

Setul 5 – Funcții definite de către utilizator. Parametri transmiși prin valoare / prin referință.

☆ **În rezolvări NU se va folosi tipul tablou pentru parametrii funcției.** Pentru altceva decât parametri, se poate utiliza. (Relația Pointeri-Funcții-Tablouri face obiectul Setului 6 de probleme.)

☆ **Problemele 1, 5 și 6 se vor rezolva în timpul orelor de laborator.**

☆ Problemele 1-3 pot fi rezolvate folosind parametri transmiși prin valoare.

☆ Problemele 4-6 necesită și folosirea unor parametri transmiși prin referință. (Un exemplu similar a fost prezentat la curs)

☆ **Cu parametri transmiși prin referință se va lucra conform informațiilor prezentate la curs!!!**

1. Citiți de la tastatură un număr natural **n**. Scrieți o funcție care verifică dacă numărul este palindrom. Funcția va returna valoarea 1 dacă numărul este palindrom, respectiv 0 în caz contrar. Testați funcția în cadrul unui program C. (Un număr natural este palindrom dacă este egal cu numărul obținut prin scrierea cifrelor sale în ordine inversă. De exemplu, 15651 și 23532 sunt numere palindrom, în timp ce 15652 și 23534 nu sunt numere palindrom.)

2. Scrieți și testați o funcție C care verifică dacă un număr este perfect. Numărul perfect este un număr întreg egal cu suma divizorilor săi, din care se exclude numărul însuși. Ex. $6=1+2+3$; $28=1+2+4+7+14$

3. Să se scrie și să se testeze o funcție C care, analizând 3 valori numerice întregi, să verifice dacă acestea sunt ordonate crescător, descrescător, sunt identice sau sunt neordonate (ținând cont de ordinea în care au fost precizate). Rezultatul va fi furnizat ca valoare de tip caracter: 'C', 'D', 'I' sau 'N'. Se recomandă testarea funcției în cadrul unui program C care să permită execuția în buclă, pentru mai multe seturi de câte 3 valori introduse de la tastatură (similar cu un exemplu prezentat la curs).

Variantă de enunț: rezultatul va fi furnizat ca valoare de tip întreg, cu semnificație prestabilită.

4. Scrieți și testați o funcție C care să transforme 3 caractere (precizate ca parametri) în literele mari corespunzătoare, dacă acest lucru este posibil. Funcția va transmite programului apelant și informația că s-a putut sau nu realiza conversia. În caz de succes, programul apelant va afișa apoi pe ecran literele mari obținute de la funcție. În caz de eșec, va afișa un mesaj lămuritor.

Sugestie de rezolvare: puteți consulta exemplul prezentat la curs pt. operații de „maskare” cu operatori la nivel de bit; puteți elabora și altă rezolvare, bazată pe observații legate de ordinea caracterelor literă în setul ASCII.

Variantă de enunț: transformare în literele mici corespunzătoare.

5. Să se scrie o funcție C care să primească drept informație de prelucrat un număr întreg format din cel mult 3 cifre (respectarea condiției se verifică în `main()`, la introducerea valorii ce va fi folosită de către funcție). Se cere să se determine și să se transmită ca rezultate ce pot fi folosite după încheierea execuției funcției:

- numărul de cifre din care este format numărul;
- cifrele propriu-zise (evident, cel mult 3);

Să se utilizeze funcția într-un program C în care, după încheierea execuției funcției, să fie afișate din `main()` rezultatele furnizate de către aceasta.

6. Să se scrie o funcție C care să analizeze un număr întreg strict pozitiv și, dacă este posibil, să construiască un alt număr, format doar din cifrele pare ale celui inițial. Funcția va avea 2 parametri, care să reprezinte cele 2 numere, și va returna ca rezultat o valoare care să arate că noul număr a putut sau nu să fie construit. Să se testeze funcția într-un program C în care:

- să se citească de la tastatură numărul ce trebuie analizat de către funcție;
- să se apeleze funcția;
- să se testeze rezultatul returnat de către funcție (care arată că noul număr a putut sau nu să fie construit) și, în caz de succes, să se afișeze o informație de forma (de exemplu): „numărul obținut din 2567 este 26”