

CZU: 37.04:004.4

DOI: 10.36120/2587-3636.v30i4.46-57

INSTRUMENTE ONLINE DE DEZVOLTARE AVANSATĂ A COMPETENȚEI DE PROGRAMARE

Maria PAVEL, dr., conf. univ., UPSC

<https://orcid.org/0000-0003-4803-6398>

Dorin PAVEL, dr., conf. univ., UPSC

<https://orcid.org/0000-0002-9600-1360>

Rezumat. În articol se abordează problema formării și dezvoltării competenței de programare la studenți și liceeni pentru un învățământ de performanță și se descriu unele instrumente online care facilitează acest proces.

Cuvinte cheie: competențe de programare, limbaj de programare, informatică, instrument online.

ONLINE TOOLS FOR ADVANCED DEVELOPMENT OF THE PROGRAMMING SKILL

Summary. The article addresses the issue of training and developing programming competence in students and high school students for performance education and describes some online tools that facilitate this process.

Keywords: programming skills, programming language, computer science, online tool.

Introducere

Necesitatea orientării profesionale a tinerilor spre specialități informatice este dictată de cerințele pieței muncii, tehnologizarea și digitalizarea majorității domeniilor economiei mondiale și politicile educaționale moderne, axate pe competențe matematice, digitale, de rezolvare a problemelor în echipă, de gândire logică și creativă [1]. Un aport considerabil în acest sens îl are antrenarea liceenilor și studenților în învățământul de performanță la informatică, iar Republica Moldova, la acest capitol, are o politică bine pusă la punct de către Agenția Națională pentru Curriculum și Evaluare, prin organizarea de olimpiade și concursuri la nivel local, național și implicarea la nivel internațional. În acest context, politica agenției se realizează prin organizarea olimpiadelor la informatică, concursurilor de robotică, de programare etc. Aceste evenimente se bazează în mare parte pe competențele de programare ale elevilor și studenților, care trebuie să fie motivați, susținuți și coordonați în mare parte de către profesorii de informatică. Rolul cadrului didactic este enorm în acest sens, deoarece în procesul educațional la informatică, el are posibilitatea de a identifica și încuraja elevul care are aptitudini în domeniul programării și a-l antrena în activități orientate spre obținerea performanțelor.

Strategiile și metodologia explorată în procesul didactic pentru informatica competitivă sunt foarte diverse și depind în mare parte de stilul și experiența cadrului didactic. Sunt cadre didactice dedicate acestui proces, care contribuie enorm la formarea olimpicilor la informatică, ce reprezintă cu succes țara la nivel regional și internațional [2, 3, 4]. Didactica informaticii competitive presupune strategii și metode de familiarizare a

elevilor cu diferiți algoritmi și tehnici avansate de programare care nu întotdeauna se regăsesc în programa școlară la informatică. De asemenea, modul de expunere a legendei problemelor, restricțiile asupra datelor probleme, limita de timp de execuție, citirea și afișarea datelor din fișiere textuale, testarea codurilor de program cu ajutorul bateriilor de teste, sunt nuanțe pe care elevii trebuie să și le însușească. Dar cel mai important este motivația elevului de a persevera în domeniul programării, auto-eficiența și auto-reglarea procesului de învățare în ritm propriu.

Metode și mijloace

În scopul facilitării procesului de formare a competențelor de programare performante, în ajutorul profesorului pot veni un șir de instrumente online, care furnizează un set larg de probleme, foarte diverse ca tematică, ce permit realizarea codului în diferite limbaje de programare și care sunt verificate cu seturi impunătoare de teste. Rolul profesorului constă în identificarea acestor instrumente, analiza funcționalității acestora și selectarea și recomandarea instrumentului potrivit pentru fiecare elev. Instrumentele online sunt performante, conțin compilatoare de ultimă generație, au o interfață prietenoasă, ceea ce le fac atractive pentru elevi, în detrimentul cărților clasice pe care tinerii moderni nu le prea agreează.

Cele mai populare site-uri web ce conțin astfel de instrumente, își propun să contribuie la îmbunătățirea competențelor analitice, creșterea productivității printr-o manieră eficientă de rezolvare de probleme, pentru utilizatorii cu diferite niveluri de formare inițială: începători sau programatori profesioniști. Cei din urmă, și nu numai, își pot dezvolta competențele avansate de programare în domeniile de inteligență artificială, realitate augmentată, dezvoltare web, dezvoltare de aplicații etc.

