
{
    cin>>x;
    p=1;
    while(x!=0)
    {
        if(x%10<5)
            y=y+(x%10)*2*p;
        else y=y+(x%10)*p;
        p=p*10;
        x=x/10;
    }
    cout<<y;
    return 0;
}
```



Subiectul nr 18

Se citește de la tastatură un număr n ($n \leq 100$) și apoi n elemente ale unui tablou unidimensional v . Să se insereze după fiecare număr prim exact două valori de 0 și să se afișeze pe ecran vectorul astfel modificat.

Exemplu: Pentru $n=7$ și elementele 55, 8, 37, 94, 101, 29, 15

se afișează 55, 8, 37, 0, 0, 94, 101, 0, 0, 29, 0, 0, 15

```
#include <iostream>
using namespace std;
int n, v[101];
bool ok;
int main()
{
    cin >> n;
    for (int i = 0; i < n; i++)
        cin >> v[i];
```

```
for (int i = 0; i < n; i++)
{ok = true;
    if (v[i] == 0 || v[i] == 1)
        ok = false;
for (int d = 2; d <= v[i] / 2; d++)
    if (v[i] % d == 0)
        ok = false;
if (ok)
{for (int j = n + 1; j > i + 2; j--)
    v[j] = v[j - 2];
v[i + 1] = v[i + 2] = 0;
n += 2;
i += 2;
    }
}
for (int i = 0; i < n; i++)
    cout << v[i] << " ";
return 0;
}
```

Subiectul nr 19

Se citește de la tastatură două numere naturale a și b ($a \leq 1000, b \leq 1000$).

Să se afișeze pe ecran toate numerele din intervalul $[a, b]$ pentru care suma divizorilor lor este un număr prim.

Se va realiza o funcție care calculează și returnează suma divizorilor unui număr natural și o altă funcție care verifică dacă un număr este prim.

Exemplu: pentru $a=10, b=70$, se vor afișa 16, 25, 64

```
#include <iostream>
using namespace std;
int sumaDiv(int n)
{
    int s=0;
    for(int d=1; d<=n; d++)
        if(n%d==0)
            s=s+d;
    return s;
```

```
bool estePrim(int n)
{
    bool ok = true;
    if (n == 0 || n == 1)
        ok = false;
    for (int d = 2; d <= n / 2; d++)
        if (n % d == 0)
            ok = false;
    return ok;
}
int main()
{
    int a, b;
    cin >> a >> b;
    for(int i=a; i<=b; i++)
        if(estePrim(sumaDiv(i)))
            cout << i << " ";
    return 0;
}
```

Subiectul nr 20

Se citește de la tastatură o matrice A cu m linii și n coloane ($m \leq 20, n \leq 20$).

Să se verifice dacă există linii care au toate elementele palindromuri. Se va afișa indicele liniei/liniilor.

Exemplu:

Pentru $m=4, n=5$

și matricea

12	1	9	1	5
121	1	0	33	141
18	1	91	1	5
88	828	11	8	4

Se afișează 2, 4

```
#include <iostream>
using namespace std;
int n, i, a[21][21], j, k, m, ok, aux, p;
int main()
{
```

```
cin >> m >> n;
for(i=1; i<=m; i++)
    for(j=1; j<=n; j++)
        cin >> a[i][j];
for(i=1; i<=m; i++)
{
    ok=1;
    for(j=1; j<=n; j++)
    {
        aux=a[i][j];
        p=0;
        while(aux)
        {
            p=p*10+aux%10;
            aux/=10;
        }
        if(p!=a[i][j])
            ok=0;
    }
    if(ok==1) cout << i << " ";
}
return 0;
}
```



Subiectul nr 21

Se citește de la tastatură o matrice A cu m linii și n coloane ($m \leq 20, n \leq 20$) cu elemente cifre ale sistemului zecimal (0, 1, ..., 9). Se cere să se afișeze pe ecran perechile de linii complementare (adică suma elementelor aflate pe aceeași coloană trebuie să fie 9)

Exemplu:

Pentru $m=4, n=5$ și matricea A

```
1 1 9 1 5
1 1 0 3 1
1 1 9 1 5
8 8 0 8 4
```

Se afișează (1,4), (3,4)

```
#include <iostream>
using namespace std;
```

```
int n,i,a[21][21],j,k,m;
int main()
{
    cin>>m>>n;
    for(i=1; i<=m; i++)
        for(j=1; j<=n; j++)
            cin>>a[i][j];
    for(i=1; i<m; i++)
        for(j=i+1; j<=m; j++)
        {
            int ok=1;
            for(k=1; k<=n; k++)
                if(a[i][k]+a[j][k]!=9)
                    ok=0;
            if(ok==1)
                cout<<"("<<i<<" "<<j<<" ",<<endl;
        }
    return 0;
}
```

Subiectul nr 22

Pe prima linie a fișierului text sir.in se găsește un cuvânt de exact 9 litere mici ale alfabetului englez. Pe a doua linie a fișierului se găsește o cifră nenulă k. Se cere să se afișeze pe ecran, de k ori, cel de al k-lea caracter de pe prima linie a fișierului sir.in.

Exemplu: dacă din fișier se citește

algoritmi

5

se va afișa pe ecran: rrrrr

```
#include <iostream>
#include <fstream>
using namespace std;
char s[10];
int k,i;
ifstream f ("sir.in");
int main()
{
    f>>s>>k;
    for(i=1; i<=k; i++)
        cout<<s[k-1]<<endl;
    return 0;
}
```

Subiectul nr 23

Se citește de la tastatură un text de maxim 50 caractere litere mici ale alfabetului englez. Să se afișeze textul obținut prin dublarea primelor k vocale (k citit de la tastatură). Dacă în text nu există k vocale se vor dubla toate vocalele.

Exemplu:

dacă textul citit este *voi lua nota maxima la bacalaureat* și $k=11$ se va afișa *vooi luuaa nootaa maaxiimaa laa baacalaureat*.

```
#include <iostream>
#include <cstring>
using namespace std;
char s[101], voc[]="aeiou";
int i, k, j;
int main()
```

```
{
    cin.get(s, 51);
    cin>>k;
    for(i=0; i<strlen(s) && k!=0; i++)
        if(strchr(voc,s[i])!=0)
        {
            for(j=strlen(s); j>i; j--)
                s[j]=s[j-1];
            i++;
            k--;
        }
    cout<<s<<endl;
    return 0;
}
```



Subiectul nr 24

Se citește de la tastatură două numere naturale a și b ($a \leq 1000, b \leq 1000$). Să se afișeze pe ecran toate numerele din intervalul [a,b] pentru care suma factorialilor cifrelor lor este mai mică sau egală cu 100.

Se va realiza o funcție care calculează și returnează factorialul unui număr natural și o altă funcție care returnează suma factorialilor cifrelor unui număr.

Exemplu:

pentru a=24, b=40, se vor afișa 24, 30, 31, 32, 33, 34, 40

```
#include <iostream>
using namespace std;
int fact(int c)
{
    int p=1;
    while(c>1)
    {
        p*=c;
        c--;
    }
}
```

```
}
return p;
}
int suma(int n)
{
    int s=0;
    while(n)
    {
        s+=fact(n%10); n/=10;
    }
    return s;
}
int main()
{
    int a, b;
    cin>>a>>b;
    for(int i=a;i<=b;i++)
        if(suma(i)<=100)
            cout<<i<<" ";
    return 0; }
}
```

Subiectul nr 25

Se citește o matrice pătratică A de dimensiune n ($n \leq 20$).

Să se afișeze la ecran matricea B formată din suma cifrelor fiecărui element din matricea A.

Exemplu: dacă matricea A este $n=5$ și

$$\begin{pmatrix} 2741 & 1199 & 27 & 13 & 5 \\ 22 & 6 & 7008 & 14 & 1999 \\ 45 & 46 & 202 & 201 & 33 \\ 1024 & 29 & 111 & 469 & 199 \\ 96 & 213 & 21 & 4096 & 16 \end{pmatrix}$$

se va afișa

$$\begin{pmatrix} 14 & 20 & 9 & 4 & 5 \\ 4 & 6 & 15 & 5 & 28 \\ 9 & 10 & 4 & 3 & 6 \\ 7 & 11 & 3 & 19 & 19 \\ 15 & 6 & 3 & 19 & 7 \end{pmatrix}$$

```
#include<iostream>
using namespace std;
int A[21][21],B[21][21], i, j, x, s, n;
int main()
{
    cin>>n;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
        {
            cin>>A[i][j];
            s=0;
            x=A[i][j];
            while(x!=0)
            {
                s=s+x%10;
                x=x/10;
            }
            B[i][j]=s;
        }
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n; j++)
            cout<<B[i][j]<<" ";
        cout<<endl;
    }
}
```

Subiectul nr 26

Se citește o matrice pătratică A de dimensiune n ($n \leq 20$).

Să se afișeze la ecran matricea B formată din cifrele de control ale fiecărui element din matricea A.

Exemplu: dacă matricea A este $n=5$ și

$$\begin{pmatrix} 2741 & 1199 & 27 & 13 & 5 \\ 22 & 6 & 7008 & 14 & 1999 \\ 45 & 46 & 202 & 201 & 33 \\ 1024 & 29 & 111 & 469 & 199 \\ 96 & 213 & 21 & 4096 & 16 \end{pmatrix}$$

se va afișa

$$\begin{pmatrix} 5 & 2 & 9 & 4 & 5 \\ 4 & 6 & 6 & 5 & 1 \\ 9 & 1 & 4 & 3 & 6 \\ 7 & 2 & 3 & 1 & 1 \\ 6 & 6 & 3 & 1 & 7 \end{pmatrix}$$

```
#include<iostream>
using namespace std;
int A[21][21],B[21][21], i, j, x, s, n;
int main()
{
    cin>>n;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
        {
            cin>>A[i][j];
            x=A[i][j];
            s=0;
            while(x>9)
            {
                while(x!=0)
                {
                    s=s+x%10;x=x/10;
                }
                x=s;s=0;
            }
            B[i][j]=x;
        }
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n; j++)
            cout<<B[i][j]<<" ";
        cout<<endl;
    }
}
```



Subiectul nr 27

Se citesc de la tastatură două șiruri de caractere cu maximum 250 de caractere fiecare (litere mici ale alfabetului englez), cu caracterele în ordine alfabetică, introduse de la tastatură. Se cere să se afișeze pe ecran un al treilea șir, format din toate caracterele primelor două, așezate în ordine alfabetică.

Exemplu: dacă se citesc $sir1="ampstz"$ și $sir2="bfgostx"$ se va afișa $abfgmopsstxz$.

