

**Examen – Programarea Calculatoarelor**  
**Seria CD**

**Data: 30 ianuarie 2018**

**Durata: 90 de minute**

**SUBIECT 1 (28 puncte + 2 puncte bonus)**

După ce ați absolvit cursul de Programarea Calculatoarelor, ați primit de rezolvat următoarea problemă de monitorizare a tranzacțiilor financiare. Aceasta poate fi modelată folosind următoarele elemente:

- Sursă de bani
    - Identificator sursă – număr întreg fără semn, maxim un byte
    - Timestamp plată inițială – șir de 10 caractere în formatul "zz/ll/aaaa"
    - Lungime lanț tranzacții – număr întreg fără semn
    - Lanț de tranzacții – pointer către prima tranzacție efectuată din această sursă
  - Tranzacție
    - Nume persoană – șir de caractere
    - Suma - număr double care va conține doar valori pozitive
    - Următoarea tranzacție din lanțul curent, dacă este ultima tranzacție trebuie să fie setat cu NULL
- 1.1. Să se definească tipurile de date *TSursa* și *TTranzactie*. *TSursa* va conține câmpurile: *id*, *lung* și *lanțTranzactii*, iar *TTranzactie* va conține câmpurile: *nume*, *suma* și *urmTranzactie* (3 puncte)
- 1.2. Să se scrie o funcție care alocă memorie pentru un element de tipul *TSursa* cu antetul:  
*TSursa \*alocaSursa(unsigned int lungTranz)* (3 puncte)
- 1.3. Scrieți o funcție care alocă memorie pentru *nrSurse*, cu următoarea semnătură:  
*TSursa \*\*alocaSurse(unsigned int nrSurse, unsigned int \*lungTranz)*  
unde *nrSurse* reprezintă numărul de surse de venit analizate. (2 puncte)
- 1.4. Definiți o funcție care citește datele de intrare dintr-un fișier text deschis deja pentru citire. În fișier, întâi este salvat un *unsigned int* care reprezintă numărul de surse de bani. După aceea, în fișier se găsesc toate informațiile necesare completării unei surse de bani (alegeți voi ordinea în care acestea se găsesc în fișier). (6 puncte)
- 1.5. Implementați o funcție care detectează un ciclu constant (începe și se termină cu aceeași persoană, iar toate tranzacțiile implicate au aceeași valoare numerică) în lanțul de tranzacții al unei surse. În caz că există un astfel de ciclu, eliminați-l din lanț:  
*void detecteazaSiEliminaCiclu(TSursa \*sursa)* (7 puncte)
- 1.6. Realizați o funcție care să calculeze persoana cu cele mai multe tranzacții în toate lanțurile dintr-un an dat.  
*char\* persoanaTranzactiiMax(TSursa \*\*surse, unsigned int an)* (6 puncte)
- 1.7. Definiți o funcție care eliberează toată memoria alocată:  
*dezalocaSurse(TSursa \*\*surse, unsigned int nrSurse)* (3 puncte)

**SUBIECT 2 (12 puncte = 3 x 4 puncte)**

Care este ieșirea (warning-uri sau erori de compilare, erori de execuție, mesaje afișate, comportament nedefinit) secvențelor de cod de mai jos? Justificați pe scurt, în maxim 5 rânduri, răspunsul vostru. Pentru fiecare exercițiu, considerați ca toate headerele standard (*stdio.h*, *stdlib.h*, etc.) sunt incluse automat.

2.1	2.2	2.3
<pre>int main(void) {     char p[20];     char *s = "string";     int length = strlen(s);     int i;      printf("%d %d\n",         sizeof(s), sizeof(p));      for (i = 0; i &lt; length; i++)         p[i] = s[length - i];     printf("Rezultatul este: %s!", p);     return -1; }</pre>	<pre>int mul(int a, int b, int c) {     return a * b * c; }  int main(void) {     int (function_pointer)(int, int, int);     function_pointer = mul;     printf("%d %d\n",         function_pointer(2, 3, 4), 2*3*4);     return -1; }</pre>	<pre>#define N 3 void g(int a[], int x) {     int i = 0;     for (; i &lt; N; i++)         printf("%d ", a[i]);     printf("\n"); }  int f(int a[], int x) {     int i;     for (i = x; i &lt; N; i++) {         a[x] += N;         if (x == N - 1)             g(a, x);         else {             f(a, x + 1);    a[x + 1] = 0;         }     } }  void main(void) {     int a[N] = {0};    f(a, 0); }</pre>

