

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

**Data: 5 februarie 2017**

**Durata: 90 de minute**

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

La finalul anului 1, aveți de implementat o parte din sistemul dezvoltat de către compania de transport Uber. Pentru aceasta, veți porni de la următorul model al datelor:

- Un șofer va avea informațiile:
  - Nume, reprezentat ca șir de caractere care va conține maximum 20 caractere utile
  - Număr mașină, șir de caractere format din 8 caractere utile (de forma JJNNNNLL – J=Județ, N=Număr, L=Literă, de ex. "B 103PPP", "IF12 CCC", "B 1 ACS", "CJ123ABC")
  - Locația curentă pe hartă, salvată ca două numere reale cu precizie dublă
  - Numărul de curse efectuate, întreg fără semn
  - Informații despre toate cursele efectuate, folosind alocare dinamică și structura de mai jos
- O cursă ca conține următoarele detalii:
  - Locație început cursă, salvată ca două numere reale cu precizie dublă
  - Starea cursei reprezentată pe un byte (folosind bitfields, măști de biți sau orice altă variantă, explicați soluția voastră) în care rețineți: cursa anulată sau nu, cursă în desfășurare, nr stele (între 0-5)

1.1. Definiți tipurile de date *TSofer* și *TCursa*. Tipul *TSofer* conține câmpurile nume, nrMasina, locatieX, locatieY, nrCurse și curse (de tip *TCursa\*\**). Tipul *TCursa* are câmpurile startX, startY și stare. Dacă adăugați câmpuri adiționale, explicați în comentarii rolul lor. (4 puncte)

1.2. Scrieți o funcție care alocă memorie pentru un șofer cu *nrCurse* curse efectuate, cu semnătura  
*TSofer\* alocSofer(int nrCurse)* (2 puncte)

1.3. Scrieți o funcție care alocă memorie pentru *nrSoferi* șoferi cu următoarea semnătură, unde *nrCurseSoferi* este un vector de dimensiune *nrSoferi*:  
*TSofer\*\* alocSoferi(int nrSoferi, int\* nrCurseSoferi)* (2 puncte)

1.4. Definiți o funcție care citește datele de intrare dintr-un fișier binar deschis deja pentru citire. În fișier sunt salvate întâi un int care reprezintă numărul de șoferi, apoi *nrSoferi* numere întregi care reprezintă numărul de curse efectuate de fiecare șofer, iar apoi datele pentru fiecare șofer (și cursele asociate) în parte. Funcția trebuie să întoarcă atât numărul de șoferi citiți din fișier (variabilă *int*), cât și datele despre șoferi (variabilă de tip *TSofer\*\**). (6 puncte)

1.5. Implementați o funcție care să calculeze care este șoferul cu cel mai mare rating mediu (*rating mediu = sumă număr stele / (număr curse total - anulate - în desfășurare)*) să fie maximă și întoarceți numele acestuia:  
*char\* numeSoferRatingMaxim(TSofer\*\* soferi, int nrSoferi)* (6 puncte)

1.6. Definiți o funcție care să întoarcă primii *nr* șoferi cei mai apropiați (folosind distanța euclidiană) de un punct de pornire (x,y). Semnătura funcției este:  
*TSofer\*\* maxVolume(TSofer\*\* soferi, int nrSoferi, int x, int y, int nr)* (7 puncte)

1.7. Definiți o funcție care eliberează toată memoria alocată: *dezalocaSoferi(TSofer\*\* soferi, int nrSoferi)* (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 M(const int n); int F(const int n) {     return (n == 0) ? 1 : n - M(F(n - 1)); } int M(const int n) {     return (n == 0) ? 0 : n - F(M(n - 1)); } int main(void) {     int i;     for (i = 0; i &lt; 6; i++)         printf("%2d ", F(i));     printf("\n");     return 0; }</pre>	<pre>int* func(int* a, int* b, int n) {     int c[100];     int i;     printf("%d\n", n);     for (i = 0; i &lt; n; ++i)         c[i] = a[i] + b[i];     return c; }  int main(void) {     int a[] = {1,2,3,4};     int b[] = {5,6,7,8};     int* c = func(a, b, sizeof(a));     return 0; }</pre>	<pre># define A(n) ((n)%(1&lt;&lt;1)) # define B(n) (!(n&amp;1))  int main(void){     int i;     for (i = 0; i &lt; 6; i++){         printf("One %d %d %d\n",             i, A(i), B(i));     }     return 0; }</pre>