```
#include <iostream>
#include <cstring>
using namespace std;
char s1[251],s2[251],s3[251],i,j;
int n1,n2,n3;
int main()
{
    cin>>s1>>s2;
    n1=strlen(s1); n2=strlen(s2);
    while(i<n1 && j<n2)
```

```
{
    if(s1[i]<s2[j])
    {
        s3[n3]=s1[i];
        n3++; i++;
    }
    else
    {
        s3[n3]=s2[j];
        n3++; j++;
    }
}
while(i<n1)
{
    s3[n3]=s1[i];
    n3++; i++; }
while(j<n2)
{
    s3[n3]=s2[j];
    n3++; j++;
}
cout<<s3;
return 0;
}
```

Subiectul nr 28

Se citește din fișierul *date.in* o matrice A cu m linii și n coloane ($m \leq 20$, $n \leq 20$). Să se construiască și să se afișeze în fișierul *date.out* matricea B obținută prin rotirea cu 90° în sensul acelor de ceasornic a matricei A.

Exemplu: dacă $m=3$ și $n=5$ și matricea A este

$$\begin{pmatrix} 7 & 4 & 2 & 3 & 7 \\ 9 & 6 & 5 & 1 & 6 \\ 4 & 7 & 2 & 5 & 9 \end{pmatrix}$$

matricea B obținută va fi

$$\begin{pmatrix} 4 & 9 & 7 \\ 7 & 6 & 4 \\ 2 & 5 & 2 \\ 5 & 1 & 3 \\ 9 & 6 & 7 \end{pmatrix}$$

```
#include <fstream>
using namespace std;
ifstream f("date.in");
ofstream g("date.out");
int A[21][21], B[21][21], m, n, i, j;
int main()
{
    f>>m>>n;
    for(i=1; i<=m; i++)
        for(j=1; j<=n; j++)
            f>>A[i][j];
    for(i=1; i<=n; i++)
        for(j=1; j<=m; j++)
            B[i][j]=A[m+1-j][i];
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=m; j++)
            g<<B[i][j]<<" ";
        g<<endl;
    }
    return 0;
}
```

Subiectul nr 29

Se citește un număr de la tastatură de cel mult 9 cifre. Să se afișeze pe ecran cifrele care nu apar în numărul n.

Exemplu: $n=29225$ se afișează 0, 1, 3, 4, 6, 7, 8.

```
#include <iostream>
using namespace std;
int n, fr[11];
int main()
{
```

```
    cin>>n;
    while(n)
    {
        fr[n%10]++;
        n/=10;
    }
    for(int i=0;i<10;i++)
    {
        if(!fr[i])
            cout<<i<<" ";
    }
    return 0;
}
```



Subiectul nr 30

Se citesc de la tastatură un număr n ($n \leq 100$) și apoi n elemente ale unui tablou unidimensional v . Să se insereze înaintea fiecărui element pătrat perfect rădăcina sa pătrată și să se afișeze pe ecran vectorul astfel modificat.

Exemplu: $n=5$ și elementele 49, 64, 37, 94, 100 se afișează 7, 49, 8, 64, 37, 94, 10, 100

```
#include <iostream>
#include <cmath>
using namespace std;
int n,i,v[101],j;
int main()
{
    cin>>n;
    for(i=1; i<=n; i++)
```

```
    cin>>v[i];
    for(i=1; i<=n; i++)
    {
        if(sqrt(v[i])==int(sqrt(v
[i])))
        {
            n++;
            int p=i;
            for(j=n; j>p; j--)
                v[j]=v[j-1];
            v[p]=sqrt(v[i]);
            i++;
        }
    }
    for(i=1; i<=n; i++)
        cout<<v[i]<<" ";
    return 0;
}
```

Subiectul nr 31

Se citesc de la tastatură un număr n ($n \leq 100$) și apoi n elemente ale unui tablou unidimensional v . Să se afișeze pe ecran perechile de numere citite consecutiv cu proprietatea că cel de-al doilea număr al perechii reprezintă pătratul numărului de apariții ale cifrei 1 în primul.

Exemplu: $n=7$ și numerele 121, 4, 89, 7, 0, 111, 9 se afișează (121,4) (7,0) (111,9)

```
#include <iostream>
using namespace std;
int n,p,u,aux,c;
int main()
```

```
{
    cin>>n;    cin>>p;
    for(int i=2; i<=n; i++)
    {
        cin>>u;
        aux=p;
        c=0;
        while(aux)
        {
            if(aux%10==1) c++;
            aux/=10;
        }
        if(c*c==u)
            cout<<"("<<p<<","<<u<<") ";
        p=u;
    }
    return 0;
}
```

Subiectul nr 32

Se citesc de la tastatură două numere naturale a și b ($a \leq 1000, b \leq 1000$). Să se afișeze pe ecran toate numerele din intervalul $[a,b]$ care sunt prime cu suma cifrelor lor. Se va realiza o funcție care calculează și returnează suma cifrelor unui număr natural și o altă funcție care returnează c.m.m.d.c. a două numere naturale.

Exemplu: pentru $a=50, b=60$, se vor afișa 52, 53, 56, 58, 59.

```
#include <iostream>
using namespace std;
int a,b,i;
int suma(int x)
{
    int s=0;
    while(x)
    {
```

```
        s+=x%10;
        x/=10;
    }
    return s;
}
int cmmdc(int a,int b)
{
    if(a<b)
    {
        int aux=a;
        a=b;
        b=aux;
    }
    while(b!=0)
    {
        int c=a%b;
        a=b;
        b=c;
    }
    return a;
}
```



<p>Subiectul nr 33</p> <p>Se citește de la tastatură un text format din cuvinte separate între ele prin câte un spațiu. Fiecare cuvânt are cel mult 40 de caractere, doar litere mici ale alfabetului englez. Textul are cel mult 200 de caractere. Să se afișeze pe ecran, pe aceeași linie separate printr-un spațiu, doar cuvintele din textul citit care conțin cel mult trei vocale. Se consideră vocale: a, e, i, o, u.</p> <p>Exemplu: dacă textul este: pentru examenul de atestat se folosesc tablouri se afișează pe ecran: pentru de atestat se folosesc</p> <pre>#include <iostream> #include <cstring> using namespace std; char s[201], *p, voc[]="aeiou"; int v; int main() {</pre>	<pre>cin.get(s,201); p=strtok(s," "); while(p) { v=0; bool ok=true; for(int i=0; i<strlen(p) &&ok==true; i++) if(strchr(voc,p[i])!=0) { v++; if(v>3) ok=false; } if(ok==true) cout<<p<<" "; p=strtok(NULL," "); } return 0; }</pre>
<p>Subiectul nr 34</p> <p>Se citește din fișierul date.in un număr natural n cu cel mult 10 cifre. Să se formeze un nou număr obținut prin eliminarea cifrei maxime din numărul inițial (dacă cifra maximă apare de mai multe ori, se vor elimina toate aparițiile) și să se afișeze pe ecran noul număr.</p> <p>Exemplu: dacă $n=296929013$ se afișează 262013 dacă $n=333$ se afișează 0</p> <pre>#include <iostream> #include <fstream> using namespace std; ifstream f("date.in"); int x; int cifra_max(int x) { int c=0; while(x) { if(x%10>c) c=x%10; x/=10; }</pre>	<pre> } return c; } int sterge_cifra_max(int x,int c) { int nr=0,p=1; while(x) { if(x%10!=c) { nr=nr+p*(x%10); p*=10; } x/=10; } x=nr; return x; } int main() { f>>x; int c=cifra_max(x); cout<<sterge_cifra_max(x,c); return 0; }</pre>
<p>Subiectul nr 35</p> <p>Se citesc un număr n ($n \leq 100$) și apoi n elemente ale unui tablou unidimensional v. Să se elimine din tablou elementele care au cifra de control egală cu o valoare a. Să se afișeze pe ecran tabloul rezultat.</p> <p>Exemplu: dacă $n=12$ și elementele 34 1109 56 875 285 531 1190 11 98 287 444 41 și $a=2$ se va afișa 34 285 531 98 287 444 41.</p> <pre>#include <iostream> using namespace std; int n, v[101], a, i, j, cc; int main() { cin>>n; for(i=1; i<=n; i++) cin>>v[i]; cin>>a;</pre>	<pre> for(i=1; i<=n; i++) { if(v[i]%9==0) cc=9; else cc=v[i]%9; if(cc==a) { for(j=i; j<n; j++) v[j]=v[j+1]; n--; i--; } } for(i=1; i<=n; i++) cout<<v[i]<<" "; return 0; }</pre>



Subiectul nr 36

Se citește un număr de la tastatură de cel mult 9 cifre. Să se afișeze pe ecran cel mai mare număr natural strict mai mic decât n care are toate cifrele egale între ele.

Exemplu: n=356 se afișează 333

```
#include <iostream>
using namespace std;
int n;
int main()
{   cin>>n;
    int pc=n,nrc=1;
    while (pc>9)
    {
        nrc++; pc/=10;
    }
    int aux=n,c=0;
    while (aux)
```

```
{   if (pc>aux%10)
    {   pc--; aux=0; }
    else if (pc==aux%10) c++;
    aux/=10;
}
if (c==nrc) pc--;
if (pc==0)
{
    pc=9; nrc--;
}
if (nrc==0) cout<<"0";
while (nrc)
{
    cout<<pc;
    nrc--;
}
return 0;
}
```

Subiectul nr 37

Se citește un număr n<100. Să se afișeze primele n numere cu următoarea proprietate: sunt fie puteri ale lui 2, fie puteri ale lui 3.

Exemplu: dacă n=12 se va afișa 1 2 3 4 8 9 16 27 32 64 81 128.

```
#include <iostream>
using namespace std;
int n,x,k,aux1,aux;
int main()
{
    cin>>n;
    x=1;
    while (k<n)
    {
```

```
        aux=x;
        while (aux%2==0)
            aux=aux/2;
        aux1=x;
        while (aux1%3==0)
            aux1=aux1/3;
        if (aux==1 || aux1==1)
        {
            cout<<x<<" ";
            k++;
        }
        x++;
    }
    return 0;
}
```



Subiectul nr 39

Se citește un număr $n < 100$. Să se afișeze primele n numere cu următoarea proprietate: la trecerea în baza 2 numărul valorilor de 1 este dublul valorilor de 0.

Exemplu: dacă $n=20$ se va afișa 2 4 5 6 8 9 10 11 12 13 14 16 17 18 19 20 21 22 23 24.

