

Examen – Programarea Calculatoarelor
Seriile CB & CD

Data: 29 august 2016

Durata: 90 de minute

SUBIECT 1 (28 puncte + 2 puncte bonus)

După terminarea cursului de programare, o prietenă vă roagă să îl ajutați să rezolve următoarea problemă. Ea este foarte pasionată de jocuri pe calculator și a făcut rost de o serie de loguri cu rezultatele meciurilor între jucători. Veți modela problema folosind următoarele două structuri:

- O structură *jucător* care conține:
 - numele jucătorului curent
 - id-ul jucătorului
 - Data nașterii salvată ca un string cu 10 caractere de forma *zz//aaaa*
- O structură *meci* care conține rezultatul meciului direct între doi jucători:
 - Pointer către primul jucător *player1* și către al doilea jucător *player2*
 - Id-uri pentru cei doi jucători
 - Scorul meciului - un număr întreg (pozitiv dacă *player1* a câștigat, zero remiză, negativ *player2* a câștigat)
- O structură *date log* care conține un pointer către toți jucătorii și un pointer către toate meciurile, plus numărul de jucători și de meciuri din setul de date.

1.1. Definiți tipurile de date *TJucator* (câmpurile nume, id și datan), *TMeci* (câmpurile player1, player2, id1, id2 și scor) și *TLog* (câmpurile jucatori, meciuri, nrj și nrm). Dacă adăugați câmpuri adiționale, explicați rolul lor. (3 puncte)

1.2. Scrieți trei funcții care alocă memorie pentru un jucător cu numele de lungime *lung*, pentru un meci, respectiv pentru un log cu *nrj* jucători și *nrm* meciuri, cu semnăturile:

TJucator alocJucator(int lung) TMeci* alocMeci() TLog* alocLog(int nrj, int nrm)* (5 puncte)

1.3. Definiți o funcție care citește datele de intrare dintr-un fișier binar deschis deja pentru citire. În fișier sunt salvate întâi două numere întregi care reprezintă numărul de jucători, respectiv de meciuri. După aceea sunt salvați toți jucătorii (întâi lungimea numelui ca întreg, apoi celelalte câmpuri), apoi toate meciurile (doar id1, id2 și scor). Funcția trebuie să întoarcă o variabilă de tip *TLog**. (6 puncte)

1.4. Implementați o funcție ajutătoare folosită după citire, care asignează pointerul corect către fiecare jucător dintr-un meci, dacă id-ul acestuia există în setul de jucători:

void adaugaPointeriJucatori(TLog log)* (5 puncte)

1.5. Definiți o funcție care calculează și afișează (sub forma *nume:nrvictorii*) câte victorii are fiecare jucător:

void afiseazaVictorii(TLog log)* (5 puncte)

1.6. Ce linii trebuie să adăugați funcției anterioare pentru a determina cel mai bun jucător născut în 1993? (3 puncte)

1.7. Definiți o funcție care eliberează toată memoria alocată: *dezaLocaLog(TLog* log)* (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, string.h, etc.) sunt incluse automat, drept urmare nu influențează compilarea programului.

2.1	2.2	2.3
<pre>char *f1(char *s, int n, size_t total) { return (n > 0) ? strcat(f1(s, n - 1, total + strlen(s)), s) : strcpy(malloc(total + 1), ""); } char *f2(char *s, int n) { return f1(s, n, 0); } int main() { char *s = "abc"; printf("%d %s", sizeof(s), f2(s, 3)); return 0; }</pre>	<pre>void swap(char *s, char* t) { char* aux = s; s = t; t = aux; printf("%s %s\n", s, t); } int main() { char *s = "abc"; char *t = "def"; swap(s, t); printf("%s %s\n", s, t); return 0; }</pre>	<pre>struct T { char c; int i; struct T* p; }; int main() { struct T* v1 = malloc(5*sizeof(struct T)); struct T v2[10]; printf("%d %d\n", sizeof(v1), sizeof(v2)); return 0; }</pre>