

1. Concepeți și implementați un algoritm care, având două liste circulare distincte, accesibile prin pointerii x și t, inserează elementele din a doua listă (al cărei pointer este t) la începutul primei liste (al cărei pointer este x).

2. Concepeți și implementați un algoritm care mută elementul maxim dintr-o listă înlănțuită pe ultima poziție din listă.

3. Concepeți și implementați un algoritm care mută elementul minim dintr-o listă înlănțuită pe prima poziție din listă.

4. Concepeți și implementați un algoritm care să transforme o listă înlănțuită în inversa sa fără a folosi memorie suplimentară.

(exemplu: (1 2 3 4 5) -> (5 4 3 2 1))

5. Concepeți și implementați un algoritm care, folosind o coadă cu două capete ale cărei operații sunt executate cu complexitatea de timp $O(1)$, rezolvă următoarea problemă: Având o clasă cu un număr egal de băieți și fete să se genereze corect perechi băiat-fată astfel încât fiecare băiat va avea ca pereche o fată din clasă.

6. Concepeți și implementați un algoritm care să rearanjeze elementele unei liste înlănțuite astfel încât toate elementele de pe poziții pare din listă să apară după elementele situate inițial pe poziții impare, ordinea elementelor rămânând însă neschimbată

(exemplu: (1 5 2 1 4 1 7 2 3 4) -> 5 1 1 2 4 1 2 4 7 3)

7. Concepeți și implementați un algoritm care folosind o listă înlănțuită de caractere pentru a reprezenta un string, verifică dacă acesta este un palindrom (se citește la fle și de la dreapta la stânga și de la stânga la dreapta) (se omit caractere albe – spațiu, tab, etc.)

(exemplu: *if i had a hifi* -> palindrom)

8. Concepeți și implementați un algoritm care folosind o stivă transcrie o expresie matematică din forma postfixată în forma infixă.

(exemplu: 5 9 8 + 4 6 * * 7 + * -> (5*(((9+8)*(4*6))+7)))

9. Concepeți și implementați un algoritm care folosind o stivă transcrie o expresie matematică din forma infixă în forma postfixată.

(exemplu: (5*(((9+8)*(4*6))+7)) -> 5 9 8 + 4 6 * * 7 + *)

10. Concepeți și implementați un algoritm care folosind o stivă evaluează o expresie matematică scrisă în forma postfixată.

(exemplu: $5 \cdot 9 + 8 + 4 \cdot 6 * * 7 + * \rightarrow (5 * (((9+8) * (4 * 6)) + 7))$)

11. Concepeți și implementați un algoritm care înversează o listă înlănțuită folosind o stivă.

(exemplu: 1 2 3 4 5 6 7 \rightarrow 7 6 5 4 3 2 1)

12. Concepeți și implementați un algoritm care folosind o coadă cu extragere aleatoare generează numerele câștigătoare la loto astfel: se generează numerele de la 1 la 99 și se poziționează pe poziții aleatoare în coadă și se extrag aleator cele cinci numere loto.

13. Concepeți și implementați tipul abstract de date Coadă cu extragere aleatoare astfel încât operațiile caracteristice put(inserează) și get(extrage) vor avea complexitatea de timp $O(1)$.

14. Concepeți și implementați exemplul de alegeri de tip Josephus (problema prezentată la seminar) folosind o listă înlănțuită circulară.

15. Concepeți o structură de date de tip stivă care să implementeze toate operațiile specifice (push , pop) cu complexitatea de timp $O(1)$ și în plus să poată returna elementul cu valoare maximă din stivă cu aceeași complexitate de timp $O(1)$.

16. Concepeți o structură de date de tip stivă care să implementeze toate operațiile specifice (push , pop) cu complexitatea de timp $O(1)$ și în plus să poată returna elementul cu valoare minimă din stivă cu aceeași complexitate de timp $O(1)$.

17. Concepeți și implementați un algoritm care, folosind o stivă, efectuează transformarea unui număr din baza 10 într-o bază de numerație dată ca parametru.