La o simplă solicitare, motoarele de căutare ne vor redirecționa către astfel de instrumente ca: [beecrowd](#), [Kaggle](#), [Topcoder](#), [CodeChef](#), [Coderbyte](#), [AtCoder](#), [GeeksforGeeks](#), [Code Jam – Google’s Coding Competitions](#), [CodinGame](#), [HacherEarth](#) etc. [5]. Unele din aceste instrumente sunt gratuite, altele oferă servicii contra cost, unele organizează concursuri și turnee pentru programatori și posibilitate de creștere profesională, altele întrunesc utilizatori dedicați într-un anumit domeniu. De exemplu, cel mai potrivit și prietenos pentru programatorii începători, deci și pentru elevii la început de cale în informatica competitivă, este site-ul *beecrowd*, care oferă probleme din categoriile: *beginner*, *ad-hoc*, *string*, *structures*, *mathematics*, *paradigms*, *graphs* și *geometry*. Fiecare utilizator poate să filtreze problemele după nume, subiect, nivel, număr de identificare și alte criterii. Dacă utilizatorul decide participarea în diferite concursuri de programare, atunci platforma va permite verificarea clasamentului internațional, național, sau chiar din cadrul universității din care eventual face parte, în cazul în care acesta este student.

În cele ce urmează, se vor descrie unele astfel de instrumente, care au fost testate și s-au dovedit a fi eficiente și prietenoase elevilor în pregătirea pentru concursuri de programare.

Rezultate obținute

Una din cele mai populare platforme de dezvoltare și valorificare a competențelor de programare, chiar și pentru începători, este site-ul [HackerRank](#). Platforma este foarte prietenoasă, definitivată în detalii, ușor navigabilă, ceea ce permite utilizarea fără emoții pentru noii utilizatori. Crearea unui cont, pentru exersare și pregătire („I’m here to practice and prepare”), este foarte accesibilă, serviciile sunt absolut gratuite: peste 1000 de probleme de rezolvat, pregătire pentru interviuri, certificate recunoscute în domeniu, competiții de programare, peste 55 de abilități de învățare. După selectarea subiectului de antrenare (figura 1), va fi propusă prima sarcină, cu detalii de execuție, cerințe, restricții, input solicitat și output așteptat (figura 2).

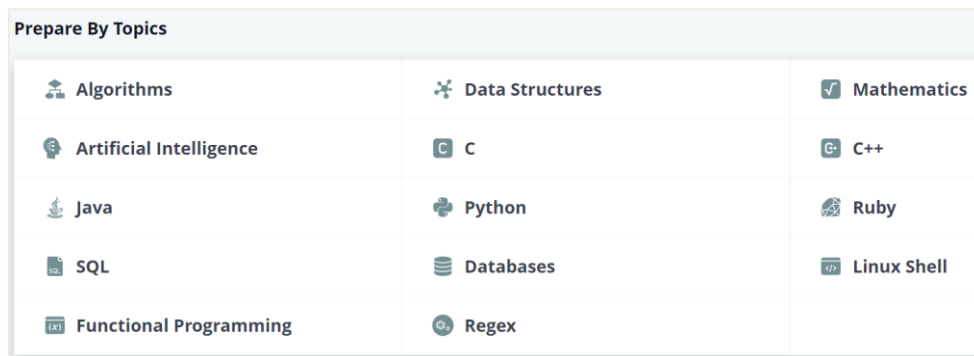


Figura 1. Subiectele propuse de HackerRank

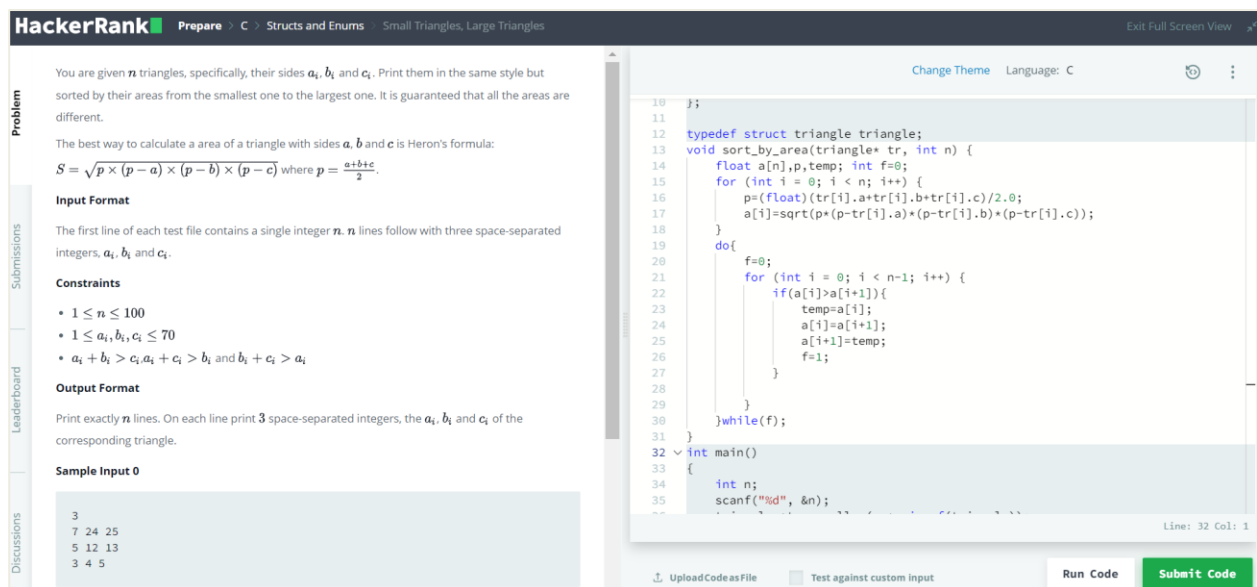


Figura 2. Problemă propusă și programul realizat în HackerRank

Utilizatorul compilează programul pentru testul 0 (*Sample Input 0*), acționând butonul *Run Code*, fără a introduce datele din *Input*, deoarece acestea sunt citite automat

