

FIȘA DE LABORATOR 6

Structuri de date alocate static. Lista, stiva și coada

1. Se citește de la tastatură două liste de elemente întregi $L1$ cu m elemente și $L2$ cu n elemente, fiecare element având cel mult șase cifre (m, n numere naturale de cel mult trei cifre). Să se scrie câte o funcție astfel:
 - a. citirea elementelor listei;
 - b. afișarea elementelor listei;
 - c. construirea listei reuniune pentru listele $L1$ și $L2$;
 - d. construirea listei intersecție pentru listele $L1$ și $L2$;
 - e. construirea listei diferență pentru listele $L1$ și $L2$;
 - f. verificarea incluziunii pentru listele $L1$ și $L2$;
 - g. testarea apartenenței unui element dat la lista $L1$.
2. Să se scrie un program care gestionează o stivă de cel mult 100 de numere naturale. Să se scrie câte o funcție pentru fiecare din următoarele operații cu stiva:
 - a. verificarea dacă o stivă este sau nu este vidă;
 - b. verificarea dacă o stivă este sau nu este plină;
 - c. inițializare stivă;
 - d. determinarea elementului din vârful stivei;
 - e. adăugarea pe stivă a unui element dat;
 - f. eliminarea unui element de pe stivă.
3. Se citește de la tastatură un număr natural n de cel mult nouă cifre. Să se afișeze pe ecran numărul dat în baza 2.
4. Se dau n șiruri de paranteze rotunde, fiecare șir de paranteze se încheie la întâlnirea caracterului *punct*. Fișierul `paranteze.in` conține pe prima linie un număr natural n , iar pe următoarele n linii cele n șiruri de paranteze, fiecare șir având parantezele separate prin câte un spațiu, iar la finalul unui șir se întâlnește caracterul *punct*. Pentru fiecare șir de paranteze să se afișeze pe ecran, separate prin câte un spațiu, 1 sau 0 după cum șirul este corect respective incorect parantezat (parantezele se închid corect).
5. Să se scrie un program care gestionează o coadă de cel mult 100 de numere naturale. Să se scrie câte o funcție pentru fiecare din următoarele operații cu coada:
 - a. verificarea faptului că o coadă este sau nu este vidă;
 - b. verificarea faptului că o coadă este sau nu este plină;
 - c. inițializare coadă;
 - d. determinarea elementului curent din coadă;
 - e. adăugarea în coadă a unui element dat;
 - f. eliminarea unui element din coadă.
6. Într-un labirint se află un șoricel și o bucată de cașcaval. Șoricelul dorește să ajungă la cașcaval efectuând un număr minim de pași. La un pas șoricelul se poate deplasa în una dintre pozițiile învecinate (sus, jos, stânga, dreapta), evident dacă acolo este culoar de trecere. Determinați numărul minim de poziții pe care șoricelul trebuie să le parcurgă pentru a ajunge la cașcaval. Se consideră o matrice cu m linii și n coloane ($1 < m < 50$, $1 < n < 50$), ce memorează valori de 0 și 1, precum și două puncte A și B din matrice. Scrieți un program care citește de la tastatură m, n , elementele matricei și coordonatele a două puncte din matrice (date prin indicii liniei respectiv coloanei pe care se află) și care afișează pe ecran numărul minim de pași prin care se poate ajunge de la punctul A la punctul B sau se afișează -1 dacă nu se poate ajunge de la punctul A la punctul B .