

Indicații pentru Tema 3

- **Tema 3** va avea **6 probleme**, fiecare punctată cu note de la 1 la 10 astfel:
 - Indentare corespunzătoare = **2 puncte**
 - Utilizarea acoladelor, lizibilitatea codului = **2 puncte**
 - Utilizarea corespunzătoare a instrucțiunilor de tip **if, for, while, do...while** = **1 punct**
 - Declararea corespunzătoare de variabile = **1 punct**
 - Funcționalitatea corespunzătoare a programului (adică programul face ce trebuie să facă) = **2 puncte**
 - Citirea și afișarea corespunzătoare a variabilelor/ meniurilor = **1 punct**
- Tema trebuie încărcată pe Campus Virtual până cel târziu **MIERCURI, 6 noiembrie 2019, ora 23:59 (hard deadline)**.
- **TOATE PROBLEMELE SUNT OBLIGATORII!!**
- Vor fi încărcate 6 fișiere cu extensia **.c** denumite astfel: **p1.c, p2.c, p3.c, p4.c, p5.c** și **p6.c** corespunzătoare rezolvărilor problemelor 1, 2, 3, 4, 5, respectiv 6.
- **NU încărcăți proiecte CodeBlocks!!!** Numai fișierele cu extensia **.c** denumite corespunzător după cele menționate la subpunctul anterior.
- **IMPORTANT!!** Dacă nu compilează codul, problema este notată cu 0. Punctul din oficiu pe fiecare problema se acordă în cazul în care codul compilează și nu are nicio eroare de compilare!
- Tema este **OBLIGATORIE** și cei care nu o fac nu vor putea da Testul 2!!
- Copiatul este strict interzis! Dacă sunt găsite astfel de cazuri, toți studenții implicați (adică cei care copiază și cei de la care se copiază) vor fi exmatriculați.
- Încercați să rezolvați voi singuri problemele. Dacă aveți nevoie de ajutor, vă stau la dispoziție.
- Pentru orice fel de întrebări nu ezitați să mă contactați.

Tema 3

Problema 1

Se dă un șir format din n ($n \leq 30$) valori întregi. Să se citească și să se memoreze elementele șirului într-un vector. **După terminarea citirii și memorării tuturor elementelor șirului în vector**, să se insereze după fiecare element egal cu 0 un element cu valoarea 1 și să se afișeze vectorul după efectuarea acestor modificări. *Exemplu:* se citește șirul 2,0,1,0,3 și se afișează șirul 2,0,1,1,0,1,3.

Problema 2

Să se citească un tablou unidimensional de $n \leq 25$ numere întregi fără semn. **După terminarea citirii și memorării tuturor elementelor** tabloului afișați elementele, câte unul pe un rând în 4 forme: în baza 10, în baza 8, în baza 16 (cifrele vor avea litere MARI, nu litere mici) și în baza 2. Pentru afișarea numerelor în baza 2 se vor folosi operatori pe biți. **Nu se poate folosi funcția showBits din laborator.** *Exemplu:* se citesc numerele 5 și 10 ($n=2$) și se va afișa pe un rând 5, 5, 5, 00...00101, iar pe următorul rând 10, 12, A, 00...01010. (Cele trei puncte dintre zerouri reprezintă zerouri pentru că în diferite calculatoare int-ul este pe 2 octeți sau pe 4 octeți.)

Problema 3

Se citesc de la tastatură numere întregi până la introducerea numărului 0. Scrieți un program care analizează pe rând numerele introduse (adică în timpul citirii acestora) și, pentru acele numere care au setați biții de pe pozițiile 0, 3 și 7, afișează valoarea originală a numărului, iar apoi, pe același rând, după un spațiu, valoarea numărului care se obține din numărul original prin resetarea celor 3 biți de pe pozițiile amintite. Pentru celelalte numere nu se face nimic. *Exemplu:* se citesc numerele 137, 1, 5, 0. Numărul 137 are reprezentarea binară 1000 1001, biții de pe pozițiile 0,3,7 sunt pe 1, ca urmare se va afișa pe ecran 137 0, deoarece prin resetarea acelor biți valoarea numărului devine 0. Restul numerelor, adică 1 și 5, nu îndeplinesc condiția, ca urmare se afișează doar 137 0.

Problema 4

Să se citească de la tastatură $n \leq 100$ numere întregi pozitive. Să se memoreze toate aceste n numere într-un vector. După citirea și memorarea tuturor numerelor în vector, să se genereze pentru fiecare număr citit x , numărul y format din biții nenuli care alcătuiesc numărul x . Considerăm pe k ca fiind numărul de biți de 1 din numărul x . Numărul y va avea, de fapt, ultimii k biți setați pe 1. Se va afișa pentru fiecare număr reprezentarea internă a numărului x și a numărului generat, y . Se vor utiliza operatorii pe biți. **Nu se poate folosi funcția `showBits` din laborator.** *Exemplu:* se citesc numerele 2 și 10 ($n=2$). Se va afișa pe ecran 00...00010 00..00001 pe un rând, iar pe următorul rând 00...01010 00...00011. (Cele trei puncte dintre zerouri reprezintă zerouri pentru că în diferite calculatoare `int`-ul este pe 2 octeți sau pe 4 octeți.)

Problema 5

Să se citească de la tastatură un vector de $n \leq 50$ numere întregi. Scrieți un program care, după citirea și memorarea tuturor numerelor în vector, creează un nou vector cu aceeași dimensiune ca vectorul anterior creat, în care fiecare element va fi elementul de pe aceeași poziție din vectorul inițial împărțit la 16. Împărțirea se va face cu operații pe biți. Programul va afișa vectorul nou obținut. *Exemplu:* se citesc numerele 16, 32 și 50 ($n=3$). Se va afișa 1, 2, 3.

Problema 6

Se citesc $n \leq 10$ perechi de numere întregi pozitive de la tastatură și se memorează într-un vector. După citirea și memorarea tuturor perechilor de numere în vector, din reprezentarea internă a primului număr din pereche se va determina numărul k de biți de 1. Se vor seta pe 1 ultimii k biți ai celui de-al doilea număr din pereche (k reprezintă numărul biților de 1 din reprezentarea internă a primului număr din pereche). Să se afișeze reprezentarea internă a numerelor inițiale și apoi a celui de-al doilea număr din pereche modificat. Se vor utiliza operatorii pe biți. **Nu se poate folosi funcția `showBits` din laborator.** *Exemplu:* se citesc numerele 12 și 8 ($n=1$). Se va afișa pe ecran 00...01100 00..01000 00...01011 pe un rând. (Cele trei puncte dintre zerouri reprezintă zerouri pentru că în diferite calculatoare `int`-ul este pe 2 octeți sau pe 4 octeți.)