din fișierul STDIN. Dacă testul este trecut cu succes, poate fi trimis codul pentru bateria de teste (*Submit Code*), după care se afișează starea testelor (figura 3) și în caz de succes se afișează punctajul acumulat și butonul de lansare a unei noi provocări (*Next Challenge*).

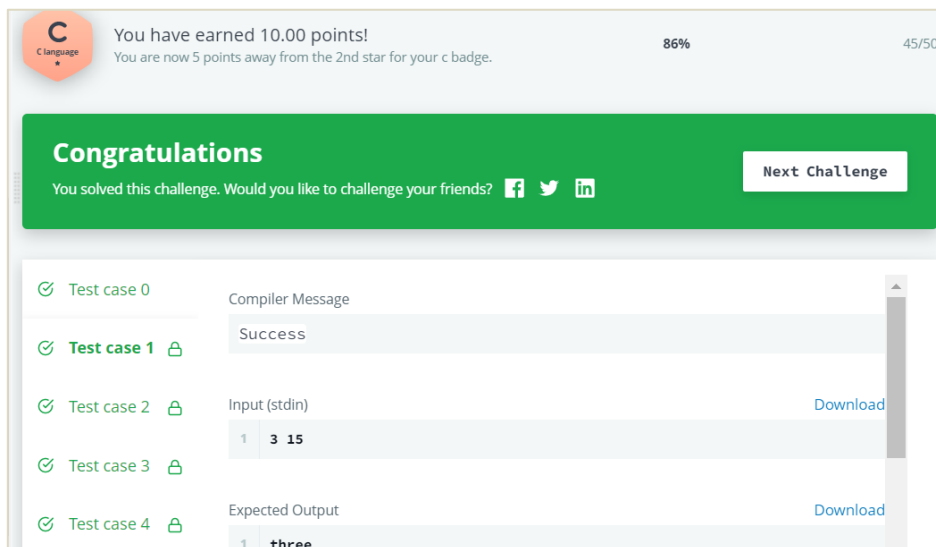


Figura 3. Bateria de teste pentru problemă propusă în HackerRank

Pentru a facilita soluționarea problemelor propuse, platforma propune codul de program completat cu partea de program elementară (preprocesări, șablonul funcției *main*, citire de date etc.). De asemenea, avem acces la statistica problemelor după diferite criterii: rezolvate, nerezolvate, nivel de dificultate și subdomenii.

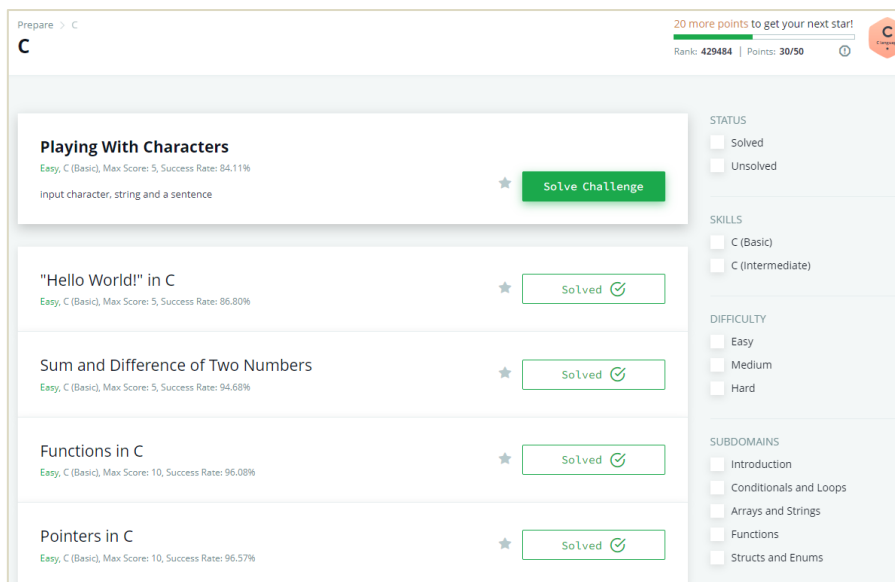
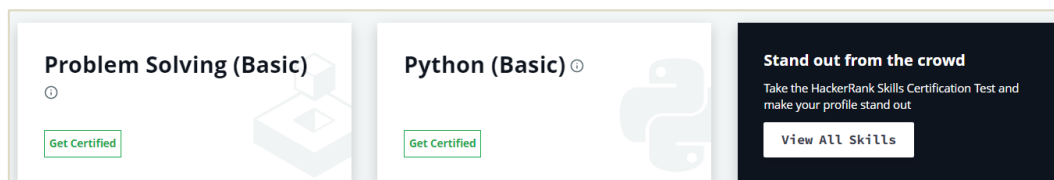


