

# Tema 2 - Arhitectura Sistemelor de Calcul

Seriile 13, 14, 15

Decembrie 2021

## Cuprins

<b>1</b>	<b>Detalii administrative</b>	<b>2</b>
1.1	Deadline . . . . .	2
1.2	Reamintirea punctajului pe tema . . . . .	2
1.3	Transmitere . . . . .	2
1.4	Ce se va transmite . . . . .	2
1.5	Cum se va face evaluarea . . . . .	2
1.6	Alte observatii . . . . .	2
<b>2</b>	<b>Formularea temei</b>	<b>3</b>
<b>3</b>	<b>Tema bonus</b>	<b>4</b>

# 1 Detalii administrative

## 1.1 Deadline

Puteți trimite soluțiile cel târziu pe 3 Ianuarie 2022, ora 23:55.

## 1.2 Reamintirea punctajului pe tema

Tema valorează 33% din nota la acest laborator (conform Cursului 0x00), și **nu** este necesară obținerea notei 5 pentru promovare, **dar** este necesar ca media celor două note obținute pe teme să fie minim 5.

## 1.3 Transmitere

Studentii vor trimite soluțiile în următorul formular, astfel:

- seriile 13 și 15: <https://forms.gle/tVJu2eqA9B763URN8>
- seria 14: <https://forms.gle/BBD1Vws8p3aceyoL9>

## 1.4 Ce se va transmite

Se va trimite **o singură sursă** cu denumirea **grupa\_nume\_prenume.asm**. Dacă aveți mai multe nume / prenume, veți trimite o sursă de forma **172\_GeorgescuXulescu\_IonVasile.asm**. Este **important** să trimiteți o sursă cu denumirea corectă, deoarece testarea va fi **automată**. În mail se va completa la subiect **Grupa, Nume Prenume, Tema 2 ASC**, de exemplu **172, Georgescu Ion Vasile, Tema 2 ASC**.

## 1.5 Cum se va face evaluarea

Există doi pași pentru obținerea notei:

- se verifică toate sursele să nu fie cazuri de plagiat. În cazul în care se detectează plagiat, se face automat sesizare către *Comisia de Etică a Universității din București*;
- sursele care au trecut de verificarea anti-plagiat, vor fi testate automat.

**Important!** Studentii care au alte configurații decât cele pe care lucram la laborator, trebuie să precizeze acest lucru în mailul prin care transmit tema, pentru a putea efectua evaluarea și pentru a nu primi 0 implicit.

## 1.6 Alte observații

1. Nu vă interzicem să discutați idei între voi, dar aveți grijă, deoarece este o diferență importantă între a da o idee și a da codul direct.
2. Nu folosiți convertoare automate din C/C++/alte limbaje în x86, le-am folosit și noi și recunoaștem fără dificultate un cod care nu este scris de voi.

**Scriptul de evaluare automată este același script utilizat și pentru prima temă.**

## 2 Formularea temei

Se citesc de la tastatura  $n$ ,  $m$  si  $3 \cdot n$  elemente care pot fi sau 0, sau cuprinse între 1 si  $n$ , unde se respecta conditia  $1 \leq n, m \leq 30$ . Se va genera **cea mai mica permutare in sens lexicografic** a multimii  $\{1, \dots, n\}$ , unde fiecare element apare **de exact 3 ori**, avand o distanta de **minimum**  $m$  elemente între oricare doua elemente egale, plecand de la anumite puncte fixe deja specificate.

De exemplu, pentru  $n = 5$ ,  $m = 1$  si secventa de 15 elemente

1 0 0 0 0 0 3 0 0 0 0 0 0 4 5

avem ca fiecare element din multimea  $\{1, 2, 3, 4, 5\}$  apare de 3 ori, si vrem sa fie cel putin  $m = 1$  element distanta între oricare doua elemente egale. Atunci, cea mai mica permutare in sens lexicografic, pastrand punctele fixe, este urmatoarea:

1 2 1 2 1 2 3 4 3 5 3 4 5 4 5

Se vor afisa la *standard output*, dupa caz,

- sau permutarea, daca exista, in formatul de mai sus: elementele se vor afisa cu spatii între ele pe ecran, iar la final recomandam sa afisati un caracter *backslash n*, in locul utilizarii *fflush*;
- sau  $-1$ , in cazul in care nu exista o permutare care sa satisfaca toate conditiile.

**IMPORTANT!** Pentru ca rezolvarile sa fie evaluate cu punctaj maxim, trebuie sa respecte urmatoarele cerinte:

1. sa fie utilizate procedurile, respectand **toate** conventiile prezentate in laborator, referitoare la constructia corecta a cadrului de apel;
2. algoritmul pe care il veti implementa va fi un algoritm **backtracking recursiv**.

In cazul in care implementati fara proceduri, veti primi doar 20% din punctajul pe care l-ati obtinut in urma evaluarii automate, iar in cazul in care implementati utilizand **backtracking iterativ** sau alti algoritmi, veti primi doar 70% din punctajul obtinut in urma evaluarii automate. Timpul maxim de rulare permis este de 2s.

### 3 Tema bonus

Pentru un bonus de 50p la aceasta tema, care se va considera bonus adaugat punctajului la laborator daca luati 100p din prima cerinta, puteti rezolva urmatoarea problema.

Se citeste dintr-un fisier text un puzzle Sudoku  $9 \times 9$ , avand elemente in multimea  $\{0, 1, \dots, 9\}$ , unde 0 indica faptul ca respectiva casuta nu are valoare completata. Sa se scrie un algoritm care rezolva acest puzzle, gasind prima solutie in ordine lexicografica (acolo unde solutia nu este unica). Puzzle-ul rezolvat va fi scris intr-un fisier de iesire.

Aceasta solutie va fi transmisa tot pana pe **3 ianuarie 2022, ora 23:55**, si va fi atasata intr-un alt fisier asm, cu denumirea **grupa\_Nume\_Prenume\_bonus.asm**. Este permis, pentru rezolvarea acestei probleme, sa utilizati si *inline assembly*, iar atunci fisierul transmis va avea denumirea **grupa\_Nume\_Prenume\_bonus.c**. Atasati in mail si fisierul de intrare din care cititi.

**Important!** Aceasta cerinta nu va fi evaluata automat, dar va rugam sa trimiteti doar daca stiti ca solutia oferita este o solutie corecta. (eventual ne puteti trimite mai multe fisiere de intrare pe care ati testat)