



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
"МИРЭА - Российский технологический университет"  
**РТУ МИРЭА**

---

Институт информационных технологий  
Кафедра математического обеспечения и стандартизации  
информационных технологий

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2**  
**Тема.** Линейные списки и способы реализации.  
Линейный однонаправленный список.  
**по дисциплине**  
«Дисциплина Структуры и алгоритмы обработки данных»

Выполнил студент группы ИНБО-15-20

Ло Ван Хунг

Принял ассистент кафедры МОСИТ

Коваленко М.А.

Практическая  
работа выполнена

«\_\_»\_\_\_\_\_2021 г.

«Зачтено»

«\_\_»\_\_\_\_\_2021 г.

Москва 2021

## 1 Задание

Имеется линейный однонаправленный список L. Реализовать в виде функции следующие операции над списком:

- 1) Является ли список пустым? Список пуст, если указатель содержит NULL.
- 2) Создание списка из n узлов, включая новые узлы в начало списка
- 3) Находит среднее арифметическое всех значений непустого списка L.
- 4) Заменяет в списке L все значения равные x на значения y.
- 5) Меняет местами первый и последний элементы.

Требования к выполнению

1. Подготовить тестовый пример списка: изобразить список в тетради.
2. Для каждой операции
  - a. Определить структуру узла списка, в котором информационная часть узла типа `int`
  - b. Определить прототипы функций
  - c. изобразить процесс выполнения действий с элементами списка на исходном списке и показать блок операторов, которые выполняют операцию
  - d. реализовать алгоритм операции e. выполнить тестирование.

## 2 Отчет по заданию

### 1) Код определений функций (файл с расширением cpp)

```
#include<iostream>
#include <math.h>
#include <stdlib.h>
using namespace std;
struct node
{
    float data;
    node*next;
};
struct list
{
    node *head;
    node *tail;
};

node *getnode(float x)
{
    node*t;
    t=new node;
    if(t==NULL)
    {
        cout<<"empty list.";
        //exit(0);
    }
    t->data=x;
    t->next=NULL;
    //cout<<"danh sach rong.";
    return t;
}
void addhead(list &l,node*p)
{
    if(l.head==NULL)
        l.head=l.tail=p;
    else
    {
        p->next=l.head;
        l.head=p;
    }
}
void addtail(list &l,node*p)
{
    if(l.head==NULL)
        l.head=l.tail=p;
    else
    {
        l.tail->next=p;
        l.tail=p;
    }
}
void input(list &l)
{
    node *p;
    int n;
    float x;
    l.head=l.tail=NULL; //khởi động ds rong
    cout<<"enter nodes: ";
    cin>>n;
    if(n==0)
    {
        cout<<" enter n>0.";
        return;
    }
    else
    for(int i=0;i<n;i++)
    {
        cout<<"enter x"<<"["<<i+1<<"]"<<"<<";
        cin>>x;
        p=getnode(x);
        addtail(l,p); //addhead(l,p);
    }
}
```

```

void output(list &l)
{
    node*i;
    if(i==NULL)
    {
        cout<<"empty list.";
        cout<<"\nNULL";
    }
    else
    {
        for(i=l.head;i!=NULL;i=i->next)
            cout<<i->data<<" ";
        //cout<<"NULL";
    }
}

int calcAverageList(list &l)
{
    node *p = l.head;
    if(!p)
        return 0;
    int i=0;
    float sum=0;
    while(p)
    {
        sum += p->data;
        i++;
        p=p->next;
    }
    cout<<"\nmedium score : "<<sum/i;
    return sum/i;
}

node *convert(list &l, int a, int b){
    node *p = l.head;
    while (p != NULL){
        if (p->data == a){
            p->data = b;
        }
        p = p->next;
    }
    return l.head;
}

void swapNode(list &l)
{
    node *p = l.head;
    node *temp = NULL;
    node *index = NULL;
    while(p->next!=NULL)
    {
        index=p;
        p=p->next;
    }
    if(l.head == p)
    {
        return ;
    }
    else if(l.head->next == p)
    {
        temp = l.head;
        l.head = p;
        l.head->next = temp;
        temp->next = NULL;
    }
    else
    {
        temp = l.head;
        l.head = p;
        l.head->next=temp->next;
        index->next=temp;
        temp->next= NULL;
    }
}

```

## 2) Код основной программы

```
int main()
{
    list l;
    node *p=l.head;
    input(l);
    output(l);
    calcAverageList(l);
    int a,b;
    cout<<"\nenter a :";
    cin>>a;
    cout<<"enter b :";
    cin>>b;
    l.head = convert(l, a, b);
    cout<<"new list: \n";
    output(l);
    swapNode(l);
    cout<<"\nnew list: "<<endl;
    output(l);
}
```

3) Тест для каждой операции  
Название операции

Номер теста	Входные данные	Ожидаемый результат
1 - Список пуст, если указатель содержит NULL.		<pre>bool isEmpty(Node *head) {     if (head == NULL)     {         cout &lt;&lt; "List is empty."         &lt;&lt; endl;         return true;     }     else     {         return false;     } }</pre>
2- Заполнение массива: с клавиатуры и вывод массива	2 6 3 2 1 4 6 2	<pre>x[1] = 3 x[2] = 2 x[3] = 1 x[4] = 4 x[5] = 6 x[6] = 2</pre>
3- Находит среднее арифметическое всех значений непустого списка L.	3	medium score: 3
4 - Заменяет в списке L все значения равные x на значения y.	4 2 8	<pre>x[1] = 3 x[2] = 8 x[3] = 1 x[4] = 4 x[5] = 6 x[6] = 8</pre>
5- Меняет местами первый и последний элементы.	5	<pre>x[1] = 8 x[2] = 8 x[3] = 1 x[4] = 4 x[5] = 6 x[6] = 3</pre>

#### 4) Скриншот выполнения теста.

Выбрать C:\Users\IVC1-5\Desktop\Алгоритм 2.exe

```
enter nodes: 6
enter x[1]:3
enter x[2]:2
enter x[3]:1
enter x[4]:4
enter x[5]:6
enter x[6]:2
3 2 1 4 6 2
medium score : 3
enter a :2
enter b :8
new list:
3 8 1 4 6 8
new list:
8 8 1 4 6 3
-----
Process exited after 30.25 seconds with return value 0
Для продолжения нажмите любую клавишу . . .
```