Figura 4. Statistica sarcinilor realizate în HackerRank

Platforma conține un șir de instrumentele ce permit dezvoltarea competențelor de programare prin rezolvarea de probleme propuse la nivelul *easy*, cu probleme accesibile utilizatorilor începători, continuând cu nivelul *medium* și permit avansarea la nivelul *hard*. Subiectele problemelor sunt foarte diverse, specifice informaticii de performanță, așa ca:

sortarea tablourilor de șiruri, inversarea tablourilor, structurarea documentelor, interogarea documentelor, tablouri dinamice, permutări de șiruri de caractere, funcții cu un număr variabil de argumente etc.

HackerRank oferă posibilitatea de a susține examenul de certificare pentru rezolvare de probleme, Python sau pentru crearea unui profil atractiv pentru diferite companii și dezvoltatori.



Un antrenament perfect pentru elevii ce sunt antrenați în informatica de performanță și pentru menținerea spiritului competitiv este accesarea paginii pentru concursuri (*Compete*). În această pagină, elevii dar mai ales studenții, pot accesa concursurile în derulare în scopul participării (de exemplu: *ProjectEuler+* sau *Cisco Hack Secure Challenge 2023*), sau cele arhivate (desigur dacă se încadrează în cerințele concursului respectiv). Un neajuns al acestui serviciu pentru elevii din Republica Moldova este că se solicită uneori donații pentru a participa la un concurs, care însă sunt colectate în scopuri caritabile.

Așa cum s-a menționat anterior, *HackerRank* este o platformă perfectă și pentru învățare, propunând în acest scop 3 tipuri de kit-uri cu probleme de bază și intermediare:

- pentru o săptămână, 21 sarcini, 6 teste;
- pentru o lună, 54 sarcini, 4 teste;
- pentru 3 luni, 104 sarcini, 13 teste.

De asemenea, atunci când se alege un anumit limbaj de programare, site-ul vine cu sugestii de studiere a limbajului respectiv, în preambulul fiecărei probleme. Însă, dacă utilizatorul cunoaște limbajul de programare, dar dorește doar să rezolve diferite probleme, atunci va accesa pagina *Algorithms*. Aceasta se utilizează exact ca și paginile anterioare, dar codul de program va conține diferiți algoritmi integrați în programul ce urmează a fi definitivat.

Un alt site web, foarte bine cunoscut, popular și cel mai utilizat din lume pentru programarea competitivă este [Codeforces](#), care se prezintă a fi mult mai complex, decât *HackerRank*, dar nu se va solicita mult timp de adaptare și familiarizare pentru utilizatorii noi înregistrați (înregistrarea este foarte ușoară, dar spre deosebire de *HackerRank*, va necesita confirmarea poștei electronice, printr-un click pe link-ul din mesajul de confirmare recepționat). Site-ul conține mai multe pagini, care se accesează în meniu: pagina principală cu noutăți (*Home*), un top al noutăților (*Top*), catalogul celor mai frecvente întrebări (*Catalog*, figura 5), pagina pentru concursuri (*Contests*), cursuri de antrenamente

(*Gym*), un set impresionant de probleme de diferite niveluri (A, B, C, D etc., figura 6), colectate din concursurile organizate (*Problemset*), grupuri de discuții (*Goups*), clasamentul utilizatorilor (*Rating*), pagina de instruire (Edu), pagina de acces la datele site-ului în format Json (API), calendarul concursurilor (*Calendar*) și o pagină de ajutor (*Help*).