```
#include <iostream>
using namespace std;
int n,x,k,aux, unu, zero;
int main()
{   cin>>n;
    x=1;
    while(k<n)
    {   aux=x; unu=0; zero=0;
        while(aux!=0)
```

```
        {   if(aux%2==1) unu++;
            else zero++;
            aux=aux/2;
        }
        if(unu==2*zero)
        {   cout<<x<<" ";
            k++; }
        x++;
    }
    return 0; }
```

Subiectul nr 40

Se citesc un număr n ($n \leq 100$) și apoi n elemente ale unui tablou unidimensional v . Să se ordoneze crescător elementele impare, iar elementele pare trebuie să rămâne pe pozițiile inițiale. Să se afișeze tabloul ordonat.

Exemplu: dacă $n=12$ și elementele 34 1109 56 879 285 531 1190 17 98 287 444 41 se va afișa 34 17 56 41 285 287 1190 531 98 879 444 1109.

```
#include <iostream>
using namespace std;
int n, v[101], aux, i, j;
int main()
{   cin>>n;
    for(i=1; i<=n; i++)cin>>v
    [i];
```

```
    for(i=1; i<n; i++)
        for(j=i+1; j<=n; j++)
            if(v[i]%2==1
            && v[j]%2==1 && v[i]>v[j])
                {   aux=v[i]; v[i]=v
                [j]; v[j]=aux; }
    for(i=1; i<=n; i++)
        cout<<v[i]<<" ";
    return 0;
}
```



Subiectul 1

Vizualizați capacitatea hard disc-ului PC-ului pe care lucrați. Utilizând instrumentul Snipping Tool, realizați o captură a informațiilor cerute și salvați-o cu numele **HDD.png**, pe desktop în folder-ul **Atestat_mate_info 2022**.

Într-un fișier text, precizați numărul de partiții existente pe HDD, numele acestora și spațiul liber disponibil pe fiecare partiție. Salvați fișierul cu numele **Partiții.txt** în folder-ul **Atestat_mate_info 2022**.

Subiectul 2

Creați pe spațiul de lucru (desktop) un folder cu numele **Atestat_mate_info 2022**. Realizați în directorul creat anterior o comandă rapidă (shortcut) către programul Paint. Creați două subfoldere cu numele *Probe practice* și *Probe teoretice* care să conțină cel puțin două documente de lucru de tip *txt*.

Subiectul 3

Creați pe desktop un folder numit **Atestat_mate_info 2022** și subfolderele **Hardware**, **Software** și **Întreținere**, subordonate acestuia.

- În subfolderul **Hardware**, amplasați un fișier de tip docx și în acesta scrieți detalii legate de structura și performanțele PC-ului tău.
- În subfolderul **Software_PC**, creați un fișier de tip docx în care enumerați cele mai relevante aplicații instalate în PC-ul vostru.
- În subfolderul **Întreținere**, creați o listă a lucrărilor de întreținere preventivă, pe care vă propuneți să le executați la PC-ul vostru.

Subiectul 4

Căutați pe Internet specificațiile tehnice pentru instalarea unui S.O. de tip Windows 10. Identificați-le și comparați-le cu cele existente pe PC-ul dumneavoastră.

Realizați o captură utilizând instrumentul Snipping Tool, și salvați-o cu numele **Resurse hardware.png**, pe desktop în folder-ul **Atestat_mate_info 2022**.

Subiectul 5

Folosind aplicația Task Manager și instrumentul Snipping Tool, realizați o imagine care să prezinte performanțele calculatorului vostru vizând: procesorul, memoria, hard-discul, placa video.

Salvați imaginea cu numele **Verificare-performanțe.gif**, pe desktop în folder-ul **Atestat_mate_info 2022**.

Subiectul 6

În directorul **Atestat_mate_info 2022** realizați un document Microsoft Word cu numele **regiune**.

- * Realizați următoarele setări de pagină: format A4, margini normal, orientare pagină pe verticală (Portret), tema pagină tip Adiere.
- * Creați o listă marcată cu imagini/simboluri a principalelor atracții ale regiunii Dobrogea (lacul Iacobdeal, Cetatea Troesmis, Munții Măcinului, mănăstirile Izvorul Tămăduirii, Cocos, Celic Dere, lacul Razelm și Murighiolul, Cetatea Enisala, Capul Doloșman și Cetatea Argamum, Cheile Dobrogei, Mănăstirea Casian, Lacul Casian și peștera Sfântului Casian).



Subiectul 7

În directorul **Atestat_mate_info 2022** creați o ofertă de turism pentru o vacanță la *Cabana Trei Brazi* folosind aplicația Microsoft Word.

Documentul creat va avea următoarele setări de pagină: format A4, margini: sus – 2.25 cm, jos – 2.25 cm, stânga – 2.25 cm, dreapta – 2.55 cm, antet 1.34, subsol 1.50 cm, orientare pagină pe verticală (Portret), bordura de tip Art – 31 pt- cu brazi.

Adăugați în document următoarele informații:



Pensiune	Preț cazare (lei/noapte)	Număr camere disponibile pensiune
Pensiunea Doi Brazi	200	8
Pensiunea Trei Brazi	250	10

Deasupra tabelului, adaugă titlul „**Ofertă Cabana Trei Brazi**”, font **Tahoma**, dimensiune **16**, **îngroșat**, **centrat** și **subliniat** cu linie **ondulată** de culoare **verde**

Inserează, sub tabel, un desen simbolic, la fel cu cel din model, folosind diverse **forme automate**. Grupează formele pentru a obține un singur obiect grafic.

Subiectul 8

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D	E	F	G
1	Nr. Crt	Nume	Prenume	Data nașterii	Vârsta	Data înscrierii	Clasa
2	1	Velicu	Cosmina	3/12/2014		25/8/2020	
3	2	Cristea	Denis	11/11/2008		11/8/2015	
4	3	Nedelcu	Corneliu	5/7/2002		5/9/2009	
5	4	Vicol	Leonard	11/9/2005		11/9/2012	
6	5	Ionescu	Gina	29/6/2009		29/8/2016	
7	6	Petrea	Sandra	30/6/2011		30/8/2018	

Completați coloana **Vârsta**, scriind formula corespunzătoare de calcul.

Calculați **Clasa** în care este fiecare elev în funcție de data înscrierii la școală (presupunând că nu a repetat niciun an) și formați astfel încât dacă clasa e mai mare sau egală cu 9 să se afișeze automat cu bold, culoare: roșu.

Subiectul 9

Pentru promovarea activității extrașcolare „**Concurs table 2022**” realizează folosind aplicația Microsoft Word un anunț care să cuprindă o parte din triunghiurile roșii și negre ale unui joc de table. Pe triunghiurile roșii așezați două piese de culoare neagră cu margini galbene. Concursul are loc în luna mai în „Săptămâna Altfel” și este la a treia ediție. Salvați documentul în directorul **Atestat_mate_info 2022** cu denumirea **concurs_table**.



Subiectul 10

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D	E	F	G
1	Nr.crt	Proprietar	Județ	Indicativ județ	Autoturism	An fabricație	Stare
2	1	Bistrițeanu Miron			Logan	1992	
3	2	Bogatu Florica			Mercedes	2012	
4	3	Dumitrache Marian			Skoda	1987	
5	4	Deleanu Eusebiu			Ford	2009	
6	5	Olteanu Camelia			Dacia	2016	
7	6	Pană Mircea			Audi	2005	
8							
9							
10	COD	JUDET					
11	B	București					
12	AB	Alba					
13	AR	Arad					
14	AG	Arges					
15	BC	Bacău					
16	BH	Bihor					
17	BN	Bistrița-Năsăud					
18	BT	Botoșani					
19	BV	Brașov					
20	BR	Brăila					
21	BZ	Buzău					
22	CS	Caras-Severin					
23	CL	Călărași					

- ☞ Completați coloanele **Județ** și **Indicativ județ**, prin preluarea informațiilor din listele de validare **JUDEȚ** și **COD**.
- ☞ În rubrica **Stare**, afișați ”vechi” dacă anul de fabricație este ≤ 1995 și ”OK” pentru anul de fabricație mai mare de 1995.

Subiectul 11

În directorul **Atestat_mate_info 2022**, creat de dumneavoastră, realizați o prezentare Microsoft Office Power Point care să conțină 2 diapositive (slide-uri).

- ☞ Primul diaporitiv are aspectul Titlu (Layout):
 - ⇒ Introduceți titlul ROMANE scris înclinat cu fontul Tahoma și mărimea 50 de puncte;
 - ⇒ Aplicați un efect de animație de intrare titlului;
 - ⇒ Desenați sub titlu o linie dublă, de culoare roșie și grosime 5 puncte.
- ☞ Al doilea diaporitiv are aspectul Titlu și Conținut (Title and Content):
 - ⇒ Adăugați titlul CLASIFICARE ROMANE și o listă cu marcatori care să conțină următoarele categorii de romane: Roman realist, Roman naturalist, Roman suprarrealist, Roman avangardist, Roman realist -socialist, Roman existențialist, Roman structuralist, Noul roman francez, Roman postmodern, Roman interactiv (pe suport electronic).

Subiectul 12

Realizați o prezentare Power Point cu denumirea ”Forme automate spațiale” pe care o salvați prezentarea în directorul **Atestat_mate_info 2022**. Informațiile din prezentare sunt:

- ☞ 2 diapositive (slide-uri) și informații referitoare la patru forme automate: soare (sun), stea (5- point star), nor (cloud) și luna (moon).
- ☞ Creați un decor intuitiv grupat din aceste forme automate.
- ☞ Se va aplica un efect de animație de intrare și un efect de umplere de tip gradient (Gradient fill), fiecărei forme automată (shapes).



Subiectul 13

Căutați toate fișierele din contul curent, care conțin în denumire cuvântul *atestat* și sunt de tipul *docx*. Copiați primele 3 fișiere găsite, într-un director din contul curent pe care îl veți denumi **Atestat_mate_info 2022**. Creați subdirectorul **Fisiere Atestat**. Creați pe desktop, un shortcut la acest director.

Realizați o captură de ecran care să ilustreze stabilirea condițiilor de căutare a fișierelor text și salvați-o cu numele Căutare.jpg, în directorul **Fisiere atestat**.

