Question **1**

Complete

Mark 10.00 out of 10.00

Subiecte

Restricții (pentru ambele subiecte): Nu se vor folosi variabile globale.

5 puncte **1. Inventar**. Inventarierea produselor într-un lanț de magazine se poate face în felul următor: pentru fiecare unitate (magazin) se păstrează un fișier text care conține, pe linii, produsele și cantitățile disponibile (ex: `pasta_de_dinti 30`). Numele unui produs poate avea maxim 32 de caractere și nu conține spații. Cantitatea este un număr natural, nenul. Într-un fișier se pot găsi de mai multe ori, pe linii diferite, informații despre același produs. Un fișier are structura corectă.

Inventarul presupune să identificăm stocul total pentru fiecare produs în parte. Vă propunem scrierea unui program care primește un număr natural n, reprezentând numărul de fișiere cu produse, apoi un alt număr natural k, care reprezintă un stoc limită, apoi citește pe rând numele celor n fișiere text. Programul va crea o listă de produse și cantitatea totală pentru fiecare produs, pe baza înregistrărilor din fișierele date.

Programul va scrie pe ecran toate produsele care au cantitatea globală mai mică decât stocul limită dat (k) și va scrie într-un fișier denumit `inventar.txt` toate produsele și cantitatea lor.

5 puncte **2. Senzori**. Un fișier binar, `data.in`, conține mai multe informații de monitorizare colectate de la diferiți senzori, sub formă de seturi de date. Acestea au structura:

```
typedef struct {
    int nv;       /* numărul de valori monitorizate */
    peer* values; /* setul de valori monitorizate */
} data_set;
```

```
typedef struct {
    long timestamp;   /* timpul de măsurare (codificat ca întreg lung) */
    double value;     /* valoare măsurată de senzor */
} peer;
```

Scrieți un program care afișează informațiile de monitorizare de la senzori din pozițiile `pos1, pos2, ...` din fișier. Programul va fi rulat prin comanda `show pos1 pos2 ...` (pozițiile pot fi la întâmplare). Pentru a putea accesa un set de date din fisier se recomandă definirea unei funcții care întoarce poziția în dreptul unui set de date cu numărul de ordine specificat. Pentru citirea datelor în memorie, în vederea afișării lor, se va aloca dinamic memoria. Orice tablou auxiliar folosit va fi alocat/re-alocat dinamic în memorie. *Atenție*: **nu se va citit întreg fișierul în memorie**, ci se va folosi o variabilă de tip structură pentru citirea unui set de date.

1 puncte **3.** Funcția `strcpy(char *dest, const char *src)` realizează copierea șirului de caractere (inclusiv a terminatorului de șir) ce începe la adresa indicată de `src` în cadrul zonei de memorie indicată de pointerul `dest` și întoarce valoarea primită pentru pointerul `dest`. Următoarele premise sunt asiugrate:

1. Memoria aferentă zonei alocate pentru șirul destinație este deja alocată;
2. `src` este un șir de caractere valid;
3. Cele 2 zone de memorie nu se suprapun.

Să se implementeze funcția `char *my_strcpy(char *dest, const char *src)` care să aibă comportamentul lui `strcpy`.

Restricții: În cadrul implementării NU se vor folosi funcții din `string.h` și nu se va determina lungimea șirului (nici prin folosirea `strlen()`). Mai mult, nu se va folosi operatorul `[]`. O implementare care nu respectă restricțiile nu va fi punctată.

Hint: Aritmetica pointerilor.