

Indicații pentru Tema 9

- **Tema 9** va avea **12 probleme**, fiecare punctată cu note de la 1 la 10 astfel:
 - Indentare corespunzătoare = **1 punct**
 - Utilizarea acoladelor, lizibilitatea codului = **1 punct**
 - Utilizarea corespunzătoare a instrucțiunilor de tip **if, for, while, do...while** = **1 punct**
 - Declararea corespunzătoare de variabile = **1 punct**
 - Funcționalitatea corespunzătoare a programului (adică programul face ce trebuie să facă) = **2 puncte**
 - Utilizarea corespunzătoare a structurilor = **1 punct**
 - Utilizarea corespunzătoare a pointerilor = **2 puncte**
 - Citirea și afișarea corespunzătoare a variabilelor/ meniurilor = **1 punct**
- Tema trebuie încărcată pe Campus Virtual până cel târziu **MIERCURI, 18 decembrie 2019, ora 23:59 (hard deadline)**.
- **TOATE PROBLEMELE SUNT OBLIGATORII!!**
- Vor fi încărcate 12 fișiere cu extensia **.c** denumite astfel: **p1.c, p2.c, p3.c, p4.c, p5.c, p6.c, p7.c, p8.c, p9.c, p10.c, p11.c** și **p12.c** corespunzătoare rezolvărilor problemelor 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, respectiv 12.
- **NU încărcăți proiecte CodeBlocks!!!** Numai fișierele cu extensia **.c** denumite corespunzător după cele menționate la subpunctul anterior.
- **IMPORTANT!!** Dacă nu compilează codul, problema este notată cu 0. Punctul din oficiu pe fiecare problema se acordă în cazul în care codul compilează și nu are nicio eroare de compilare!
- Tema este **OBLIGATORIE** și cei care nu o fac nu vor putea da Testul 3!!
- Copiatul este strict interzis! Dacă sunt găsite astfel de cazuri, toți studenții implicați (adică cei care copiază și cei de la care se copiază) vor fi exmatriculați.
- Încercați să rezolvați voi singuri problemele. Dacă aveți nevoie de ajutor, vă stau la dispoziție.
- Pentru orice fel de întrebări nu ezitați să mă contactați.

Tema 9

Problema 1

Să se creeze o structură **distanțe**. Structura va conține o matrice pătratică cu distanțele dintre N localități, dimensiunea matricii și un tablou de șiruri de caractere cu numele orașelor. De la tastatură vor fi citite cele N orașe (citirea vectorului de șiruri de caractere), iar apoi distanțele dintre ele (citirea matricii cu distanțele dintre localități). Să se afișeze toate orașele și distanțele corespunzătoare de la fiecare oraș către celelalte orașe. **Trebuie să utilizați alocare dinamică pentru tabloul de șiruri de caractere.**

Exemplu: Timisoara 0 54 67
Arad 54 0 98
Lugoj 67 98 0

Problema 2

Să se definească o structură pentru memorarea numerelor mari, cu maxim 1000 de cifre. Structura va conține două câmpuri: un întreg, care va memora numărul de cifre al numărului (număr mai mic sau egal cu 1000), și un tablou de întregi, care va memora cifrele numărului. Fiecare cifră a numărului se va memora într-un element al vectorului. Scrieți o funcție pentru **afișarea numărului memorat** într-o astfel de structură, o altă funcție pentru **citirea de la tastatură și memorarea** unui astfel de număr în structura respectivă, și o altă funcție pentru **adunarea a două numere mari** cu memorarea rezultatului într-un al treilea număr mare, toate cele trei fiind date ca argumente la funcția respectivă. Folosind cele trei funcții, în programul principal citiți de la tastatură două numere mari, afișați-le și calculați și afișați suma acestora.

Exemplu: dacă se citește 4 și 1234 pentru primul număr mare, și 5 cu 12345 pentru al doilea număr mare, pe ecran vor fi afișate:

Primul numar mare: 1234
Al doilea numar mare: 12345
Suma numerelor mari: 13579

Problema 3

Să se asocieze fiecărei litere din alfabet (considerăm numai litere mici) un punctaj (număr întreg mai mare sau egal cu zero). În aceste condiții, unui cuvânt i se poate asocia un scor care se calculează ca sumă a punctajelor literelor cuvântului. Se citește de la tastatură un număr N, urmat de **N perechi <literă, punctaj>**. Literele al căror punctaj nu este specificat vor avea punctajul zero. Definiți o structură în care se poate memora o literă împreună cu punctajul ei. Citiți în mod repetat cuvinte de la tastatură până la introducerea cuvântului "gata" și afișați scorul fiecărui cuvânt utilizând o funcție care ia ca argumente un cuvânt și vectorul cu perechile <literă, punctaj> și returnează scorul cuvântului pe baza punctajelor fiecărei litere.

