

Teme și aplicații

Formatori:

Tutor: [Stângaciu Valentin](#)  

Tutor: [Belu Claudiu-Marcel](#)  

+3

Data de începere a cursului:

 25.09.2023

 [Utilizatori înscrși](#)

 [Calendar](#)

 [Note](#)


 [Cursurile mele](#) [S1-L-AC-CTIRO1-PC](#) [Laborator 5: Operatori pe biți](#) [Teme și aplicații](#)

Teme și aplicații

- Efectuați următoarele adunări în baza 2 și verificați rezultatele: $11001011 + 00011111$; $11101100 + 10000000$
- Efectuați următoarele scăderi în baza 2 și verificați rezultatele: $11001001 - 00011111$; $11101100 - 10000000$
- Efectuați următoarele înmulțiri în baza 2 și verificați rezultatele: $1100 * 0110$; $0111 * 1011$
- Efectuați următoarea împărțire în baza 2 și verificați rezultatul: $1100 / 0110$
- (pe hârtie) Dacă $a=114$, $b=41$ și $c=158$ sunt numere **fără semn**, să se calculeze **pe foaie** rezultatul următoarelor operații: $\sim b$, $a \& (b \wedge c)$, $(a \& c) \& b$. Rezultatul se poate **verifica și pe calculator**, introducând numerele așa cum sunt date, în baza 10 și afișând rezultatul pe biți, cu funcția showBits
- (pe hârtie) Să se calculeze rezultatul expresiei: $01001011 << (01100101 >> 5)$
- Faceți următoarele transformări, pentru numere cu semn:
în baza 10: F_{16} , 76_8
în baza 2: -16_8 , -47_8 , $-9A_{16}$, $-2E_{16}$
în baza 16: 10010000_2 , 11001001_2 , 10000000_2
în baza 8: 10010000_2
- Se citește de la tastatură un număr fără semn. Să se seteze biții săi cu indicii 0,2,3, să se reseteze biții cu indicii 1,5,6 și să se complementeze biții cu indicii 4,7. Se vor afișa în binar atât numărul inițial cât și cel final.
- O variabilă de tip char ocupă în memorie 8 biți. Acești 8 biți pot fi împărțiți în 2 zone de câte 4 biți. În fiecare zonă de câte 4 biți s-ar putea memora câte un număr între 0 și 15. Scrieți un program care citește două numere între 1 și 10 și le memorează (împachetează) în cele două zone de câte 4 biți ale unei variabile de tip char. După memorare, programul va despacheta valorile memorate și le va afișa.
- Se citește o valoare de tip *int*. Se cere să se afișeze valoarea bitului de semn. Nu se cunoaște a priori dimensiunea tipului de date *int*.
- Se citește o valoare. Să se afișeze suma primilor săi 4 biți LSB (cei mai din dreapta)
- Scrieți (de mână pe hârtie) pentru câte o pereche de numere diferite pe 8 biți, cu valori între 32 și 255, reprezentarea lor pe biți, și
a) rezultatul operației ȘI pe biți și valoarea numerelor și rezultatului în baza 10
b) rezultatul operației SAU pe biți și valoarea numerelor și rezultatului în baza 8
c) rezultatul operației SAU EXCLUSIV pe biți și valoarea numerelor și rezultatului în baza 16.
- Scrieți o funcție care ia ca parametru un întreg fără semn și returnează numărul de biți de 1 din reprezentarea sa.
- Un nibble este un grup de 4 biți. Să se scrie un program care primește un număr fără semn n și această funcție
a) returnează valoarea acestui număr cu ordinea inversă a nibble-urilor
b) returnează valoarea acestui număr cu biții din fiecare nibble în ordine inversă
- Să se scrie o funcție care ia un număr fără semn n și returnează numărul de segmente de biți consecutivi egali. Exemplu: 000100 are 3 segmente: 000, 1, 00, deci programul va returna 3
- Scrieți și testați cu date de la intrare o funcție care ia ca parametri două numere întregi și returnează suma lor, folosind operatori pe biți. Calculați suma bit cu bit, cu propagarea transportului, cu algoritmul de adunare pe hartie pencil-and-paper, fara a folosi operatori aritmetici.
- Scrieți și testați cu date de la intrare o funcție care ia ca parametri două numere întregi și returnează diferența lor, folosind operatori pe biți. Calculați diferența bit cu bit, cu propagarea transportului, cu algoritmul de scădere pe hartie pencil-and-paper, fara a folosi operatori aritmetici.
- Să se scrie o funcție care primește ca argument un număr pe 32 de biți fără semn (nr), un număr pe 8 biți fără semn (n) și o poziție (p). Funcția va returna numărul (nr) în care octetul de pe poziția p va fi înlocuit cu numărul n . Octeții se numără se la dreapta (cel mai puțin semnificativ, poziția 0) spre stanga
exemplu: $nr = 0011\ 0110\ \mathbf{0101\ 1011}\ 0011\ 0110\ 0101\ 1011 = 911.947.355_{10}$, $n = 1110\ 1111 = 239_{10}$, $p = 2 \rightarrow$ rezultat: $0010\ 0110\ 1110\ 1111\ 0011\ 0110\ 0101\ 1011$
- Rescrieți funcția anterioară astfel încât numărul (nr) să fie pe o dimensiune de 64 de biți fără semn.

20. Scrieți o funcție care primește ca parametru 2 numere întregi pe 1 byte fără semn (n_1 și n_2) și returnează un număr întreg pe 2 bytes (n) format din cele două numere ca și parte cea mai semnificativă (n_2) și parte mai puțin semnificativă (n_1)
exemplu: $n_1 = 1100\ 1011$, $n_2 = 0110\ 0111 \rightarrow n = 0110\ 0111\ 1100\ 1011$
21. Scrieți o funcție care primește ca și argumente 4 numere întregi pe 1 byte fără semn (n_3 , n_2 , n_1 , n_0) și returnează un număr întreg pe 4 bytes (n) format din cele 4 numere unde n_3 este partea cea mai semnificativă și n_0 partea cea mai puțin semnificativă
exemplu: $n_3 = 1111\ 0000$, $n_2 = 0011\ 1100$, $n_1 = 1010\ 0011$, $n_0 = 0010\ 1101 \rightarrow n = 1111\ 0000\ 0011\ 1100\ 1010\ 0011\ 0010\ 1101$
22. Scrieți o funcție similară cu funcțiile realizate la aplicații anterioare în care se primesc ca argumente 8 numere pe 8 biți fără semn (n_7 , n_6 , ..., n_0) și returnează un număr de 64 de biți fără semn format din cele 8 numere primite ca și argumente în care n_7 devine cel mai semnificativ byte și n_0 cel mai puțin semnificativ byte
23. Rezolvați exercitiile din cartea BRIAN W KERNIGHAN DENNIS M. RITCHIE - The C Programming Language File, pag 46-47

[◀ Considerații teoretice](#)[Test grilă 1 ▶](#)[✉ Contactați serviciul de asistență](#)

Sunteți conectat în calitate de 
S1-L-AC-CTIRO1-PC

Meniul meu

[Profil](#)[Preferințe](#)[Calendar](#) [ZOOM](#)[Română \(ro\)](#)[English \(en\)](#)[Română \(ro\)](#)[Rezumatul păstrării datelor](#)[Politici utilizare site](#)