Subiectul 14

Folosind aplicația Microsoft Excel realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop

	A	B	C	D	E
1	Marca	Nume	Prenume	Data nasterii	Salariu
2	7305	Dandiș	Maria	3/1/1957	3000
3	7306	Marinescu	Alexia	10/4/1969	2900
4	7307	Georgescu	Dan	3/5/1970	2100
5	7308	Mihalea	Emilian	3/20/1978	2500
6	7309	Stancu	Ivan	2/9/1975	2000
7	7310	Alexandrescu	Georgiana	6/5/1980	1500

☞ Trasați un grafic care să prezinte comparativ salariile persoanelor din tabelul anterior. Adăugați graficului următoarele elemente: titlul: **Salarii**, etichete de axe: **Nume** și **Salariu**, etichete de date.

☞ În aceeași foaie de calcul, începând cu celula B20 să se afișeze informațiile referitoare la angajații cu salarii cuprinse între 2200 și 4000 lei.

Subiectul 15

Folosind aplicația Microsoft Excel realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D	E	F	G
1	ID	Nume	DataN	Profesie	Orasul natal	Stare civila	Salariul
2	1	Lefterescu	09.12.1952	Antreprenor	Constanța	căsătorit	8500
3	2	Paunescu	11.12.1977	Psiholog	Iași	divorțat	3400
4	3	Florescu	15.07.1969	Inginer navalist	Constanța	văduv	6700
5	4	Mihaescu	25.03.1995	Electrician	Arad	necăsătorit	2800
6	5	Andronescu	03.08.1980	Veterinar	Sibiu	căsătorit	4800
7	6	Iliescu	25.03.1996	Brutar	Constanța	necăsătorit	3600
8	7	Udrescu	07.08.1981	Profesor	Constanța	căsătorit	4400
9	8	Radulescu	20.07.1955	Cardiolog	Suceava	căsătorit	12800
10	9	Barbulescu	13.04.1979	Economist	Constanța	căsătorit	7300
11	10	Codrescu	17.10.1991	Stomatolog	Iași	necăsătorit	14500

☞ Sortați datele din tabelul de mai sus, în ordine alfabetică ascendentă după **Orașul natal** iar pentru cei care sunt din același oraș, după **Nume**, ascendent.

☞ Filtrați datele din tabelul de mai sus, astfel încât să fie afișate doar persoanele ce au salariul cuprins între 5000 și 13000.



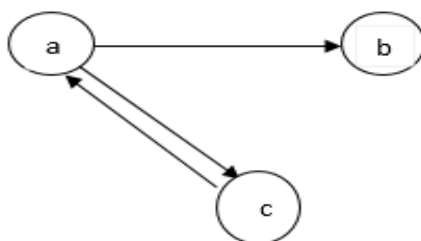
Subiectul 16

- ☞ Scanați selectiv două foldere de pe partiția la care aveți acces, folosind aplicația antivirus instalată pe computerul vostru. Realizați o captură de ecran cu setările stabilite pentru scanare și una cu raportul obținut în urma scanării.
- ☞ Salvați cele două imagini în folder-ul **Atestat_mate_info 2022**, creat pe desktop.

Subiectul 17

În directorul **Atestat_mate_info 2022** realizați un document Microsoft Word cu numele **graf_orientat**.

- ☞ Realizați următoarele setări de pagină: format A4, margini moderat, orientare pagină pe verticală (Portret).
- ☞ Să se deseneze graful orientat cu trei vârfuri etichetate cu litere de la a la c din figura alăturată. Folosind o funcție realizați o răsturnare orizontală la 90°a desenului.



Subiectul 18

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D	E	F	G	H	I	J
1	Nr.crt	Nume	Prenume	Județ	Proba1	Proba2	Proba 3	Media	Rezultat	
2	1	Bobeică	Eugenia	Arad	92.50	64.5	88.00			
3	2	Scînteianu	Claudiu	Botoșani	57.00	54.5	60.00			
4	3	Drăgoi	Victor	Brăila	42.00	10.00	94.50			
5	4	Damian	Elena	Arad	70.00	66.50	75.00			
6	5	Orzan	Nicoleta	Giurgiu	72.00	39.00	80.00			
7	6	Panait	Cezar	Arad	83.50	63.00	79.00			
8										
9										
10		Județe								
11		București								
12		Alba								
13		Arad								
14		Arges								
15		Bacău								

- ☞ Completați coloana **Media**, scriind formula corespunzătoare de calcul și afișând rezultatul cu primele două zecimale, fără rotunjire (aproximare).
- ☞ În rubrica **Rezultat**, afișați ”Calificat” dacă media este cel puțin 70 și ”Respins” pentru medii mai mici de 70.



Subiectul 19

Realizați o prezentare Power Point, cu tema “Liceul meu” și salvați fișierul cu numele liceul_meu.pptx în directorul **Atestat_mate_info 2022**, de pe desktop.

Prezentarea va fi compusă din două diapozitive:

- Diapozitivul 1 cuprinde numele clasei, numele liceului și numele profesorului diriginte; -
- Diapozitivul 2 cuprinde o listă cu marcatori de tipul ☺ (culoare roșu), care să conțină top 5 - discipline preferate studiate în liceu;
- Aplicați, pentru cele două diapozitive, efectul de tranziție dorit.

Subiectul 20

În directorul **Atestat_mate_info 2022** creați orarul clasei dumneavoastră într-un document word

- Aplicați următoarele setări de pagină: format A4, margini: sus – 2.25 cm, jos – 2.25 cm, stânga – 2.25 cm, dreapta – 2.55 cm, antet 1.34, subsol 1.50 cm, orientare pagină pe verticală (Portret);
- Subsolul să conțină: denumirea liceului, data și ora inserate automat, numărul paginii.
- Folosiți pentru textul din document fontul de tip Courier New, dimensiunea 10; culoarea verde, stilul cursiv.

Subiectul 21

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D	E	F	G	H	I
1	ID	Nume	DataN	Profesie	Orarul natal	Stare civila	Salariul		
2	1	Ionescu	14.12.1970	Chimist	București	căsătorit	2200		
3	2	Vasilescu	05.09.1960	Farmacist	Bacău	căsătorit	3400		SUM
4	3	Radulescu	02.03.1990	Biolog	Constanța	divorțat	4200		COUNT
5	4	Antonescu	06.11.1972	Secretar	Hunedoara	necăsătorit	3600		AVERAGE
6	5	Marinescu	01.08.1975	Actor	Constanța	căsătorit	5000		MIN
7	6	Popescu	15.08.2002	Muzician	București	căsătorit	4600		MAX
8	7	Albescu	16.09.1997	Balerin	București	necăsătorit	3800		
9	8	Craciunescu	21.10.1994	Fotograf	Bârlad	văduv	6000		
10	9	Enescu	05.05.1958	Inginer	Iași	necăsătorit	6500		
11	10	Georgescu	03.12.1944	Medic	Vaslui	căsătorit	13000		
12									
13									
14		ORAȘ			Stare civilă				
15		București			căsătorit				
16		Arad			necăsătorit				
17		Bacău			divorțat				
18		Brașov			văduv				
19		Brăila							
20		Buzău							
21		Călărași							
22		Constanța							

- Completați coloanele **Orașul natal** și **Stare civilă**, prin preluarea informațiilor din listele de validare **Oraș** și **Stare civilă**.
- Folosind funcțiile corespunzătoare, determinați fondul de salarii necesar, numărul de salarii, salariul mediu, maxim și cel minim



Subiectul 22

Folosind aplicația Microsoft Word realizați o invitație pentru promovarea activității extrașcolare ”**Dăruiește o carte**” care să conțină motto-ul: ”*Citește! E una dintre cele mai frumoase aventuri pe care le poți trăi. – Lloyd Alexander*”.

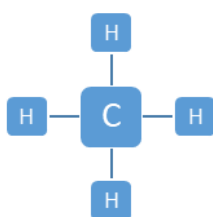
- ☞ Încadrați textul motto-ului într-un chenar cu bordură de tip **linie dublă, culoare albastră, grosime 3**.
- ☞ Adăugați o imagine sugestivă care să fie aliniată la stânga textului.
- ☞ Salvați documentul în directorul **Atestat_mate_info 2022** cu denumirea **invitație**.

Subiectul 23

În directorul **Atestat_mate_info 2022** creați o scurtă prezentare a parcului Tivoli. Utilizați informații din Wikipedia (https://ro.wikipedia.org/wiki/Parcul_Tivoli) într-un document word cu următoarele setări de pagină: dimensiune hârtie – A4, margini: sus 1 cm, jos 1, 20 cm, stânga - 2 cm, dreapta – 2.50 cm, antet 2 cm, subsol 1.45 cm, orientare - tip portret. Textul utilizat să fie așezat în două coloane.

Subiectul 24

În aplicația Microsoft Office PowerPoint creați o prezentare care să conțină un singur diapozitiv cu fundalul de tip textură care să cuprindă formula metanului – CH₄. Salvați prezentarea în directorul **Atestat_mate_info 2022**



Subiectul nr. 25

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D	E	F	G
1	Nr. crt.	Denumire	Preț	Nr bucăți	Preț total	Nr medicamente vândute	Stoc
2	1	Paracetamol	2.7	100			88
3	2	Algocalmin	9.17	70			50
4	3	Septosol	13.5	90			20
5	4	Nurofen	24.7	65			10
6	5	Biseptol	34.65	50			25
7	6	Tertensif	28.75	45			36
8	7	Prestarium	29.5	50			28

- ☞ Pentru medicamentele din tabelul de mai sus, calculați **Preț total** și numărul de produse rămase în **stoc**. Sortați-le în ordine alfabetică ascendentă după **Denumire**.
- ☞ Reprezentați grafic informațiile din tabel sub formă de diagramă pie, arătând contribuția procentuală a fiecărui medicament, în valoarea totală a produselor.



Subiectul 26

Să se creeze o prezentare cu aplicația MS Office PowerPoint cu titlul ”**Om de zăpadă.ppsx**” îndeplinind cerințele următoare:

- ☞ Primul diapozitiv va conține titlul ”IARNA – anotimp al albului infinit” scris cu fontul Bookman Old Style, dimensiune 25, culoare mov.
- ☞ Al doilea diapozitiv va conține cel puțin 5 forme de aspecte și forme diferite care să alcătuiască un om de zăpadă. Să se aplice pe aceste forme efecte de animație de intrare diferite setate astfel încât să apară singure una după cealaltă. Salvați prezentarea în directorul **Atestat_mate_info 2022**.

Subiectul 27

Creați o prezentare „**Festival film**” cu 3 diapozitive care să conțină

- ☞ În primul diapozitiv: titlul unui film și o imagine sugestivă din filmul ales
- ☞ În celelalte două diapozitive prezentarea va cuprinde date generale despre film: informații despre cel mai bun film al anului 2021 și regizorul lui.
- ☞ Aplicați prezentării tranziții între diapozitive și animații la alegere.
