Lucrări practice la cursul
Structuri de Date și Algoritmi
Lucrare practică nr. 3

[Task I]

[Task II]

Tema: Liste înlănțuite și stive

Obiective:

- Înțelegerea și implementarea listelor înlănțuite;
- Aplicarea conceptelor de listă în gestionarea unei colecții de obiecte;
- Înțelegerea și implementarea stivelor;
- Utilizarea stivelor în implementarea unui calculator pentru expresii matematice.

Task I: Liste înlanțuite

O listă înlanțuită este o colecție ordonată de elemente, în care fiecare element are o referință (pointer) către elementul următor.

Implementați o listă înlănțuită
Definiția nodului pentru listă înlănțuită este:

```
typedef struct Node {
    int data;
    struct Node* link;
} Node;
```

Exerciții comune:
a. Implementați o funcție care verifică dacă lista este goală.
b. Implementați o funcție de inserare a unui nod la începutul listei.
c. Implementați o funcție de ștergere a unui nod după o anumită valoare.
d. Implementați o funcție de afișare a listei.
e. Implementați o funcție de modificare a informației dintr-un nod.

Exerciții pe variante:

Varianta 1
Lista de cărți
Structura unui nod:

```
typedef struct Carte {
    char titlu[100];
    char autor[50];
    int an_publicare;
```

```
    struct Carte* next;
} Carte;
```

Trebuie să avem posibilitatea de a face:

1. Inserarea unei cărți la sfârșitul listei.
2. Ștergerea unei cărți după titlu.
3. Afișarea tuturor cărților scrise de un anumit autor.
4. Actualizarea anului de publicare pentru o carte dată.

Varianta 2
Lista de email-uri
Structura unui nod:

```
typedef struct Email {
    char expeditor[100];
    char destinatar[100];
    char subiect[150];
    char continut[1000];
    struct Email* next;
} Email;
```

1. Adăugarea unui nou email în listă.
2. Ștergerea unui email după expeditor.
3. Afișarea tuturor email-urilor primite de la un anumit expeditor.
4. (Avansat) Căutarea unui email după un cuvânt cheie în conținut.
5. (Avansat) Sortarea email-urilor după dată, în ordine cronologică.

Varianta 3
Lista de operațiuni bancare
Structura unui nod:

```
typedef struct OperatiuneBancara {
    char codBanca[6];
    char codClient[11];
    char dataOperatiune[11];
    double sumaOperatiune;
    struct OperatiuneBancara* next;
} OperatiuneBancara;
```

Exerciții:

1. Adăugarea unei noi operațiuni în listă.
2. Ștergerea ultimei operațiuni.
3. Afișarea operațiunilor unui client specific.
4. (Avansat) Calculul sumei totale a operațiunilor unui client într-o perioadă specificată.
5. (Avansat) Sortarea operațiunilor descrescător după sumă.

Task II: Stive

O stivă este o colecție de elemente, cu principiul de funcționare LIFO (last-in, first-out). Operațiile principale asupra stivei sunt: push (inserare) și pop (extragere).

Definiția stivei:

```
typedef struct Stiva {
    int top;
    unsigned capacitate;
    int* array;
} Stiva;
```

Implementați un calculator pentru expresii matematice
Folosind două stive, una pentru valori și una pentru operatori, realizați un calculator care să evalueze o expresie matematică dată de utilizator.
Tratați cazurile de operatori: +, -, *, / și (,).
Implementați funcții de push, pop, isEmpty și de evaluare a operatorilor pentru a evalua expresia.