

Programarea calculatoarelor

– Seminar nr. 4 –

Funcții. Fișiere text. Tablouri

1. Scrieți o funcție care rearanjează elementele unui tabloului unidimensional cu n componente întregi astfel încât toate valorile nule să se afle la sfârșitul tabloului. Ordinea în cadrul secvenței de elemente nenule nu contează. Scrieți un program care citește de la tastatură două numere naturale n și k ($2 \leq k \leq n \leq 100$), apoi un șir format din n numere întregi cu cel mult 4 cifre fiecare și, folosind apeluri utile ale funcției definite anterior, verifică dacă șirul conține cel puțin k valori nenule, nu neapărat distincte. În caz afirmativ programul afișează, în fișierul text *date.out*, k dintre acestea, iar altfel mesajul NU EXISTA.
2. Scrieți o funcție *sumad* care primește prin parametrul x un număr natural și returnează suma divizorilor numărului x , mai puțin el însuși. Scrieți un program care citește de la tastatură un număr natural n și afișează, folosind apeluri utile ale subprogramului *sumad*, perechile de numere prietene de forma (a, b) , cu $1 \leq a < b \leq n$. Două numere naturale se numesc *numere prietene* dacă suma divizorilor fiecărui număr (mai puțin el însuși) este egală cu celălalt număr.
3. Scrieți o funcție care calculează cifra minimă și cifra maximă pentru un număr natural dat ca parametru. Folosind apeluri utile ale acestei funcții, scrieți un program care afișează toate numerele având toate cifrele egale și sunt mai mici sau egale decât un număr natural n citit de la tastatură.
4. Scrieți o funcție care să calculeze valoarea maximă dintr-o secvență cuprinsă între doi indici i și j ($0 \leq i \leq j < n$) ai unui tablou unidimensional format din n numere întregi (elementele tabloului se citesc, printr-o funcție, din fișierul text *numere.in*). Scrieți o funcție care, folosind apeluri utile ale funcției definite anterior, afișează mesajul DA în cazul în care un tablou unidimensional de numere întregi, primit ca parametru, are toate elementele egale sau mesajul NU în caz contrar.
5. Scrieți o funcție *cautare* care să returneze prima poziție pe care apare într-un tablou unidimensional o valoare având o anumită proprietate sau -1 dacă nu există nicio astfel de valoare. Funcția *cautare* va avea un parametru de tip pointer la funcție prin care va primi o funcție care implementează criteriul de selecție dorit.
6. Operații cu două numere complexe. Definiți o structura COMPLEX care să permită memorarea părții reale, respectiv a părții imaginare a unui număr complex de forma $x+yi$. Scrieți funcții care implementează operațiile de adunare, scădere, înmulțire și respectiv împărțire a două numere complexe. Reluați pentru altă pereche de numere de câte ori dorește utilizatorul. (*switch*, *funcții callback*)

7. Să se citească (cu ajutorul unei funcții de citire) următoarele informații despre elevii participanți la un concurs de admitere: nume, numărul de înscriere și cele trei note obținute. Să se afișeze, printr-o funcție, informațiile citite. Să se scrie, în fișierul *lista_ordonata.txt*, o listă cu elevii participanți la concurs, ordonați alfabetic, notele și media obținută (funcție de ordonare, funcție de calcul a mediei). Să se afișeze lista elevilor înscriși la concurs, în ordinea descrescătoare a mediilor.
8. Se dă a , un șir de n numere naturale nenule. Numim o secvență specială, o secvență $a[i], \dots, a[j]$, unde $1 \leq i \leq j \leq n$, dacă cifra de control a sumei secvenței respective este 9. Să se determine numărul de secvențe speciale din a .
9. Se citește un număr natural n format din o cifră și apoi să realizeze următoarele operații, prin intermediul unor funcții:
- se construiește în memorie o matrice pătratică de dimensiune n , având forma indicată mai jos (pentru $n = 4$):

0	1	2	3
1	0	1	2
2	1	0	1
3	2	1	0
 - se scrie matricea, linie cu linie, elementele fiind separate prin spațiu, în fișierul text *matrice.txt*;
 - se citesc două numere i și j cuprinse între 0 și $n - 1$ și se interschimbă liniile i și j din matrice.