

Examen – Programarea Calculatoarelor
Seria CD

Data: 2 februarie 2016

Durata: 90 de minute

SUBIECT 1 (28 puncte + 2 puncte bonus)

În cadrul internshipului de la terminarea primului an de facultate, aveți de implementat un program care gestionează informațiile dintr-o bibliotecă. Pentru aceasta, toate cărțile din bibliotecă sunt salvate ca o colecție de autori, iar pentru fiecare autor știți volumele sale. Informațiile reținute sunt următoarele:

- Numele autorului, salvat ca un șir de caractere încheiat cu null terminator
- Codul ISO al țării în care este născut autorul, salvat ca un șir de 2 caractere (de ex. RO sau FR)
- Un număr întreg care reprezintă numărul de volume scrise de autor care se află în bibliotecă
- O secvență de volume, iar pentru fiecare volum sunt prezente următoarele date:
 - Numele volumului, salvat ca un șir de caractere încheiat cu null terminator
 - Anul de publicare al volumului salvat ca un întreg scurt
 - Starea volumului salvată pe un octet: bitul cel mai puțin semnificativ înseamnă carte donată (0) sau cumpărată (1), al doilea bit dacă cartea este împrumutată (1) sau nu (0), iar următorii biți conțin alte informații despre starea ei fizică (gradul de deteriorare, etc.)
 - Un număr întreg care codifică identificatorul (între 1-10000) ultimei persoane care a împrumutat volumul

1.1. Definiți tipurile de date *TAutor* și *TVolum* cu care veți salva în memorie datele. *TAutor* conține câmpurile *numeAutor*, *codTara*, *nrVolume*, *volume* (de tip *TVolum**). Tipul *TVolum* are câmpurile *numeVolum*, *anPublicare*, *stare* și *idPersoana*. Dacă adăugați câmpuri adiționale, explicați în comentarii rolul lor. (3 puncte)

1.2. Scrieți o funcție care alocă memorie pentru un autor cu *nrVolume* cărți, cu semnătura
TAutor alocAutor(int nrVolume)* (3 puncte)

1.3. Scrieți o funcție care alocă memorie pentru *nrAutori* autori cu următoarea semnătură, unde *nrVolumeAutor* este un vector de dimensiune *nrAutori*:
*TAutor** alocAutori(int nrAutori, int* nrVolumeAutor)* (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 autori, apoi *nrAutori* numere întregi care reprezintă numărul de volume al fiecărui autor, iar apoi datele pentru fiecare autor (și volumele asociate) în parte. Funcția trebuie să întoarcă atât numărul de autori din fișier (variabilă *int*), cât și toți autorii citați din fișier (variabilă de tip *TAutor***). (6 puncte)

1.5. Implementați o funcție care să calculeze câte volume sunt în bibliotecă pentru autori dintr-o anumită țară:
*void nrVolumeTara(TAutor** autori, int nrAutori, char codTara[])* (5 puncte)

1.6. Definiți o funcție care calculează cine este persoana care a împrumutat cele mai multe cărți și nu le-a returnat (flag-ul de împrumutat trebuie să fie 1). Semnătura funcției este: *int maxVolume(TAutor** autori, int nrAutori)* (8 puncte)

1.7. Definiți o funcție care eliberează toată memoria alocată: *dezalocaScoli(TScoala** autori, int nrAutori)* (3 puncte)

SUBIECT 2 (12 puncte = 3 x 4 puncte)

Care este ieșirea (warning-uri sau erori de compilare, erori de execuție, mesaje afișate la stdout / stderr, existența unui comportament nedefinit) următoarelor secvențe de cod? 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, drept urmare nu influențează compilarea programului.

2.1	2.2	2.3
<pre>void f(int i, int j) { if (j > 0) f(i, j-1); printf(" %d * %d = %d \n", i, j, i*j); } int main() { f(9 , 9); return 0; }</pre>	<pre>int main() { char s[]="abbacdde"; int i; i=0; while (i<strlen(s)-1) if (s[i]==s[i+1]){ strcpy(s+i,s+i+2); if (i>0) i=i-1; } else i=i+1; printf("%s\n", s); return 0; }</pre>	<pre>typedef int (*funcptr_t)(int); int myfoo(int i) { printf("%d\n", i + 1); return i; } funcptr_t foo(int i) { printf("%d\n", i + 1); return myfoo; } int main() { funcptr_t foptr = foo(0); foptr(10); printf("%p %p\n", foptr, myfoo); return 0; }</pre>