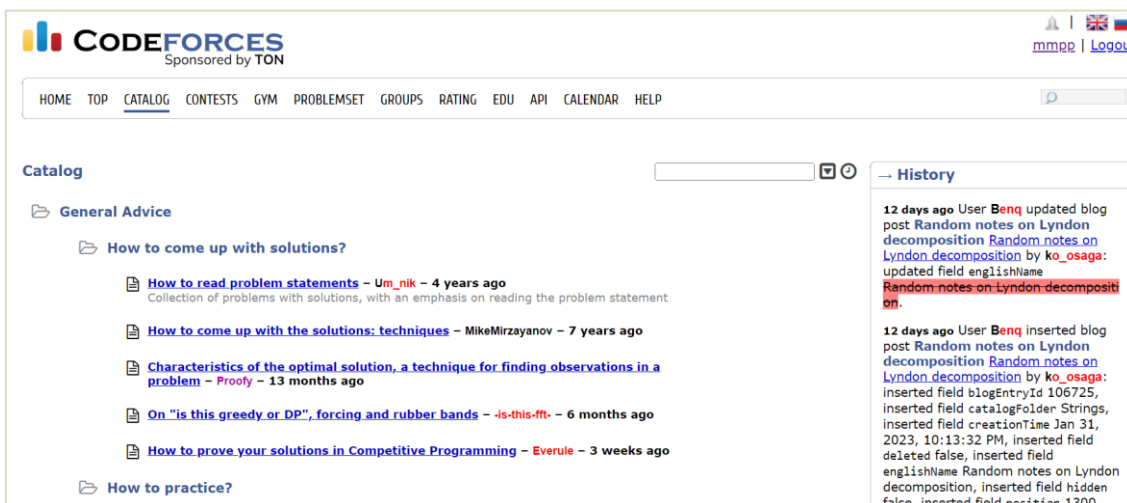


Figura 5. Pagina *Catalog* din Codeforces

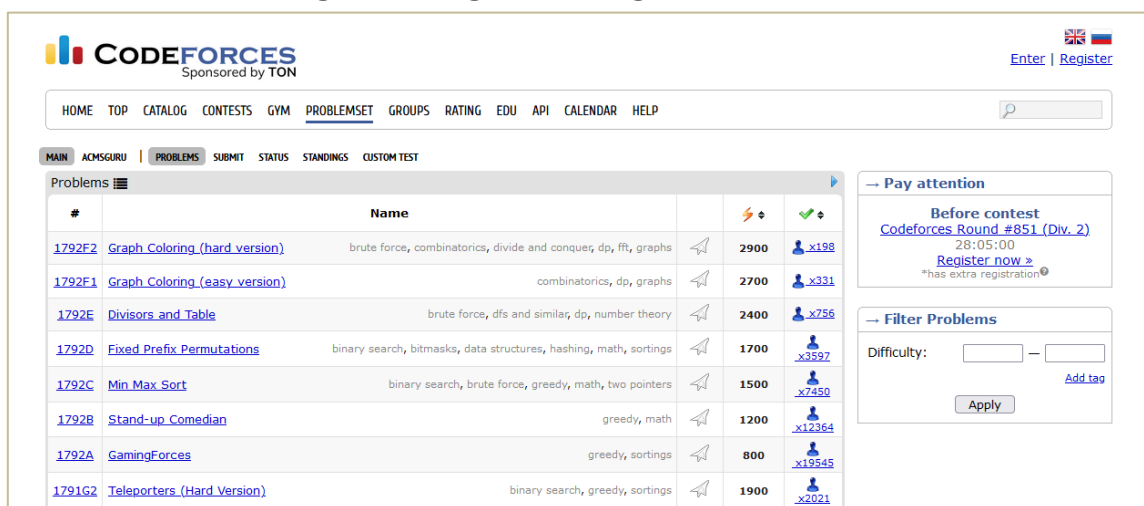


Figura 6. Pagina *Problemset* din Codeforces

După cum se vede în figura 6, problemele propuse se încadrează în categoria problemelor clasice pentru programarea competitivă, deoarece tratează așa subiecte ca: ecuații exponențiale, număr de factorizări, grafuri, colorarea grafurilor, permutări, transformări inverse, diferite jocuri cu permutări, puzzle, întrebări etc., interogări, toate tipurile de sortări: maxim, minim, interclasare, quick sort ș.a., arbori, căutarea binară, parole, numere Fibonacci etc. Pentru fiecare problemă, platforma sugerează algoritmi și tehnicile de programare care ar putea fi utilizate: forță brută, mascarea biților, algoritmi constructivi, cătarea binară, căutare ternară, metoda Greedy, programarea dinamică; dar și tipul de date și aria lor de aplicabilitate: șiruri de caractere, matrice, structuri de date, grafuri, arbori, pointeri; cu aplicare în matematică, geometrie, teoria numerelor, combinatorică, probabilități, sortări, teoria jocurilor etc.

După selectarea problemei, utilizatorul va introduce programul direct în fereastra *source code* de pe site (meniul *Submit*, din pagina problemei), alegând în prealabil limbajul de programare dorit, sau poate să încarce fișierul sursă cu programul realizat pe un compilator local.

B. Taxi

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After the lessons n groups of schoolchildren went outside and decided to visit Polycarpus to celebrate his birthday. We know that the i -th group consists of s_i friends ($1 \leq s_i \leq 4$), and they want to go to Polycarpus together. They decided to get there by taxi. Each car can carry at most four passengers. What minimum number of cars will the children need if all members of each group should ride in the same taxi (but one taxi can take more than one group)?

Input
The first line contains integer n ($1 \leq n \leq 10^5$) — the number of groups of schoolchildren. The second line contains a sequence of integers s_1, s_2, \dots, s_n ($1 \leq s_i \leq 4$). The integers are separated by a space, s_i is the number of children in the i -th group.

Output
Print the single number — the minimum number of taxis necessary to drive all children to Polycarpus.

Examples

input	Copy
5	
1 2 4 3 3	
output	Copy
4	

input	Copy
8	
2 3 4 4 2 1 3 1	
output	Copy
5	

Figura 7. Problemă propusă în Codeforces

Tastarea butonului de executare a codului *Submit*, va duce la verificarea tuturor testelor problemei respective și va returna verdictul de acceptare în caz de reușită, sau numărul testului la care s-a obținut un răspuns greșit.