- ☞ Salvați prezentarea în directorul **Atestat_mate_info 2022**.

Subiectul 28

În directorul **Atestat_mate_info 2022** realizați un document Microsoft Word cu numele **statistica_tari**.

- ☞ Realizați următoarele setări de pagină: format A4, margini normal, orientare pagină pe verticală (Portret).
- ☞ Inserați un tabel cu 4 linii și 4 coloane. Creați un chenar tabelului: linie dublă de culoare roșie pentru exterior, de grosime 3 pct, linie punctată de culoare albastru pentru interior, de grosime 6 pct și dați o culoare de fundal tabelului. Completați primul rând din tabel cu **Țară, Limbă, Naționalitate, Populație** și apoi salvați-l.
- ☞ Realizați un grafic de tip coloană pentru țara și populația asociată acestuia.

Subiectul 29

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D
1	Nr.crt	Denumire produs	Data expirării	Stare produs
2	6	Detrical	7 august 2021	Expirat
3	5	Indapamid	6 mai 2024	În termen
4	2	Molekin Imuno	23 august 2023	În termen
5	1	Paracetamol	5 mai 2021	Expirat
6	3	Prestarium	12 octombrie 2021	Expirat
7	4	Tertensif	24 decembrie 2025	În termen

☞ Scrieți o formulă pe baza căreia să completați automat coloana *Stare produs* cu mesajul ”*Expirat*” dacă data expirării este mai mică decât data curentă sau ”*În termen*” dacă data expirării este mai mare decât data sistemului. Evidențiați prin aplicarea unui fundal ocru, toate situațiile în care produsele sunt expirate.



Subiectul 30

În directorul **Atestat_mate_info 2022** realizați un document Microsoft Word cu numele **carte_vizita** cu două pagini. : sus – 2.5 cm, jos – 2.05 cm, stânga – 2.55 cm, dreapta – 2.25 cm, antet 1.5, subsol 1.5 cm, orientare pagină pe verticală (Portret). Prima pagină să conțină două forme automate de bază grupate. Pe pagina 2 realizați următoarea carte de vizită

S.C. TONIOTES S.R.L.

AGENȚIE IMOBILIARĂ

Vânzări Cumpărări Închirieri Intermedieri
Apartamente Case Terenuri Spații comerciale

Str.Vișinului Nr. 1 Sc. I, Ap.1
0236123123

Tel/Fax: +4

Primele două rânduri ale cărții de vizită vor fi scrise cu Arial – 12, următoarele două cu Arial – 11, iar ultimele două cu Arial – 8.

Subiectul 31

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D	E	F	G
1	Nr. crt.	Cosmetice	Preț	Nr bucăți	Preț total	Nr.cosmetice vândute	Stare stoc-uri
2	7	Parfum	150.3	50		28	
3	5	Ruj	84	50		25	
4	3	Vopsea păr	56.5	90		20	
5	2	Cremă călcâie	32.2	70		50	
6	6	Deodorant	31.1	45		36	
7	4	Șampon	18	65		65	
8	1	Cremă mâini	11.5	100		88	

☞ Calculați **Preț total**, pe baza unei formule stabilite de voi. În coloana **Stare stoc-uri**, afișați mesajul “**Stoc epuizat**” dacă $Nr\ bucăți - Nr.\ cosmetice\ vândute = 0$ și “**În stoc**” dacă $Nr\ bucăți - Nr.\ cosmetice\ vândute > 0$

☞ Pe baza datelor din coloanele **Cosmetice** și **Preț** creați o diagramă de tip coloană. Titlul pentru diagramă: “Total cosmetice”. Titlul va fi scris cu fontul Arial, dimensiunea 14, culoarea albastră. Legenda să fie scrisă în partea de jos a diagramei.

Subiectul 32

În directorul **Atestat_mate_info 2022** realizați un document Microsoft Word cu numele **formule_mate**.

☞ Realizați următoarele setări de pagină: format A4, margini: sus – 2.5 cm, jos – 2.05 cm, stânga – 2.55 cm, dreapta – 2.25 cm, antet 1.5, subsol 1.5 cm, orientare pagină pe verticală (Portret), bordura de tip Artă la alegere, tema pagină tip Alegere.

☞ Editați următorul text fără a folosi editorul de ecuații:

Aria unui **pătrat** de latură L este dată de formula:

$$A_{\square} = L^2;$$

Aria unui **dreptunghi** de lungime L și lățime l este dată de formula:

$$A_{\square} = L * l;$$

Aria unui **cerc** de rază R este dată de formula:

$$A_{\bigcirc} = \pi R^2;$$



Subiectul 33

Realizați o prezentare Power Point, compusă din trei diapozitive (slide-uri) în care să prezentați obiective turistice din România.

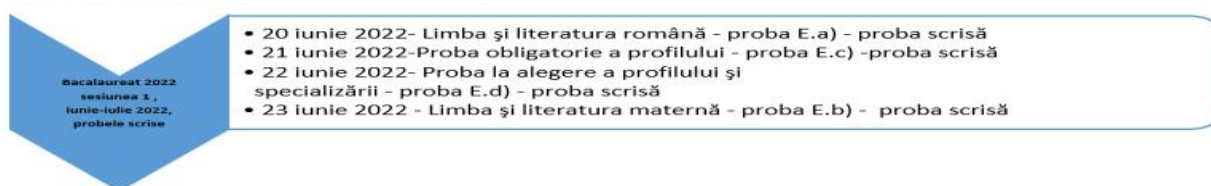
- ↖ Primul diapozitiv va fi de tip titlu și va conține titlul „România – obiective turistice”.
- ↖ În celelalte două diapozitive veți adăuga informații și imagini sugestive ale obiectivelor turistice alese
- ↖ Adăugați un fundal de tip gradient pentru toate diapozitivele având următoarele formatări: de sus în jos culorile roșu, galben și albastru, cu transparența culorilor 50%.
- ↖ Stabiliți orientarea diapozitivelor (slide-urilor) în tip portret.
- ↖ Salvați prezentarea în directorul **Atestat_mate_info 2022**.

Subiectul 34

Creați pe spațiul de lucru (desktop) un folder cu numele **Atestat_mate_info 2022**. Realizați în directorul creat anterior o arhivă cu numele **Subiecte_teorie.rar** și un document care să cuprindă imaginea desktop-ului (print screen) salvat cu numele **image_desktop**.

Subiectul 35

Să se creeze o prezentare cu numele ”Calendar BAC 2022.ppsx” salvată în directorul **Atestat_mate_info 2022** care să conțină următoarea schemă:



Bacalaureat 2022
sesiunea 1,
Iunie-Iulie 2022,
probele scrise

- 20 iunie 2022- Limba și literatura română - proba E.a) - proba scrisă
- 21 iunie 2022-Proba obligatorie a profilului - proba E.c) -proba scrisă
- 22 iunie 2022- Proba la alegere a profilului și specializării - proba E.d) - proba scrisă
- 23 iunie 2022 - Limba și literatura maternă - proba E.b) - proba scrisă

Subiectul 36

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat_mate_info 2022**, de pe desktop.

	A	B	C	D	E	F
	Nr. Crt	Nume	Prenume	Clasa	Nr absențe nemotivate	Măsura adoptată
1						
2	1	Codreanu	Magda	11	23	
3	2	Ionescu	Corina	9	5	
4	3	Florescu	Mădălin	12	8	
5	4	Ștefănescu	Victor	10	2	
6	5	Dănilă	Camelia	9	120	
7		Filimon	Nicușor	12	63	

Pentru completarea coloanei **Clasa**, creați o listă din care să puteți alege valorile 9, 10, 11 și 12.

- ↖ În coloana **Măsura adoptată** introduceți o formulă de calcul care încarcă coloana cu mesaje astfel: - **”Mustrare scrisă”** dacă numărul absențelor nemotivate este mai mare sau egal cu 20 dar mai mic ca 40 și elevul este în clasa a 9-a sau a 10-a
- ↖ **”Avertisment de exmatriculare”** pentru elevii din clasele 11-12 dacă numărul absențelor nemotivate este mai mare sau egal cu 20 dar mai mic ca 40 și **”Exmatriculare”** cazul în care este mai mare sau egal cu 40.
- ↖ Dacă numărul absențelor nemotivate este mai mic de 20, atunci se va afișa mesajul ”OK”.



Subiectul 37

Creați pe spațiul de lucru (desktop) un folder cu numele **Atestat_mate_info 2022**.

- ↳ În acest folder creați trei subfoldere: *Imagini*, *Texte* și *Prezentări multimedia* care vor structura informații despre alcătuirea unui calculator.
- ↳ Salvați de pe Internet două imagini care să ilustreze un PC în subfolderul *Imagini*.

Subiectul 38

Creați o prezentare „forme de relief” pe care să o salvați în directorul **Atestat_mate_info 2022**, cu 4 diapozitive ce trebuie să conțină:

- ↳ Primul diapozitiv va fi de tip titlu și va conține titlul „*România – Forme de relief*”.
- ↳ Celelalte 4 diapozitive vor cuprinde imagini ale principalelor forme de relief din România, fiecare imagine va fi însoțită de un text corespunzător,
- ↳ Aplicați prezentării tranziții între diapozitive și animații la alegere

Subiect nr. 39

Creați pe spațiul de lucru (desktop) un folder cu numele **Atestat**.

- ↳ Localizați pe calculatorul la care susțineți proba aplicația **Wordpad** și creați o comandă rapidă (shortcut) către aceasta.
- ↳ Creați fișierul **atestat2022.txt** și modificați-i proprietățile astfel încât acesta să fie accesibil numai pentru citire (ReadOnly);

Subiectul 40

Realizați o prezentare Power Point cu 4 diapozitive ce trebuie să conțină:

- ↳ Primul diapozitiv va fi de tip titlu și va conține titlul „*Idei pentru o vacanță*”.
- ↳ Celelalte 3 diapozitive să conțină informații despre o destinație posibilă pentru o vacanță de iarnă: condiții de cazare, cel puțin două obiective turistice, două imagini sugestive, o legătură (link) la o pagină web a destinației turistice propuse la alegere,
- ↳ Aplicați prezentării tranziții între diapozitive și animații la alegere
- ↳ Salvați prezentarea cu numele *vacanta_iarna.ppt* în directorul **Atestat_mate_info 2022**



Baza de date FIRMA

Conține următoarele tabele:

Angajati (ida number(3), nume varchar2(30), profesie varchar2(30), data_nastere date, data_angajare date, salariu number(5), iddep number(3))

create table **Angajati** (ida number(3), nume varchar2(30), profesie varchar2(30), data_nastere date, data_angajare date, salariu number(5), iddep number(3))

Departamente (iddep number(3), denumire varchar2(30), etaj number(2), idsef number(3))

create table **Departamente** (iddep number(3), denumire varchar2(30), etaj number(2), idsef number(3))

Următoarele zece probleme fac referire la baza de date Firma

Subiectul nr. 1

Afișați numele, profesia și salariul angajaților descrescător după salariu.

