DATA 18.10.2022 ALGORITMI SI STRUCTURI DE DATE

LABORATOR

PROBLEMA:

1. Sa se scrie un program C++ care implementeaza o lista de intregi, alocata inlantuit. Operatiile de baza se vor face prin intermediul functiilor.
   HEAD, pointer la primul element al listei, se va declara global
   La rulare trebuie sa se poata alege dintr-un meniu una din operatiile:
   1. Inserarea unui element la inceputul listei
   2. Inserarea unui element la sfarsitul listei
   3. Inserarea unui element dupa un element dat
   4. Accesarea unui element
   5. Afisare lista – afisarea elementelor listei

   Orice alt input---exit iesirea din program

```
#include <iostream>
#include <unistd.h>


using namespace std;


struct Nod{
   int info;
   Nod *link;
};


Nod *Head=NULL;


int a1,a2,a3,a4,a5,a6,a9,b;


void printList(){
   Nod *iter = new Nod;
   iter = Head;
```

LABORATOR

```
    while(iter!=NULL && iter) {
        cout << iter->info << " --> " ;
        iter = iter->link;
    }
    cout << "NULL" << endl;
}

void editElement(int a, int b){
    Nod *iter = Head;
    if (iter!=NULL && iter->info!=a){
        iter = iter->link;
    }
    if(iter!=NULL && iter->info==a){
        cout << "Elementul " << a << " a fost gasit!" << endl;
        iter->info = b;
    } else {
        cout << "Elementul " << a << " nu a fost gasit!" << endl;
    }
    cout << "----" << endl;
    printList();
}

void insertElement_Start(int a){
    Nod *p = new Nod;
    if (p==NULL){
```

LABORATOR

```
        cout << "OVERFLOW";
        return;
    }
    p->info = a;
    p->link = Head;
    Head = p;
    cout << "----" << endl;
}


void insertElement_Last(int a){
    Nod *p = new Nod;
    if (p==NULL){
        cout << "OVERFLOW";
        return;
    }
    p->info = a;
    p->link = NULL;
    Nod *iter = Head;
    while(iter && iter->link) iter = iter->link;
    if(iter){
        iter->link = p;
    } else {
        Head=p;
    }
    cout << "----" << endl;
```

```
}

void stergereStart(){
   Nod *temp = Head;
   Head = Head->link;
   delete temp;
   cout << "----" << endl;
}

void stergereLast(){
   Nod *iter = Head;
   while (iter && iter->link->link!=NULL && iter->link!=NULL){
      iter=iter->link;
   }
   if(iter->link!=NULL){
      delete iter->link;
      iter->link=NULL;
   } else {
      delete iter;
      Head=NULL;
   }
   cout << "----" << endl;
}

void stergere_After(Nod *q){
```

LABORATOR

```
    Nod *iter = Head;
    while(iter && iter->link!=q){
        iter=iter->link;
    }
    if(iter==NULL){
        cout << "Nu exista elementul q!" << endl;
        return;
    }
    iter->link=q->link;
    delete q;
    cout << "----" << endl;
}

void insertElement_After(int a, Nod *q){
    Nod *p = new Nod;
    if(q==NULL){
        cout << "Elementul dupa nu exista" << endl;
        return;
    }
    if (p==NULL){
        cout << "OVERFLOW";
        return;
    }
    p->info = a;
    p->link = q->link;
```

LABORATOR

```
    q->link = p;
    cout << "----" << endl;
}


Nod* accessElement(int a){
    Nod *iter = Head;
    if (iter->info!=a && iter!=NULL) {
        iter = iter->link;
    }
    if(iter!=NULL && iter->info==a){
        cout << "Elementul " << a << " a fost gasit!" << endl;
    } else {
        cout << "Elementul " << a << " nu a fost gasit!" << endl;
    }
    cout << "----" << endl;
    return iter;
}


int main() {
    int ans;
    while(1){
        reswitch:
        cout << "Alegeti din urmatoarele:" << endl << "1. Insert element from start - 2. Insert element from last - 3. Insert element AFter... - 4. Access element - 5. Edit element - 6. Print List - 7. Delete First Element - 8. Delete Last Element - 9. Delete Element After ... - 0. Exit: ";
        cin >> ans;
```

LABORATOR

```
switch (ans) {
    case 0:
        cout << "Exiting the program..." << endl;
        sleep(3);


        return 0;
    case 1:
        cout << "You chose 'Insert element from start'..." << endl;
        cout << "Insert element: ";
        cin >> a1;
        insertElement_Start(a1);
        break;
    case 2:
        cout << "You chose 'Insert element from last'..." << endl;
        cout << "Insert element: ";
        cin >> a2;
        insertElement_Last(a2);
        break;
    case 3:
        cout << "You chose 'Insert element after'..." << endl;
        cout << "Insert element: ";
        cin >> a5;
        cout << "Access element: ";
        cin >> a6;
        insertElement_After(a5, accessElement(a6));
        break;
    case 4:
        cout << "You chose 'Access element'..." << endl;
        cout << "Insert element: ";
        cin >> a3;
        accessElement(a3);
        break;
    case 5:
        cout << "You chose 'Edit element'..." << endl;
        cout << "Insert element to change: ";
        cin >> a4;
        cout << "Insert element to insert: ";
        cin >> b;
        editElement(a4,b);
        break;
    case 6:
        cout << "You chose 'Print list'..." << endl;
        cout << "----------------------------------------" << endl;
        printList();
        break;
    case 7:
        cout << "You chose 'Delete first element'..." << endl;
        cout << "----------------------------------------" << endl;
        stergereStart();
        break;
```

```
        case 8:
            cout << "You chose 'Delete last element'..." << endl;
            cout << "----------------------------------------" << endl;
            stergereLast();
            break;
        case 9:
            cout << "Insert element: ";
            cin >> a9;
            stergere_After(accessElement(a9));
            break;
        default:
            cerr << "You inserted an invalid answer." << endl;
            return 69;
        }
        goto reswitch;
    }

}
```