De cele mai multe ori, greșelile pot apărea la testarea valorilor limită din constrângeri, caz în care se vor revedea tipurile de date. Pentru a identifica testul cu răspunsul greșit se va selecta numărul din prima coloană (#) cu link către codul sursă și se va consulta câmpul *checker log*, în care se indică valoarea așteptată și cea obținută. Platforma va afișa odată cu testele verificate și caracteristicile execuției: timpul de execuție, memoria utilizată, codul de verificare și verdictul fiecărui test. Redactările de cod nu pot fi făcute în această fereastră cu protocolul de hotărâre, care trebuie mai întâi închisă, ci se va accesa din nou meniul *submit*, se va alege titlul problemei, după care se va insera programul corespunzător și se va trimite la execuție. În câmpul *Judgement Protocol*, se vor rula toate testele până la final pentru a vizualiza starea lor. De exemplu pentru problema inițială #1A-Theatre Square, foarte simplă de altfel, bateria conține 9 teste.

Neajunsul platformei este redactarea codului de program în caz de ratare a unor teste, în care este necesar să se mai selecteze o dată titlul problemei și adăugarea repetată a codului de program.

By mmpp, contest: Codeforces Beta Round #1, problem: (A) Theatre Square, **Accepted**, #, [Copy](#)

```
#include <stdio.h>
int main()
{
    long long int m, n,a,nr1,nr2,nr;
    scanf("%lld %lld %lld", &m, &n, &a);
    nr1=m/a;
    if(nr1*a<m)nr1++;
    nr2=n/a;
    if(nr2*a<n)nr2++;
    nr=nr1*nr2;
    printf("%lld\n",nr);
return 0;
}
```

→ **Judgement Protocol**

Test: #1, time: 15 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input	6 6 4
Output	4
Answer	4
Checker Log	ok 1 number(s): "4"

Figura 8. Protocolul de hotărâre pentru problemă propusă în Codeforces

Complexitatea problemelor este ascendentă, ceea ce permite antrenarea elevilor și studenților începători în programarea competitivă, iar susținerea periodică a efortului de antrenare, consultarea blogurilor de discuții, trecerea treptată și progresivă la un nivel de complexitate mai avansat va contribui la dezvoltarea competenței de programare de performanță.

Figura 9. Pagina de start din LeetCode

Majoritatea elevilor și studenților ce fac informatică de performanță, aleg să continue o carieră profesională în acest domeniu la marile companii, care recrutează viitorii angajați

anume printre cei cu portofolii impresionante la capitolul programare competitivă. Aceste companii, ca *Apple*, *Google*, *Amazon*, *Facebook*, sau *Netflix* utilizează platforme de coding pentru a-și completa stafful de angajați. O astfel de platformă este [LeetCode](#), care contribuie la îmbunătățirea competențelor de programare, îmbogățirea cunoștințelor și pregătirea pentru interviuri tehnice (figura 9). Și pe acest site există meniul *Problems*, care permite selectarea tipului de problemă, domeniu, complexitate etc. De asemenea, platforma conține instrumente pentru instruire pe diferite subiecte și perioade de timp (figura 10).