```
SELECT nume, profesie, salariu FROM angajati ORDER BY salariu DESC
```

Afișați numele, profesia și departamentul în care lucrează fiecare angajat

```
SELECT nume, profesie, denumire FROM angajati ang, departamente dep  
WHERE ang.iddep=dep.iddep
```

Subiectul nr. 2

Afișați numele, profesia și data angajării pentru cel mai nou angajat.

```
SELECT nume, profesie, data_angajare FROM angajati  
WHERE data_angajare=(SELECT max(data_angajare) FROM angajati)
```

Afișați numele, profesia și departamentul pentru fiecare șef de departament

```
SELECT nume, profesie, denumire FROM angajati, departamente  
WHERE ida=idsef
```

Subiectul nr. 3

Afișați numele, profesia și data angajării pentru angajații cu o vechime de peste 10 ani.

```
SELECT nume, profesie, data_angajare FROM angajati  
WHERE ADD_MONTHS(data_angajare, 12*10) < SYSDATE
```

Afișați numele, profesia și salariul pentru angajații din departamentul IT

```
SELECT nume, profesie, salariu FROM angajati ang, departamente dep  
WHERE ang.iddep=dep.iddep AND UPPER(dep.denumire) = 'IT'
```



Subiectul nr. 4

Afișați numele, profesia și data nașterii pentru angajații care au sub 30 de ani.

```
SELECT nume, profesie, data_nastere FROM angajati  
WHERE ADD_MONTHS (data_nastere, 30*12) > SYSDATE
```

Afișați toți subordonații lui Avram Mihai. Se vor afișa numele și profesia

```
SELECT nume, profesie FROM angajati ang, departamente dep  
WHERE ang.iddep=dep.iddep AND idsef=(SELECT ida FROM angajati WHERE UPPER(nume)  
='AVRAM MIHAI') AND UPPER(nume)<>'AVRAM MIHAI'
```

Subiectul nr. 5

Să se majoreze cu 10% salariul celui mai vechi angajat.

```
UPDATE angajati SET salariu=1.1*salariu  
WHERE data_angajare=(SELECT min(data_angajare) FROM angajati)
```

Să se afișeze pentru toți angajații care lucrează la etajul 2 numele, profesia și departamentul.

```
SELECT nume, profesie, denumire FROM angajati ang, departamente dep  
WHERE ang.iddep=dep.iddep AND etaj=2
```

Subiectul nr. 6

Să se elimine din tabela **angajați** angajatul Vasilescu Andrei

```
DELETE FROM angajati WHERE UPPER(nume)='VASILESCU ANDREI'
```

Afișați denumirea departamentului condus de Avram Mihai

```
SELECT denumire FROM departamente  
WHERE idsef=( SELECT ida FROM angajati WHERE UPPER(nume)='AVRAM MIHAI')
```

Subiectul nr. 7

Afișați numele și data nașterii angajaților care își sărbătoresc azi data de naștere

```
SELECT nume, data_nastere FROM angajati  
WHERE EXTRACT(MONTH FROM SYSDATE) = EXTRACT(MONTH FROM data_nastere)  
AND EXTRACT(DAY FROM SYSDATE) = EXTRACT(DAY FROM data_nastere)
```

Afișați descrescător după salarii numele, profesiile și salariile angajaților de la departamentul economic

```
SELECT nume, profesie, salariu FROM angajati ang, departamente dep  
WHERE ang.iddep=dep.iddep AND UPPER(dep.denumire)= 'ECONOMIC'
```

Subiectul nr. 8

Afișați numele colegilor de departament pentru angajatul Costin Andrei

```
SELECT nume FROM angajati WHERE iddep=(SELECT iddep FROM angajati  
WHERE UPPER(nume)='COSTIN ANDREI') AND UPPER(nume)<>'COSTIN ANDREI'
```

Afișați numele celui mai tânăr șef de departament

```
SELECT nume FROM angajati  
WHERE data_nastere=(SELECT MAX(data_nastere) FROM angajati WHERE ida IN (SELECT  
idsef FROM departamente))
```



Subiectul nr. 9

Afișați numele șefilor de departamente

```
SELECT nume FROM angajati WHERE ida IN (SELECT idsef FROM departamente)
```

Afișați denumirea fiecărui departament și salariul maxim din acel departament

```
SELECT denumire, MAX(salariu) FROM angajati ang, departamente dep  
WHERE ang.iddep=dep.iddep GROUP BY denumire
```

Subiectul nr. 10

Să se majoreze cu 10% salariile angajaților din departamentul IT

```
UPDATE angajati SET salariu=1.1*salariu WHERE iddep=(SELECT iddep FROM departamente  
WHERE UPPER(denumire)= 'IT')
```

Afișați numele șefilor de departamente și denumirile departamentelor pe care le conduc

```
SELECT nume, denumire FROM angajati ang, departamente dep WHERE ang.iddep=dep.iddep  
AND ida=idsef
```

Baza de date MAGAZIN conține următoarele tabele:

Produse (idp number(5), denumire varchar2(30), categorie varchar2(30), data_fab date, stoc number(5), pret number(8,2))

```
create table Produse (idp number(5), denumire varchar2(30), categorie varchar2(30),  
data_fab date, stoc number(5), pret number(8,2) )
```

Vanzari (idv number(5), data_vanzare date, idp number(5), numar_bucati number(3))

```
create table Vanzari (idv number(5), data_vanzare date, idp number(5), numar_bucati  
number(3) )
```

Următoarele zece probleme fac referire la baza de date magazin

Subiectul nr. 11

Să se majoreze cu 10% produsele *lactate*

```
UPDATE pret SET pret=1.1*pret WHERE UPPER(categorie)= 'LACTATE'
```

Să se afișeze toate vânzările (denumire produs, data, număr de bucăți).

```
SELECT denumire, data_vanzare, numar_bucati FROM produse pd, vanzari vz WHERE  
pd.idp=vz.idp
```

Subiectul nr. 12

Să se afișeze denumirile produselor vândute azi

```
SELECT DISTINCT(denumire) FROM produse pd, vanzari vz WHERE pd.idp=vz.idp AND  
TO_CHAR(data_vanzare)=TO_CHAR(SYSDATE)
```

Să se afișeze categoria cu cele mai multe produse pe stoc (ca număr de bucăți)

```
SELECT categorie, SUM(stoc) FROM produse GROUP BY categorie HAVING SUM(stoc)=  
(SELECT MAX (SUM(stoc) ) FROM produse GROUP BY categorie)
```



Subiectul nr. 13

Să se afișeze denumirile produselor cu prețul cel mai mic

```
SELECT denumire, pret FROM produse  
WHERE pret=(SELECT min(pret) FROM produse)
```

Pentru fiecare produs vândut să se afișeze denumirea, totalul valorilor încasate precum și totalul de cantitate vândută.

```
SELECT denumire, SUM(pret*numar_bucati), SUM(numar_bucati) FROM produse pd, vanzari vz  
WHERE pd.idp=vz.idp GROUP BY denumire
```

Subiectul nr. 14

Să se afișeze categoriile produselor, prețul minim și prețul maxim pentru fiecare categorie

```
SELECT categorie, MIN(pret), MAX(pret) FROM produse GROUP BY categorie
```

Să se afișeze în ordine alfabetică denumirea produselor care nu s-au vândut.

```
SELECT denumire FROM produse  
WHERE idp NOT IN (SELECT DISTINCT(idp) FROM vanzari)
```

Subiectul nr. 15

Să se afișeze produsele cu stoc epuizat

```
SELECT denumire FROM produse WHERE stoc=0
```

Să se determine câte bucăți s-au vândut azi din produsul *Fanta*

```
SELECT SUM(numar_bucati) FROM produse pd, vanzari vz  
WHERE pd.idp=vz.idp AND UPPER (denumire) = 'FANTA'  
AND TO_CHAR(data_vanzare)=TO_CHAR(SYSDATE)
```

Subiectul nr. 16

Să se afișeze denumirea și prețul produselor cu prețul cuprins între 5 și 10 lei.

