МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Отчет по выполнению практического задания 5

Тема: Однонаправленный динамический список.

Дисциплина Структуры и алгоритмы обработки данных

Выполнил студент Зубов Н. А.

Фамилия И.О.

группа ИКБО-24-20

Москва 2021

Цель работы

Получить знания и практические навыки управления динамическим однонаправленным списком.

Ответы на вопросы

1. **Расскажите о трех уровнях представления данных в программной системе.** Внешний или пользовательский, промежуточный или логический, внутренний или физический. Внешний уровень определяет наиболее полезную и удобную для конкретного пользователя форму представления подмножества данных, необходимых для выполнения стоящих перед пользователем задач. На промежуточном уровне основной акцент делается на семантике, т.е. смысловом значении данных, этот уровень содержит в себе логическое представление совокупности элементов данных с отображением связей между ними в программе. На физическом уровне описывается физическая реализация данных – это биты, байты, сведения о размещении записей в файлах и таблицах, сведения о сжатии данных, выбранных методах их шифрования и т.д.
2. **Что определяет тип данных?** Тип данных определяет возможные значения и их смысл, операции, а также способы хранения значений типа.
3. **Что определяет структура данных?** Структура данных определяет способ хранения данных в компьютере.
4. **Расскажите о структуры хранения данных в компьютерных технологиях.** Существуют два вида структур хранения данных: вектор и списковые структуры. Вектор – структура хранения многоэлементных структур данных. Вектор представляет линейную структуру, состоящую из нескольких последовательно расположенных ячеек. Ячейка предназначена только для хранения значения и связи со следующей за ней ячейкой она не хранит. Для отображения структуры данных в виде вектора в языках программирования используется тип данных – массив. Списковые структуры предназначены, так же, как и вектор, для хранения многоэлементных структур данных. Но в отличие от вектора, ячейка списка хранит значение элемента структуры данных и связь (ссылку) с другими ячейками списка. В зависимости от количества связей в элементе списка различают: однонаправленные (или односвязные) и двунаправленные (двухсвязные) списки. При организации таких списковых структур в языках программирования используются указатели.

5. **Дайте определение линейной структуре данных.** Линейная структура данных – это упорядоченная структура, в которой адрес каждого элемента однозначно определяется его номером (индексом).
6. **Дайте определение структуре данных линейный список.** Линейный список – это структура данных, каждый элемент которой посредством указателя связывается со следующим элементом.
7. **Дайте определение структуре данных стек.** Стек – это структура данных, представляющий собой список элементов, организованных по принципу LIFO.
8. **Дайте определение структуре данных очередь.** Очередь – это структура данных, представляющий собой список элементов, организованных по принципу FIFO.
9. **Чем стек отличается от структуры данных линейный список?** Стек заточен исключительно под задачу FIFO (первым пришел – первым ушел), а линейный список, в свою очередь, является лишь просто списком.
10. **Какой из видов линейных списков лучше использовать, если нужно введенную последовательность вывести наоборот?** В таком случае лучше использовать односвязный линейный список.
11. **Определите сложность алгоритма операции вставки элемента в i-тую позицию: а) массива; б) линейного списка.** а) O(n); б) O(l)
12. **Определите сложность алгоритма операции удаления элемента из i-той позиции: а) массива; б) линейного списка.** а) O(n); б) O(l)
13. **В чем суть трюка Вирта при выполнении операции удаления элемента из списка?** В однонаправленном списке при выполнении операции вставки нового узла new перед узлом next в заданной позиции pos возникает сложность: нужна ссылка на узел, предшествующий узлу next. Но можно обойти эту сложность вставить узел new после узла next, а затем поменять значения информационных частей узлов new и next.
14. **Определите структуру узла однонаправленного списка.**

```
struct unit
{
    int index;
    double degree;
    unit* next;
};
```

Рис. 1 Структура узла однонаправленного списка

15. **Реализуйте алгоритм вывода линейного однонаправленного списка.**

```
void printList(unit* head)
{
    unit* current = head;
    while (current != NULL)
    {
        cout << current->info << " ";
        current = current->next;
    }
    cout << endl;
}
```