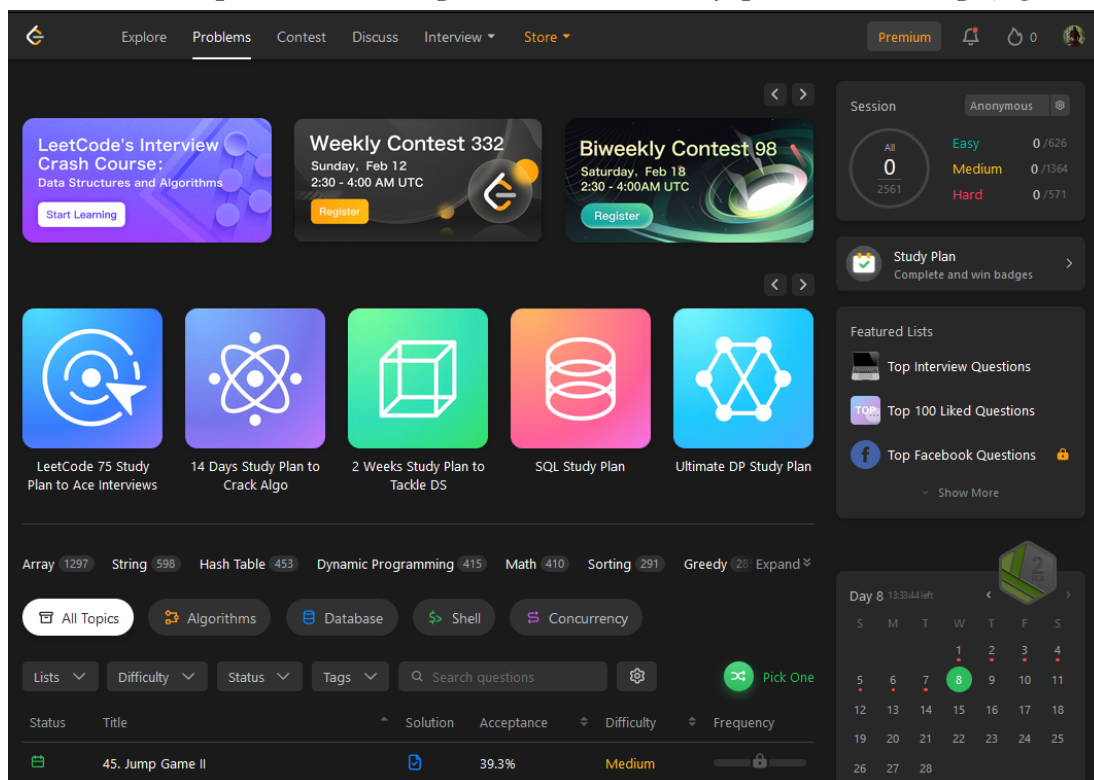


Figura 10. Pagina *Problems* din LeetCode

Setul de probleme este clasificat fie pe subiecte (tablouri, șiruri de caractere, stive, cozi, liste, grafuri, arbori binari, tabele de hashare, funcții de hashare, sortare, Greedy, Backtracking, recursie, mulțimi ordonate, teoria jocurilor, circuit eulerian etc.), fie pe companii pentru care utilizatorul aspiră să dea interviu. În coloana *Solution*, pentru fiecare problemă este atașat un fișier cu legenda problemei și indicații cu privire la algoritmul de rezolvare și programul în C++, Java sau Python.

Platformele descrise, și nu doar, contribuie la obținerea performanțelor în domeniul programării competitive, în contextul unei altfel de abordare STEAM, deoarece:

- permit asigurarea unei *stări de bine* (*S*) pentru utilizator, care se află în mediul familiar de programare, utilizând dispozitivul și compilatorul cu care este obișnuit;
- blogurile la care au acces utilizatorii contribuie la crearea oportunităților de lucru în echipă, fără restricții de *timp* (*T*), datorită particularităților de funcționare a blogurilor;
- oferă *șanse egale* (*E*) pentru orice utilizator, indiferent de țară, sex, vârstă etc.;

- fiecare utilizator programează în stil propriu, favorizând creativitatea și *abordarea inovativă (A)* a problemelor;
- permit utilizatorului să fie stăpân pe propriul traseu de dezvoltare și să asigure un *management eficient (M)* al acestui proces [6].

În contextul abordării inovative a problemelor, site-urile de programare competitivă oferă toate instrumentele necesare, deoarece majoritatea utilizatorilor sunt elevi sau studenți, tineri ai secolului XXI care apreciază viteza, căile rapide de soluționare a unor situații. Prin urmare, utilizarea limbajelor de programe care conțin biblioteci cu facilități de realizare a unor algoritmi, funcții, metode etc., sunt agreate și utilizate cu succes de competitorii înregistrați pe site-urile descrise. Acest fapt, însă, este mai puțin familiar profesorilor de informatică din Republica Moldova, care fiind formați profesional în context mai conservator, se adaptează un pic mai greu ritmului elevilor săi.

De exemplu, la faza de sector a olimpiadei de informatică din 2023 pentru clasa a XI, se propune spre rezolvare următoarea problemă (reformulată):

Toamna aurie. Andreea se plimbă prin pădure toamna și își propune să culeagă un buchet din frunzele căzute, astfel încât să conțină doar câte o frunză de o anumită culoare și de la o anumită specie de copac. Cunoscând culoarea și specia frunzelor pe care le găsește Andreea, calculați pe câte din ele le va alege pentru buchetul său?

Date de intrare: în fișierul *toamna.txt* pe prima linie se conține un număr întreg n ce reprezintă cantitatea de frunze găsite de Andreea ($1 \leq n \leq 100$). Pe următoarele n linii se descriu frunzele găsite: numele speciei copacului și culoarea frunzei, separate printr-un spațiu, fiecare din care conține nu mai mult de 10 caractere.

Date de ieșire: un singur număr, ce reprezintă numărul de frunze adunate de Andreea.

Un exemplu de program pe care l-a elaborat un elev, pregătit pe site-urile de programare competitivă este dat în tabelul 1 de mai jos.