Problema 4

Să se rezolve **Problema 3** utilizând alocare dinamică pentru tabloul de N perechi <literă, punctaj>. Scrieți o funcție pentru eliberarea de memorie necesară și utilizați-o corespunzător.

Problema 5

Să se rezolve **Problema 4** alocând dinamic și structura propriu-zisă care conține perechi de tipul <literă, punctaj>. Se știe clar câte elemente vor fi alocate dinamic de la bun început, toate literele alfabetului, adică 26 de elemente. **Indicație:** veți folosi un vector de pointeri către structuri și veți alocă vectorul de pointeri către structuri și structurile respective.

Problema 6

Să se citească o listă de N persoane cu următoarele date: Nume, Prenume, Sex, Înălțime. Să se creeze o funcție de citire care citește datele unei persoane, le pune într-o structură și returnează structura. Pentru Nume și Prenume utilizați maxim 30 de caractere. Afișați persoanele adăugate în listă.

Problema 7

Să se rezolve **Problema 6** alocând dinamic câmpurile de șiruri de caractere Nume și Prenume. Modificați în acest sens funcția de citire și modalitatea în care puneți datele citite în structură. Modificați programul astfel încât atunci când se introduce o persoană nouă, să se verifice dacă această persoană (identificată cu Nume, Prenume, Sex și Înălțime) nu a mai fost introdusă deja listă pentru a nu introduce aceeași persoană de două ori. Afișați persoanele adăugate în listă.

Problema 8

Să se rezolve **Problema 7** alocând dinamic și vectorul de persoane. Creați un meniu cu următoarele opțiuni: 1. Citire 2. Ștergere 3. Afișare 4. Ieșire . La citire se va citi o singură persoană și va fi memorată doar dacă nu apare deja în listă (condiția introdusă la Problema 7). În cazul în care este citită o persoană care apare deja în listă, se va afișa un mesaj că persoana respectivă există deja în listă. Funcția de ștergere va șterge o persoană citită de la tastatură. Dacă nu există persoana cu numele, prenumele, sexul și înălțimea introduse de la tastatură, se va afișa un mesaj că persoana respectivă nu există în listă. La afișare afișați fiecare persoană din listă una sub alta cu Nume Prenume Sex Înălțime separate de câte un spațiu.

Exemplu: La afișare: Popescu Ion M 160
Cartarescu Mircea M 170

Problema 9

Să se completeze meniul de la **Problema 8** astfel încât să apară 2 noi opțiuni: Afișare în ordine alfabetică după nume a băieților și Afișarea fetelor în ordinea descrescătoare a înălțimii. Ambele afișări vor fi sub formă tabelară (lățimea unei coloane va fi lățimea celui mai lat cuvânt care va fi afișat în acea coloană + 1 spațiu liber). Pentru toate opțiunile din meniu utilizați o funcție care să facă totul, fără a scrie cod suplimentar pe lângă apelul funcției în main la switch case.

Exemplu: dacă se cere o afișare sub formă tabelară veți afișa băieții sub forma următoare:

```
+-----+-----+-----+-----+
|Nume      |Prenume   |Sex |Inaltime |
+-----+-----+-----+-----+
|Cartarescu|Mircea    |M   |170      |
|Mihailovici|Daniel    |M   |150      |
|Popescu    |Stefanut  |M   |160      |
+-----+-----+-----+-----+
```

Problema 10

Să se gestioneze un dicționar de cuvinte ca un vector de structuri de date care vor conține câte un cuvânt și explicația acestuia. Considerăm un cuvânt ca având maxim 50 de caractere, iar explicația maxim 1000 de caractere. Să se formeze un dicționar cu N cuvinte și explicații citite de la tastatură. Se citește un nou cuvânt de la tastatură și dacă acesta există în dicționarul creat anterior, va fi afișată pe ecran explicația acestuia, dacă nu, se va afișa un mesaj sugestiv. **Indicație:** Aveți opțiunea de a crea o singură structură dicționar cu un vector de cuvinte și vector de explicații, sau de a crea două structuri, o structură care conține o pereche <cuvânt, explicație>, și un dicționar care va conține un vector de perechi <cuvânt, explicație>.

Problema 11

Să se rezolve **Problema 10** alocând dinamic cuvintele și explicațiile acestora pentru a utiliza spațiul de memorie minim necesar. Scrieți o funcție pentru citirea unui element în dicționar și o funcție afișarea dicționarului. Transformați în funcție cerința de la Problema 10 referitoare la căutarea unui cuvânt în dicționar și afișarea explicației aferente acestuia.

Problema 12

Să se rezolve **Problema 11** alocând dinamic și dicționarul de cuvinte și explicații. Utilizați alocarea dinamică pentru alocarea vectorului de perechi <cuvânt, explicație>. Creați un meniu cu următoarele opțiuni: 1. Introducere cuvânt si explicatie 2. Afișare dicționar 3. Cautare cuvânt in dicționar 4. Iesire .