

**ATENȚIONARE**

**1) Nefolosirea în rezolvare a unui proiect (format din cel puțin două fișiere – fișierele header nu se număra) și a fișierului header conduce la anularea punctajului obținut la rezolvarea celorlalte puncte din problemă. De asemenea, folosirea variabilelor globale conduce la anularea punctajului obținut.**

**2) Codul cu erori de compilare (inclusiv atenționări) și link-editare se notează cu 0 (zero).**

3) Nefolosirea funcțiilor indicate în barem atrage o penalizare de 50% din punctaj.

4) Punctajul maxim se acordă pentru rezolvarea **CORECTĂ** a fiecărei subprobleme (punct din barem).

Se reia enunțul din problema **P1**. În plus se definește tipul de date **FUNCTII** care o structură care cuprinde ca membri:

- ✓ un pointer la char (pentru denumirea prelucrării)
- ✓ un pointer la o funcție capabil să stocheze adresa unei funcții de prelucrare (prelucrarea se referă la ceea ce trebuie să facă programul).

Se reia problema P1 cu specificarea prelucrărilor care trebuie făcute (pentru problema P2 se va construi un alt proiect).

Prelucrările care trebuie realizate sunt următoarele:

**a)** determinarea producției medii de mere

**b)** determinarea numărului de pomi care trebuie tăiați. Un pom trebuie tăiat dacă producția de fructe realizată este sub o anumită valoare citită de la tastatură. Se transformă matricea într-o matrice binară (cu valorile elementelor egale numai cu 1 sau 0) astfel dacă valoarea producției este sub valoarea "critică" elementul ia valoarea 0, iar dacă producția este peste valoarea "critică" elementul ia valoarea 1. Se linarizează matricea binară și se stochează sub forma unui șir de biți (elementului de pe linia 0 și coloana 0 îi corespunde bitul 0). Se numără numărul de biți de 1, iar valoarea obținută este scăzută din numărul total de biți.

**Stocarea sub forma unui șir de biți și numărarea biților se face folosind operatori de lucru pe bit.**

**c)** știind că primul și ultimul pom de pe un rând sunt pomi din soiul 1, pe rândul (sau rândurile, după cum numărul de rânduri este sau nu impar) din mijloc sunt pomi din soiul 2, iar în rest sunt pomi din soiul 3, să se determine soiul cel mai productiv (care are producția medie cea mai mare).

Programul are o linie de comandă de următoarea formă:

`livada [-p][-t][-v][-h] [fisier_intrare] [fisier_iesire]`

În această linie de comandă **livada** este numele programului. Putem avea un singur tip de prelucrare la o rulare a programului (opțiunea de prelucrare este argumentul liniei de comandă care are ca prim caracter semnul '-'). Opțiunile de prelucrare pot fi (parantezele pătrate indică faptul că parametrii respectivi sunt opționali):

- p pentru prelucrarea de la punctul a).
- t pentru prelucrarea de la punctul b);

-v pentru prelucrarea de la punctul c).

-h pentru afișarea unui mesaj de help.

**fișier\_intrare** → fișierul de unde se citesc datele cu următorul format:

1. prima linie                      – numele livezii
2. a doua linie                    - numele tipului de pomi
3. a treia linie                    - numărul de rânduri și numărul de pomi pe rând  
(separate printr-un spațiu)
4. următoarele linii                      – pe fiecare linie din fișier sunt înscrise  
producțiile date de pomii de pe rândul  
corespunzător

**fișier\_iesire** → fișierul unde se înscriu rezultatele cu următorul format:

1. prima linie                      – numele prelucrării realizate
2. următoarele n+1 linii           - datele citite sub forma dată în problema P1  
punctul 5
3. ultima linie                      - rezultatul prelucrării

Opțiunea implicită de prelucrare este cea corespunzătoare opțiunii **-p**.

Dacă opțiunea de prelucrare este **-h**, atunci restul argumentelor din linia de comandă se ignoră și se afișează mesajul de help (care conține forma liniei de comandă și descrie ceea ce face programul).

În cazul în care în linia de comandă este specificată o altă opțiune decât **-h** (sau nu este specificată nici o opțiune) și nu sunt specificate fișiere, citirea datelor se face de la tastatură, iar afișarea rezultatului se face pe monitor. În cazul în care este specificat numele unui singur fișier, acesta se consideră a fi fișierul de intrare din care se citesc datele de intrare, iar rezultatul se afișează pe monitor.

Apelul prelucrării dorite se va face prin intermediul pointerilor la funcții.

Programul trebuie să realizeze următoarele operații:

1. analiza liniei de comandă
2. inițializarea unui tablou cu elemente de tip structura FUNCTII cu mesajele și funcțiile corespunzătoare;
3. citirea din fișierul de intrare a informațiilor referitoare la producția unei livezi (specificate în problema P1);
4. scrierea în fișierul de ieșire a informațiilor citite (ca în problema P1).
5. calcularea rezultatului cerut de opțiunea de prelucrare citită de la tastatură (folosindu-se pointeri la funcții).
6. scrierea rezultatului în fișierul de ieșire (trebuie să se respecte formatul fișierului de ieșire dat mai sus).

**Barem de notare**

Programul va prelua de la problema P1 punctele A1, A2, A3, A4, A5, A6, A8 (care nu se vor mai nota la problema P2). Se notează numai ceea ce este în plus față de P1.

**Tabel nr. 2**

<b>B1.</b> Analiza liniei de comandă	
<b>B1a.</b> Stabilește dacă se afișează sau nu mesajul de help	0,3
<b>B1b.</b> Stabilirea fișierelor cu care se lucrează	0,6
<b>B1c.</b> Deschiderea corectă a fișierelor (funcție care primește ca parametri doi pointeri la char și returnează un pointer la structura de tip FILE).	0,2
<b>B2.</b> Prelucrarea indicată la punctul <b>a)</b> . Funcția primește ca parametru un pointer la o structură LIVADA și returnează o uniune de tip <b>REZULTAT</b> .	0,3
<b>B3.</b> Prelucrarea indicată la punctul <b>b)</b> . Funcția primește ca parametru un pointer la o structură LIVADA și returnează o uniune de tip <b>REZULTAT</b> .	0,5
<b>B4.</b> Prelucrarea indicată la punctul <b>c)</b> . Funcția primește ca parametru un pointer la o structură LIVADA și returnează o uniune de tip <b>REZULTAT</b>	0,4
<b>B5.</b> Citirea datelor din fișier (funcții asemănătoare cu cele de la prima problemă, numai că de această dată citirea se face dintr-un fișier indicat printr-un pointer la FILE).	0,6
<b>B6.</b> Înscrierea rezultatelor într-un fișier (funcția are parametri corespunzători pentru înscrierea rezultatelor în fișier și nu returnează nimic).	0,4
<b>B7.</b> Inițializarea corectă a tabloului de structuri de tip FUNCTII	0,3
<b>B8.</b> Folosirea pointerilor la funcții pentru apelarea prelucrării dorite.	0,4
<b>TOTAL TABEL 2</b>	4 p