```
SELECT denumire, pret FROM produse WHERE pret >=5 AND pret <= 10
```

Să se afișeze toate încasările realizate la produsele lactate

```
SELECT denumire, SUM(numar_bucati*pret) FROM produse pd, vanzari vz  
WHERE pd.idp=vz.idp AND UPPER(categorie)='LACTATE' GROUP BY denumire
```

Subiectul nr. 17

Să se afișeze denumirile produselor lactate care expiră mâine știind că acestea expiră după 30 de zile de la data fabricației

```
SELECT denumire, data_fab FROM produse WHERE data_fab+30 <= SYSDATE+1
```

Să se afișeze produsele și numărul bucăților vândute descrescător după numărul bucăților vândute

```
SELECT denumire, SUM(numar_bucati) FROM produse pd, vanzari vz  
WHERE pd.idp=vz.idp GROUP BY denumire ORDER BY SUM(numar_bucati) DESC
```



Subiectul nr. 18

Să se șteargă produsele cu stoc 0

```
DELETE FROM produse WHERE stoc=0
```

Să se afișeze denumirile produselor vândute luna aceasta

```
SELECT DISTINCT(denumire) FROM produse pd, vanzari vz  
WHERE pd.idp=vz.idp AND TO_CHAR(data_vanzare,'mm')=TO_CHAR(SYSDATE,'mm')
```

Subiectul nr. 19

Să se afișeze categoria și valoare încasărilor pentru fiecare categorie de produse

```
SELECT categorie, SUM(numar_bucati*pret) FROM produse pd, vanzari vz WHERE pd.idp=vz.idp  
GROUP BY categorie
```

Să se micșoreze cu 10% prețul produselor cu preț maxim din categoria *dulciuri*

```
UPDATE produse SET pret=0.9*pret  
WHERE UPPER(categorie)= 'DULCIURI'
```

Subiectul nr. 20

Să se afișeze produsele ordonate pe categorii, denumirile lor și prețul acestora ordonate descrescător după preț în cadrul fiecărei categorii

```
SELECT categorie, denumire, pret FROM produse ORDER BY categorie, pret DESC
```

Să se afișeze denumirea/denumirile produselor pentru care s-au realizat cele mai multe vânzări (ca număr de bucăți cumulate în toate vânzările)

```
SELECT denumire FROM produse  
WHERE idp IN  
(SELECT idp FROM vanzari GROUP BY idp HAVING SUM(numar_bucati)=  
(SELECT MAX(SUM(numar_bucati)) FROM vanzari GROUP BY idp) )
```

Baza de date ȘCOALA conține următoarele tabele:

Elev (*ide* number(3), *nume* varchar2(30), *data_nastere* date, *medie* number(5,2), *cls* varchar2(5))

```
create table Elev (ide number(3), nume varchar2(30), data_nastere date, medie number(5,2),  
cls varchar2(5))
```

Clasa (*cls* varchar2(5), *specializare* varchar2(30), *sala* number(2), *id_dirig* number(3))

```
create table Clasa (cls varchar2(5), specializare varchar2(30), sala number(2), id_dirig  
number(3))
```

Profesor (*idp* number(3), *nume* varchar2(30), *disciplina* varchar2(30), *data_ang* date, *salariu* number(5))

```
create table Profesor (idp number(3), nume varchar2(30), disciplina varchar2(30), data_ang  
date, salariu number(5))
```

Următoarele zece probleme fac referire la baza de date Școala



Subiectul nr. 21

Afișați numele, disciplina predată și salariul profesorilor, descrescător după salariu.

```
SELECT nume, disciplina, salariu FROM profesor ORDER BY salariu DESC
```

Afișați numele și clasa elevilor de la profilul matematica-informatica

```
SELECT nume, e.cls FROM elev e, clasa c  
WHERE e.cls=c.cls AND specializare='matematica-informatica'
```

Subiectul nr. 22

Afișați disciplinele predate în școală, în ordine alfabetică.

```
SELECT DISTINCT disciplina FROM profesor ORDER BY disciplina
```

Afișați numele și clasa elevilor care au media 10, precum și diriginții acestora.

```
SELECT e.numa AS nume_elev, e.cls, p.numa AS nume_diriginte FROM elev e, clasa c,  
profesor p WHERE e.cls=c.cls AND idp=id_dirig AND medie=10
```

Subiectul nr. 23

Afișați salariul mediu al profesorilor de matematică.

```
SELECT AVG(salariu) FROM profesor WHERE disciplina='matematica'
```

Afișați elevii din clasele a 9-a și sălile lor de clasă.

```
SELECT e.numa, e.cls FROM elev e, clasa c WHERE e.cls=c.cls AND e.cls LIKE '9%'
```

Subiectul nr. 24

Câți profesori de matematică și informatică sunt în școală?

```
SELECT COUNT(*) FROM profesor WHERE disciplina='matematica' OR  
disciplina='informatica'
```

Afișați elevii care au ca dirigințe profesorul cu cea mai mică vechime.

```
SELECT e.numa FROM elev e, clasa c, profesor p  
WHERE e.cls=c.cls AND idp=id_dirig AND data_ang=(SELECT MAX(data_ang) FROM  
profesor)
```

Subiectul nr. 25

Afișați numele, salariul și data angajării pentru profesorii cu vechime mai mare de 10 ani.

```
SELECT nume, salariu, data_ang FROM profesor  
WHERE ADD_MONTHS(data_ang, 12*10) < SYSDATE
```

Afișați numele elevilor care au ca diriginți profesori de matematică.

```
SELECT e.numa FROM elev e, clasa c, profesor p  
WHERE e.cls=c.cls AND idp=id_dirig AND disciplina='matematica'
```

Subiectul nr. 26

Care sunt clasele care nu au specializarea FILOLOGIE?

```
SELECT cls, specializare FROM clasa  
WHERE LOWER(specializare)!='filologie'
```

Afișați mediile elevilor pe clase.

```
SELECT e.cls, AVG(medie) FROM elev e, clasa c, profesor p  
WHERE e.cls=c.cls AND idp=id_dirig GROUP BY e.cls
```

Subiectul nr. 27

Să se ștergă elevii cu media mai mică decât 5.

```
DELETE FROM elev WHERE medie<5
```

Să se afișeze diriginții ai căror elevi au medii între 8 și 9.

```
SELECT e.num_e AS elev, e.medie, p.num_p AS profesor FROM elev e, clasa c, profesor p  
WHERE e.cls=c.cls AND idp=id_dirig AND medie BETWEEN 8 AND 9
```

Subiectul nr. 28

Să se afișeze elevii care își serbează ziua de naștere în luna curentă.

```
SELECT nume FROM elev  
WHERE EXTRACT(MONTH FROM data_nastere)=EXTRACT(MONTH FROM SYSDATE)
```

Să se afișeze profesorii care au salariul mai mare decât salariul dirigintelui de la clasa 9B.

```
SELECT nume, salariu FROM profesor  
WHERE salariu>(SELECT salariu FROM profesor, clasa WHERE id_dirig=idp AND cls='9B')
```

Subiectul nr. 29

Câți elevi învață la etajul 1 (sălile de la 10 la 19)?

```
SELECT COUNT(*) FROM elev e, clasa c  
WHERE e.cls=c.cls AND sala BETWEEN 10 AND 19
```

Afișați salariul celui mai vechi angajat.

```
SELECT nume, salariu FROM profesor WHERE data_ang=(SELECT MIN(data_ang) FROM  
profesor)
```

Subiectul nr. 30

Să se afișeze numele elevilor și clasele din care fac parte, împreună cu numerele lor de ordine din tabela ELEV.

```
SELECT ROWNUM AS nr_crt, nume, cls FROM elev
```

Afișați elevii care au aceeași specializare cu Popa Daniela.

```
SELECT nume, e.cls FROM elev e, clasa c  
WHERE c.cls=e.cls AND UPPER(nume)<>'POPA DANIELA' AND specializare=  
(SELECT specializare FROM clasa c, elev e  
WHERE c.cls=e.cls AND UPPER(nume)='POPA DANIELA')
```



Baza de date CLINICA

Conține următoarele tabele:

Medic (**idm** number(3), **nume** varchar2(30), **data_ang** date, **salariu** number(8,2), **specializare** varchar2(30))

create table Medic (idm number(3), nume varchar2(30), data_ang date, salariu number(8,2), specializare varchar2(30))

Pacient (**idp** number(4), **nume** varchar2(30), **data_n** date, **adresa** varchar2(20), **statut** varchar2(20), **tel** varchar2(10))

create table Pacient (idp number(4), nume varchar2(30), data_n date, adresa varchar2(20), statut varchar2(20), tel varchar2(10))

Consultatie (**idp** number(4), **idm** number(3), **data_cons** date, **diagnostic** varchar2(30))

create table Consultatie (idp number(4), idm number(3), data_cons date, diagnostic varchar2(30))

Următoarele zece probleme fac referire la baza de date CLINICA

Subiectul nr. 31

Afișați numele și salariile tuturor medicilor, în ordine alfabetică

```
SELECT nume, salariu FROM medic ORDER BY nume
```

Să se afișeze pentru fiecare consultație atât data consultației, cât și datele pacienților (nume, adresa, statut) și ale medicilor (nume, specializare). Numele se vor afișa cu majuscule.

```
SELECT data_cons, specializare, UPPER(m.nume), diagnostic, UPPER(p.nume), adresa, statut  
FROM medic m, pacient p, consultatie c WHERE m.idm=c.idm AND p.idp=c.idp
```

Subiectul nr. 32

Să se afișeze pacienții din Galați care sunt elevi sau studenți

```
SELECT * FROM pacient WHERE (statut='elev' OR statut='student') AND adresa LIKE 'Galati'
```

Să se afișeze consultațiile pacientei Barbu Ionela (data consultației, numele medicilor, specializările, diagnosticul)

```
SELECT data_cons, specializare, UPPER(m.nume), diagnostic FROM medic m, pacient p,  
consultatie c WHERE m.idm=c.idm AND p.idp=c.idp AND p.nume='Barbu Ionela'
```

Subiectul nr. 33

Câți pacienți care nu sunt pensionari sunt în tabela PACIENT?

```
SELECT COUNT(*) FROM pacient WHERE statut NOT LIKE 'pensionar'
```

Să se afișeze toate consultațiile medicilor cu salariu mai mare decât salariul mediu al tuturor medicilor.

```
SELECT nume, specializare, data_cons, diagnostic FROM medic m, consultatie c WHERE  
c.idm=m.idm AND salariu>(SELECT AVG(salariu) FROM medic)
```



Subiectul nr. 34

Să se afișeze toate datele medicului / medicilor cu salariu maxim

```
SELECT * FROM medic  
WHERE salariu=(SELECT MAX(salariu) FROM medic)
```

Să se afișeze, pentru fiecare diagnostic, numărul de pacienți diagnosticați.

```
SELECT diagnostic, COUNT(*) AS nr_pacienti FROM consultatie GROUP BY diagnostic
```

Subiectul nr. 35

Câte consultații au fost cu fractură sau entorsă?

```
SELECT COUNT(*) FROM consultatie  
WHERE diagnostic='fractura' OR diagnostic='entorsa'
```

Care este suma salariilor medicilor care nu au acordat consultații în luna octombrie?