Tabelul 1. Codul de program pentru rezolvarea problemei Toamna Aurie

```

1. #include <bits/stdc++.h>
2. using namespace std;
3. int main(){
4. ifstream file("toamna.txt");
5. vector<string> unique;
6. string temp;
7. int n, k = 0; file >> n;
8. for (int j = 0; j < n; ++j){
9. getline(file, temp);
10.     k = 0;
11.     for (int i = 0; i < unique.size(); ++i){
12.         if (temp == unique[i]) k++;
13.     }
14.     if (k==0) {
15.         unique.push_back(temp);

```

```

16.     }
17.     }
18.     cout << unique.size();
19.     return 0;
20.     }

```

Unele exemple pentru testare sunt:

Date de intrare: 5 stejar verde mesteacan rosu artar galben mesteacan galben artar galben	Date de intrare: 4 artar oranj artar oranj artar oranj artar oranj
Ieșire: 4	Ieșire: 1

Se observă faptul că întâi s-a inclus doar un singur fișier la preprocesare, care include toate fișierele antet necesare pentru apelul funcțiilor programului `bits/stdc++.h`, apoi s-a utilizat structura de date `vector`, pentru care s-au accesat metodele `push_back` și `size`. Acest fapt a condus la economisirea timpului de execuție și a memoriei alocate, dar și la reducerea numărului de linii de cod. În caz contrar, ar fi trebuit de inclus fișierul standard `iostream`, și fișierul de gestionare a șirurilor de caractere `string.h`. De asemenea, s-ar fi construit de exemplu, un tablou bidimensional de șiruri de caractere, prima coloană pentru specia copacului și a doua pentru culoarea frunzei; s-ar fi utilizat funcția de comparare a șirurilor (`strcmp(sir1, sir2)`) atât pentru șirul de caractere din prima coloană, cât și pentru cel din a doua coloană și nemijlocit s-ar fi utilizat operația logică `&&` (and). Dacă funcția de comparare ar fi returnat un rezultat diferit de 0 în ambele cazuri, atunci s-ar fi contorizat linia respectivă a tabloului.

Cele expuse mai sus nu influențează rezultatul parcurgerii bateriei de teste, deoarece acesta este un prim criteriu de acordare a punctajului pentru fiecare problemă, algoritmul utilizat nu este verificat inițial. Se recurge la analiza algoritmului, doar în cazul contestațiilor, prin urmare fiecare concurent poate utiliza stilul propriu de programare.

Concluzii

Utilizarea site-urilor de programare competitivă reprezintă o metodă eficientă de pregătire a elevilor și studenților pentru obținerea de performanțe și un ajutor esențial pentru profesorii ce doresc să implice discipolii săi în aceste activități. Profesorul este astfel scutit de necesitatea de a elabora sau selecta probleme, site-urile punând la dispoziție o colecție consistentă de probleme. De asemenea, este facilitat procesul de testare a programelor, bateria de teste fiind pregătită pentru fiecare problemă. Elevul nu este

constrâns în timp de pregătire, care altfel ar trebui acordat cu timpul pe care îl are la dispoziție profesorul. Elevul concurent cunoaște în timp real locul în clasamentul utilizatorilor site-ului respectiv, racordat la nivelului propriu de dezvoltare a competenței de programare. Totodată, acesta poate apela la ajutorul altor utilizatori prin intermediul blogurilor și forumurilor de discuții.

Un alt aspect, esențial al acestor site-uri este că sunt disponibile în marea majoritate doar în limba engleză, iar acest lucru contribuie la dezvoltarea abilităților de comunicare, percepere a mesajului problemei în limba engleză, ceea ce contribuie la o pregătire avansată pentru concursurile internaționale.

Articol realizat în cadrul proiectului de cercetări științifice „Metodologia implementării TIC în procesul de studiere a științelor reale în sistemul de educație din Republica Moldova din perspectiva inter/transdisciplinarității (concept STEAM)”, inclus în „Program de stat” (2020-2023), Prioritatea IV: Provocări societale, cifrul 20.80009.0807.20, cu suportul financiar oferit de Agenția Națională pentru Dezvoltare și Cercetare

Bibliografie

1. PAVEL, Maria; PAVEL, Dorin. Pre-achiziții din domeniul programării ale viitorilor informaticieni. În: *Acta et commentationes (Științe ale Educației)*. 2022, nr. 1(27), pp. 55-62. ISSN 1857-0623.10.36120/2587-3636.v27i1.55-62.
2. <https://moldovainprogres.com/sergiu-corlat-profesor-de-informatica-succesul-depinde-de-o-multitudine-de-factori-si-circumstante/>
3. CORLAT, Sergiu; IVANOV, Lilia. Problems of junior contestants training for participation in national and international programming competitions. În: *Acta Et Commentationes (Sciences of Education)*. 2019, nr. 18(4), pp. 120-124. ISSN 1857-0623. <https://doi.org/10.36120/2587-3636.v18i4.120-124>
4. GLOBALA, Angela; CORLAT, Sergiu. Didactic aspects regarding to creating test sets for competition problems. În: *Acta Et Commentationes (Sciences of Education)*. 2019, nr. 18(4), pp. 73-85. ISSN 1857-0623. <https://doi.org/10.36120/2587-3636.v18i4.73-85>
5. <https://www.freecodecamp.org/news/the-most-popular-coding-challenge-websites/>
6. GLOBALA, Angela; PAVEL, Maria; PAVEL, Dorin. *Repere didactice în rezolvarea problemelor de combinatorică. Aplicații C++ și MAPLE*. Ghid metodic. Chișinău: UST, 2022.185 p. ISBN 978-9975-76-403-2.