**Protejat: Laborator 03**

**1. Discutarea unor teme din curs**

* tipuri de date
* conversii intre tipuri de date
* operatori pentru numere întregi/reale: prioritate și asociativitate
* operarorii virgulă, ? :, sizeof()

**2. Exerciții**

1. (0,5p) Se consideră funcția f:R->R, continuă pe intervalul [a,b], cu f(a)\*f(b)<0. Să se scrie un program C++, care să determine o soluție a ecuației f(x)=0, folosind metoda următoare:  
   *Dacă a<b, se calculează m=(a+b)/2, apoi f(m). Dacă f(m) este foarte apropiat de zero, atunci m este o soluție a ecuației. Dacă nu, atunci se verifică dacă graficul funcției ar intersecta cel puțin o dată axa Ox, pe intervalul [a,m], respectiv [m,b] și se rezolvă problema doar pe acel interval.*
2. (0,25p) Să se scrie un program C++ care să determine rădăcina pătrată a numărului real pozitiv *x*, ca limită a șirului de numere an, definit astfel: a0=1, an=1/2\*(an-1+x/an-1). Se va evita folosirea vectorilor.
3. (0.5p) Să se definească un tip de date pentru reprezentarea numerelor complexe şi să se scrie secvențe de cod sau funcţii pentru diferite operaţii cu astfel de numere: adunare, scădere, înmulțire și împărțire.
4. (0,4p) Definiți un tip de date adecvat pentru reprezentarea unei date calendaristice, apoi scrieți o secvență de cod sau o funcție pentru determinarea zilei următoare unei zile date.
5. (0,75p) Definiți un tip de date adecvat pentru reprezentarea unei partide în jocul de șah, stocând date despre:
   1. numele jucătorilor cu piesele albe, respectiv negre;
   2. cine este jucătorul care urmează să mute;
   3. așezarea curentă a pieselor de ambele culori, pe tabla de șah.  
      Cu această structură definită, scrieți o secvență de cod pentru a plasa pe tabla de șah toate piesele așa cum sunt ele la începutul jocului de șah.
6. (0.75p) Definiți o structură de date pentru a reprezenta datele personale ale unor concurenți la un concurs (nume, prenume, vârstă, sex), precum și punctajul obținut de fiecare concurent. Scrieți un program C++ care să citească aceste date, apoi, folosind un algoritm de sortare, ordonați descrescător persoanele de sex masculin cu vârste între 16 și 20 ani, în funcţie de punctaj şi afișați lista ordonată.
7. (0.25p) Să se exemplifice folosirea reuniunilor (union) pentru citirea şi afişarea datelor despre mai multe persoane de ambele sexe.
8. (0.5p) Definiți o structură de date corespunzătoare pentru a memora simpatiile de diferite grade dintre cele *NrF* fete și *NrB* băieți dintr-o grupă de studenți, știind că avem relații de simpatie atât de la fete pentru băieți, cât și de la băieți pentru fete. Scrieți un algoritm pentru a determina care este cea mai simpatizată fată din grupă, ca sumă a simpatiilor băieților. De asemenea, scrieți un algoritm pentru a determina care sunt băieții care nu sunt deloc simpatizați de această fată.
9. (0,1p) Definiți o structură de date pentru jocul *Snake*, în care pe tablă (de culoare galbenă) se află șarpele (reprezentat de pătrățele verzi, cu excepția capului care este roșu), precum și diferite fructe (de culori roșu (1p) sau albastru (2p)), care pot fi mâncate de șarpe. Codificați printr-un câmp al structurii de date definite și direcția de deplasare a șarpelui. Scrieți o secvență de cod sau o funcție pentru a verifica dacă șarpele lăsat în pace, în situația curentă, ar urma să se mânânce pe sine, să iasă din tabla de joc sau să mânânce un fruct.
10. Definiți o structură de date adecvată pentru a stoca datele despre codul numeric personal al unei persoane, apoi scrieți o secvență în C++ (o funcție) care să verifice dacă un anumit CNP este corect. Informații despre CNP se găsesc la adresa: <https://ro.wikipedia.org/wiki/Cod_numeric_personal>
11. (0.75p) Să se parcurgă o matrice cu *n* linii și *m* coloane sub formă de spirală și să se populeze celulele matricei cu primele *nxm* numere prime.
12. (1p) Să se parcurgă o matrice patratică în zig-zag, începand din colțul stânga sus, prin deplasări dreapta, apoi stânga-jos, apoi jos, apoi dreapta sus ș.a.m.d.
13. (0.25p) Scrieţi o funcţie *bitcount(n)* care să contorizeze numărul de biţi pe 1 dintr-un argument întreg.
14. (0.5p) Scrieţi o funcţie *getbits(x, p, n)* care să returneze (cadrat la dreapta) câmpul de lungime *n* biţi al lui *x*, care începe la poziţia *p*.