```
SELECT SUM(salariu) FROM medic WHERE idm NOT IN (SELECT idm FROM consultatie  
WHERE EXTRACT(MONTH FROM data_cons)=10)
```

Subiectul nr. 36

Să se afișeze medicii cu o vechime mai mare de 25 de ani.

```
SELECT nume, data_ang FROM medic  
WHERE ADD_MONTHS(data_ang, 25*12) <= SYSDATE
```

Care este salariul maxim al medicilor care au consultat elevi sau copii în luna curentă?

```
SELECT MAX(salariu) FROM medic m, consultatie c, pacient p  
WHERE m.idm=c.idm AND c.idp=p.idp AND (statut='copil' OR statut='elev') AND EXTRACT  
(MONTH FROM data_cons)=EXTRACT (MONTH FROM SYSDATE)
```

Subiectul nr. 37

Să se afișeze cel mai mic și cel mai mare salariu al medicilor

```
SELECT MAX(salariu), MIN(salariu) FROM medic
```

Câți pacienți diferiți au avut consultații în această lună? (se raportează la data curentă)

```
SELECT COUNT(DISTINCT p.idp) FROM pacient p, consultatie c  
WHERE p.idp = c.idp AND EXTRACT (MONTH FROM data_cons) = EXTRACT (MONTH  
FROM SYSDATE)
```

Subiectul nr. 38

Să se afișeze numărul de consultații din fiecare lună.

```
SELECT COUNT(*), TO_CHAR(data_cons,'MONTH') FROM consultatie GROUP BY  
TO_CHAR(data_cons,'MONTH')
```

Care sunt pacienții care au fost consultați de medicul cel mai recent angajat?

```
SELECT p.nume FROM medic m, consultatie c, pacient p  
WHERE m.idm=c.idm AND c.idp=p.idp AND data_ang=(SELECT MAX(data_ang) FROM  
medic)
```



Subiectul nr. 39

Să se afișeze datele pacienților care sunt pensionari

```
SELECT * FROM pacient WHERE statut='pensionar'
```

Să se afișeze alfabetic numele pacienților care au avut consultații și diagnosticele primite.

```
SELECT p.nume, diagnostic FROM medic m, consultatie c, pacient p  
WHERE m.idm=c.idm AND c.idp=p.idp ORDER BY diagnostic
```

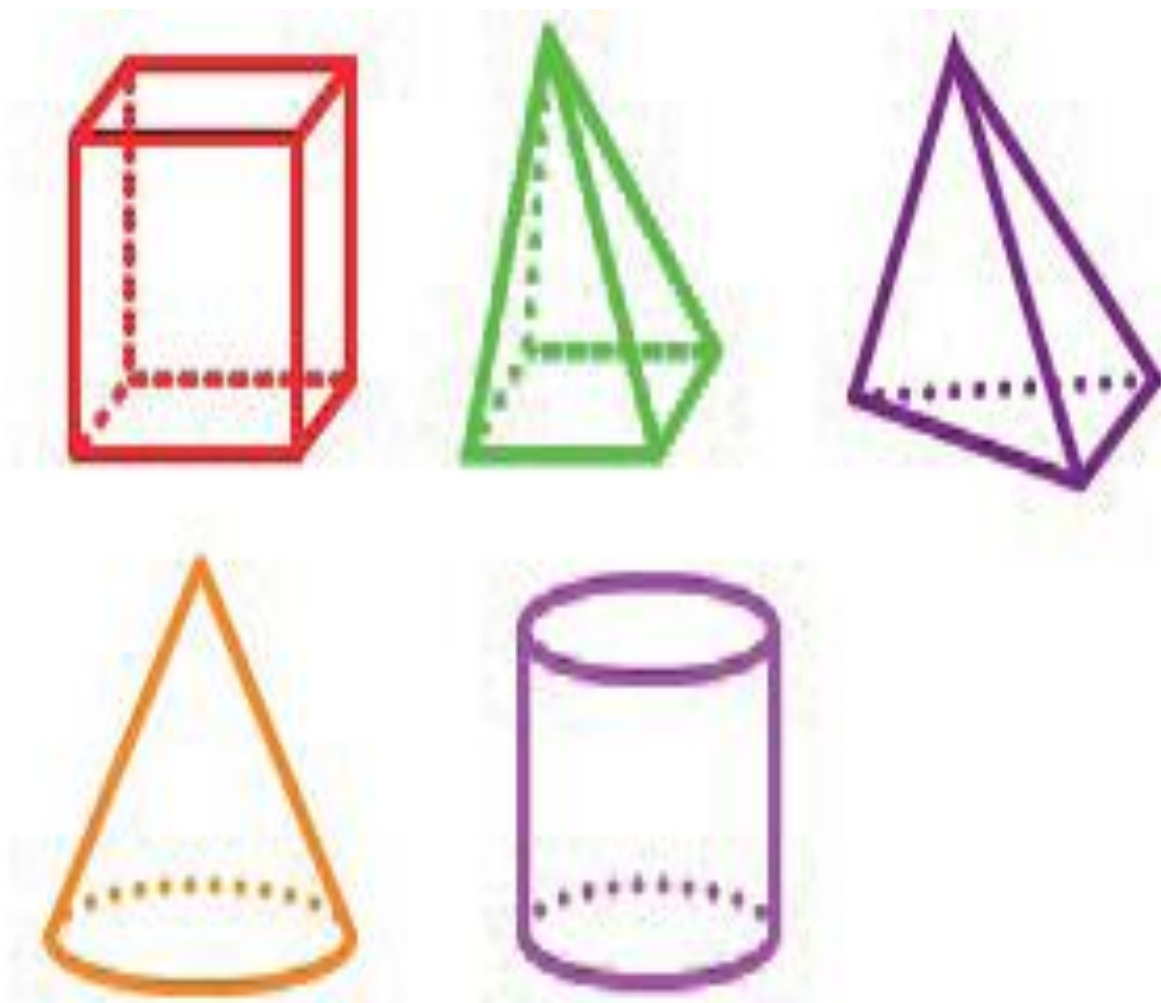
Subiectul nr. 40

Să se afișeze suma salariilor tuturor medicilor.

```
SELECT SUM(salariu) FROM medic
```

Să se afișeze toți pacienții care au avut aceleași diagnostice ca pacientul Costin Ion.

```
SELECT nume, diagnostic FROM consultatie c, pacient p  
WHERE c.idp=p.idp AND diagnostic IN  
(SELECT diagnostic FROM consultatie c, pacient p
```



PREPARE YOURSELF FOR THE FUTURE

Erasmus+, Orahovica 2022

„In a short amount of time I realised how funny and full of joy all of you are. Romania's team showed me that our countries aren't that different; we all love to have good time, we have similar dances, similar food etc. You became friends that I never want to lose”

studentă din Croația

piața muncii.

Erasmus+ este programul UE în domeniile educației, formării, tineretului și sportului pentru perioada 2021-2027. Educația, formarea, tineretul și sportul sunt domeniile cheie care sprijină cetățenii în dezvoltarea lor personală și profesională. Educația și formarea de înaltă calitate, favorabile incluziunii, precum și învățarea informală și nonformală conferă, în ultimă instanță, tinerilor și participanților de toate vârstele calificările și competențele necesare pentru participarea lor semnificativă la societatea democratică, pentru înțelegerea interculturală și pentru tranziția de succes pe

După cum ne-a demonstrat pandemia de COVID-19, accesul la educație se dovedește a fi, mai mult ca niciodată, esențial pentru asigurarea unei redresări rapide, promovând în același timp egalitatea de șanse pentru toți. Ca parte a acestui proces de redresare, programul Erasmus+ își duce dimensiunea favorabilă incluziunii către un nou orizont, sprijinind oportunități pentru dezvoltarea personală, socio-educativă și profesională a cetățenilor din Europa și din afara acesteia, astfel încât nimeni să nu fie lăsat în urmă.



Scopul final al acestui proiect este ca prin jocuri educaționale distractive, interacțiuni și muncă de echipă, să facă tinerii mai deschiși la minte și toleranți cu cei din jur, în timp ce îi încurajează să își îmbunătățească încrederea în sine și cel mai important să pregătească tinerii pentru viitor și să îi învețe cum să acționeze singuri în viață. Acest proiect, alături de experiența acumulată, ne-a informat despre diferitele sisteme educaționale din întreaga lume cât și despre contrastul dintre școlile private și cele publice, iar pe lângă toate acestea ne-a învățat să dezvoltăm cunoștințele și competențele care ne vor pregăti pentru găsirea unui job în viitor, ascultând poveștile motivaționale de succes ale oamenilor din întreaga lume.



*Articol scris de participant proiect Erasmus+,
Alexandra Maria Manole, clasa a XI-D,*

Coordonator prof. Burlacu Cătălina

Liceul Teoretic Dunărea din Galați



București, 17.04.2018

Către Asociația Profesorilor de Informatică „Tudor Sorin” Galați

Vă anunțăm ca publicația pe care o editați a fost înregistrată și a primit codul de identificare ISSN, după cum urmează:

Revista informatică (Im)puls++ (Online) =

ISSN 2601 - 5951, ISSN-L 2067 - 9815

Stimați colegi și elevi, vă invităm să colaborați la revista noastră. Responsabilitatea în legatură cu originalitatea și conținutul științific al articolelor, problemelor aparține în exclusivitate autorilor.

Felicitări elevilor și profesorilor care au înțeles că scopul acestei reviste este de a crea o relație elev-profesor fructuoasă. Așteptăm opiniile, problemele rezolvate și propuse pe adresa de mail a revistei noastre: revistainformaticagalati@yahoo.com

Colectivul de redacție:

- Redactor șef: **CĂTĂLINA BURLACU** – Liceul Teoretic „Dunărea”
- Redactor șef adjunct **GEORGETA-IULIA BALACEA, C. N. V. A.**
- Redactori principali profesorii:





(iM)PULS++ DE LA A LA Z:

Ambiție	Bunavoință	Curaj
Dorință	Energie	Forță
Geniu	Hotărâre	Intelligență
Joc	Knowledge	Luptă
Muncă	Nădejde	Optimism
Putere	Qualité	Reușită
Succes	Stiință	Tehnologie
Tel	Utilizare	Voință
Www	Xilografie	Youth
	Zel	