Рис. 2 Алгоритм вывода линейного однонаправленного списка

16. Приведите фрагмент кода программы на языке C++ выполнения операции перемещения последнего элемента в начало списка.

```
void moveLastToFirst(unit*& L)
{
    unit* elem = L;
    int i = 0;
    while (elem->next != 0)
    {
        elem = elem->next;
        i++;
    }
    getNode(L, i - 1)->next = NULL;
    elem->next = L:
    L = elem;
}
```

Рис. 3 Операция перемещения

17. Какое из действий лишнее в следующем фрагменте кода? Куда вставляется новый узел?

```
struct Node {
int info;
Node* next;
};
typedef Node* List;
List L = new List;
void insertToList(List LL, int x) {
    List q = new Node; q->info = x; q->next = 0;
    if (LL == nullptr) LL->next = q;
    else
        q->next = LL->next;
    LL->next = q;
}
```

Рис. 4 Исходный код

Если в LL находится нулевой указатель, то LL->next – ошибка.

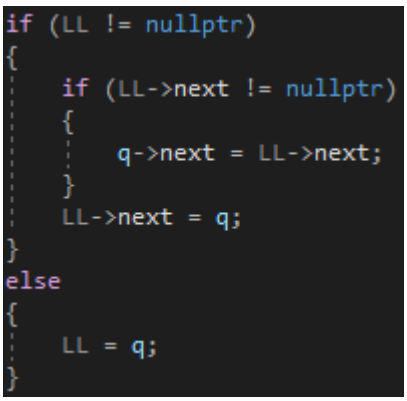


Рис. 5 Новый фрагмент кода

Отчёт по заданию 1

Условие задания

Реализуйте программу решения задачи варианта по использованию линейного однонаправленного списка. Разработать функцию для создания исходного списка, используя функцию вставки нового узла перед первым узлом. Разработать функцию вывода списка. Разработать функции дополнительного задания варианта. При необходимости можно добавлять функции, декомпозируя задачу. В основной программе выполните только тестирование каждой функции. Меню можно не создавать. Тесты обязательны.

Линейный многочлен n-ой степени представлен в программе как линейный однонаправленный список. Каждый i-ый узел списка хранит информацию по i-му члену многочлена. Поэтому информационная часть узла состоит из двух значений: степень члена и коэффициент при этой степени. Если i-ый член в многочлене отсутствует, то узел не создается. Разработать функцию, которая создает список по переданному в качестве параметра многочлену: он представлен массивом коэффициентов и их степеней. Разработать функцию, которая выводит многочлен и представляет его в форме выражения. Разработать функцию, которая вычисляет значение многочлена при заданном значении x. В вычислении использовать алгоритм Горнера.

Выполнение задания

Код программы

```
#include <iostream>
#include <cmath>
using namespace std;

struct unit
{
	int index;
	double degree;
	unit* next;
```

```
};

int HornerAlgh(unit* L, int x)
{
	unit* node = L;
	int result(0);
	while (node != 0) {
		result += node->index * pow(x, node->degree);
		node = node->next;
	}
	return result;
}

void outPolynomial(unit* L)
{
	unit* node = L;
	cout << "Многочлен: ";
	while (node != 0) {
		if (node->index > 0 and node->degree > 0) {
			cout << node->index << "*x^" << node->degree << " + ";
		}
		node = node->next;
	}
	cout << " \b\b\b " << endl;
}

void createList(unit*& L, int n)
{
	unit* q, * q1 = NULL;
	cout << "Введите " << n * 2 << " чисел в формате ИНДЕКС пробел СТЕПЕНЬ" << endl;
		for (int i = 1; i <= n; i++) {
			q = new unit;
			cin >> q->index >> q->degree;
			q->next = NULL;
			if (L == NULL) {
				L = q;
				q1 = L;
			}
			else {
				q1->next = q;
				q1 = q;
			}
		}
}

int main(int argc, const char* argv[])
{
	setlocale(LC_ALL, "ru");
	unit* HeadList = NULL, * ptr;
	int n, x;
	cout << "n = ";
	cin >> n;
	createList(HeadList, n);
	outPolynomial(HeadList);
	cout << "x = ";
	cin >> x;
	cout << "Результат: " << HornerAlgh(HeadList, x);
	return 0;
}
```

Анализ алгоритма

Структура узла однонаправленного списка:

```
struct unit
{
    int index;
    double degree;
    unit* next;
};
```

Рис. 6 Структура узла

Вывод многочлена и представление его в форме выражения:

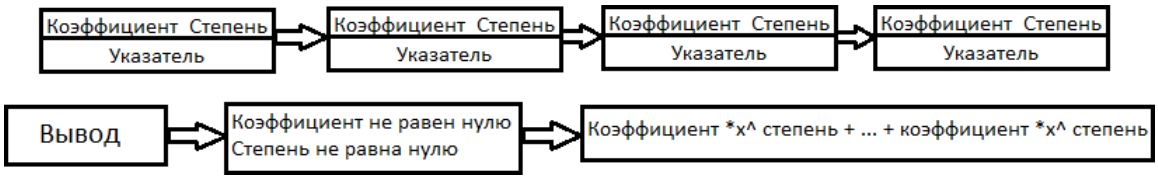


Рис. 7

Вычисление значения многочлена при заданном X:

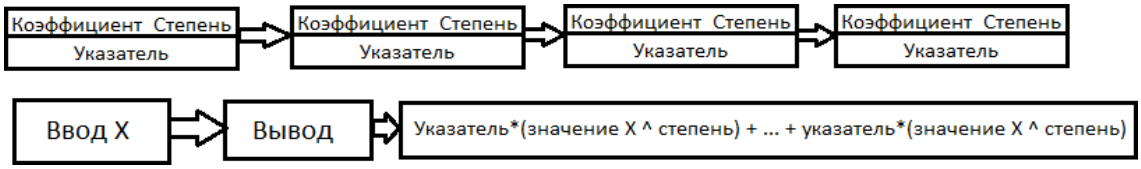
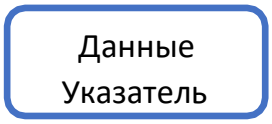


Рис. 8

Изображение структуры данных:

Узел:



В коде помимо основной функции main были также использованы вспомогательные функции outPolynomial, HornerAlgh, createList.

Таблицы тестов

Таблица 1. Таблица тестирования каждой операции

Разработать функцию, которая выводит многочлен и представляет его в форме выражения.	n = 4 Введите 8 чисел в формате ИНДЕКС пробел СТЕПЕНЬ 1 2 4 5 2 3 1 9 Многочлен: 1*x^2 + 4*x^5 + 2*x^3 + 1*x^9
Разработать функцию, которая вычисляет значение многочлена при заданном значении x.	Многочлен: 1*x^2 + 4*x^5 + 2*x^3 + 1*x^9 x = 3 Результат: 20718

Таблица 2. Таблица тестирования алгоритма

Ввод	Вывод
2 1 2 3 4 5	n = 2 Введите 4 чисел в формате ИНДЕКС пробел СТЕПЕНЬ 1 2 3 4 Многочлен: 1*x^2 + 3*x^4 x = 5 Результат: 1900
5 3 2 0 1 2 4 9 1 7 0 3	n = 5 Введите 10 чисел в формате ИНДЕКС пробел СТЕПЕНЬ 3 2 0 1 2 4 9 1 7 0 Многочлен: 3*x^2 + 2*x^4 + 9*x^1 x = 3 Результат: 223
3 1 0 2 2 0 98 40	n = 3 Введите 6 чисел в формате ИНДЕКС пробел СТЕПЕНЬ 1 0 2 2 0 98 Многочлен: 2*x^2 x = 40 Результат: 3201

Вывод

В результате выполнения данной практической работы я получил практические навыки управления динамическим однонаправленным списком и научился работать с ним.