

Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»



ОТЧЁТ

О выполнении лабораторной работы № 6

"Работа со структурами данных на основе списков"

Студент: Ким В.А.

Группа: Б22-703

Преподаватель: Овчаренко Е.С.

Москва - 2022

1. Формулировка индивидуального задания

Вариант №15

Задание

Осуществить циклический сдвиг на N символов влево в каждом слове строки.

2. Описание использованных типов данных

В данной работе использовались тип данных «int» для работы с целыми числами, char* для работы со строками, структура List

```
struct Node {  
    Node* next;  
    CLASS data;  
};
```

3. Описание использованного алгоритма



Рис. 1: Блок-схема алгоритма работы функции main

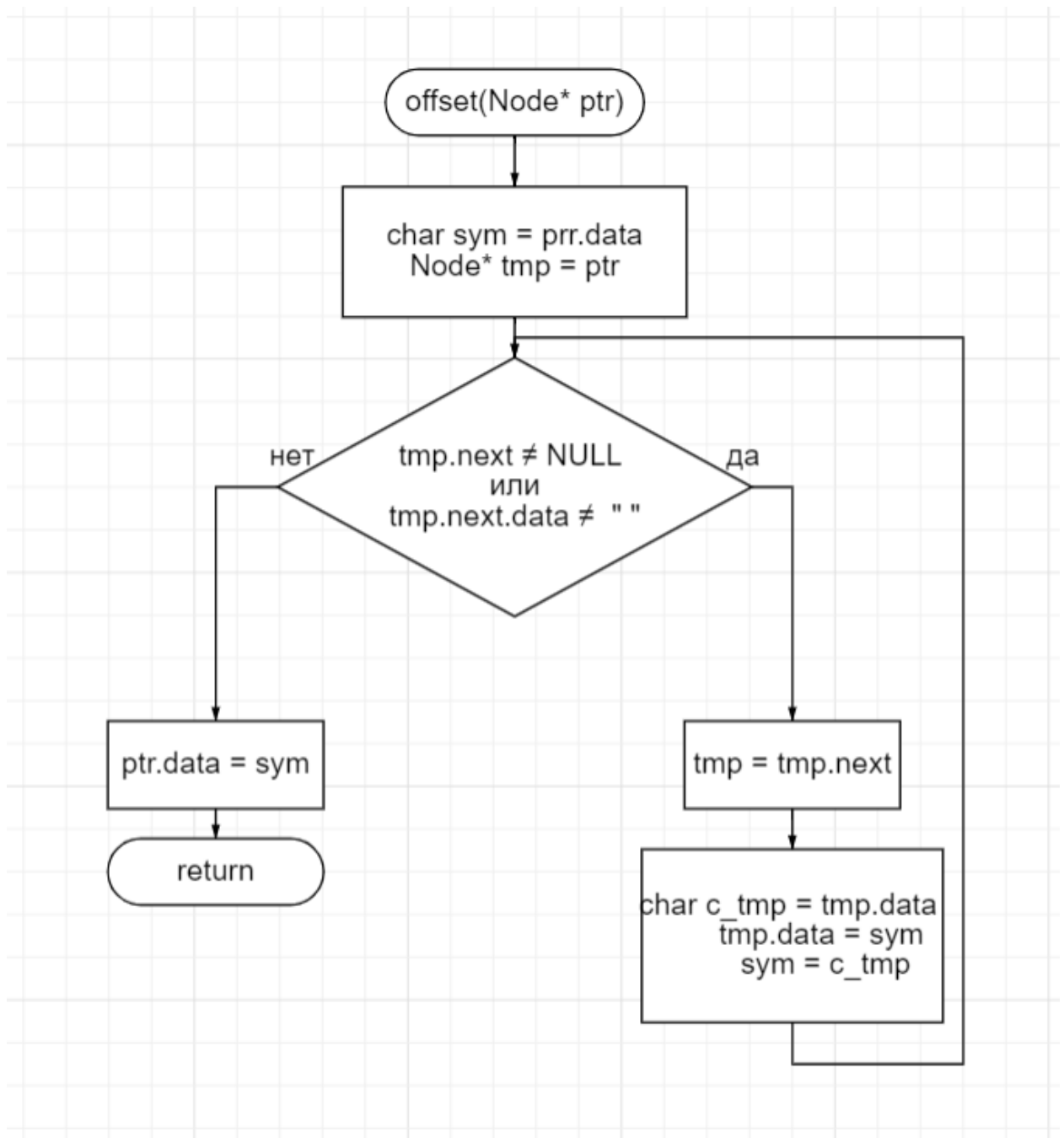


Рис. 2: Блок-схема алгоритма работы функции offset

4. Исходные коды разработанных программ

Листинг 1: Исходные код программы файл: lab_06.c

```

1  #include <stdio.h>
2  #include <locale.h>

3  #include "lib_list(process).h"
4  #include "../codes/lib/lib_scanf.h"
  
```

```

5 void offset(Node* ptr) {
6 char sym = ptr->data;
7 Node* tmp = ptr;

8 while (tmp->next != NULL) {
9 if (IS_SPACE(tmp->next->data)) break;
10 tmp = tmp->next;

11 char c_tmp = tmp->data;
12 tmp->data = sym;
13 sym = c_tmp;
14 }
15 ptr->data = sym;
16 }

17 void get_string(List* ls) {
18 printf("str > ");
19 do {
20 char a = getchar();
21 if (a == '\n') break;

22 list_push_back(ls, a);
23 } while (1);
24 }

25 void del_spaces(List* ls) {
26 Node* ptr = ls->first;

27 while (ptr->next != NULL) {
28 if (ptr == ls->first && IS_SPACE(ptr->data)) {
29 Node* temp = ptr;
30 ptr = ls->first = ptr->next;
31 free(temp);
32 }
33 else if (IS_SPACE(ptr->data) && IS_SPACE(ptr->next->data)) {
34 Node* temp = ptr->next;
35 ptr->next = ptr->next->next;
36 free(temp);
37 }
38 else ptr = ptr->next;
39 }
40 if (IS_SPACE(ptr->data)) {
41 Node* temp = ls->first;
42 while (temp->next->next != NULL) temp = temp->next;

43 free(ptr);
44 temp->next = NULL;
45 ls->last = temp;
46 }

47 ls->size = list_get_size(ls);
48 }

49 int main() {
50 setlocale(LC_ALL, "rus");

```

```

51 List* ls = list_init();

52 get_string(ls);
53 unsigned n = get_i("N > ");
54 del_spaces(ls);
55 list_print(ls);

56 main: {
57 Node* ptr = ls->first;

58 while (ptr->next != NULL) {
59 int len = 1;

60 Node* tmp = ptr;
61 while (tmp->next != NULL) {
    a. if (IS_SPACE(tmp->next->data)) break;
    b. ++len;
    c. tmp = tmp->next;
62 }
63 for (unsigned i = 0; i < n; ++i) offset(ptr);

64 if (tmp->next == NULL) break;
65 else if (IS_SPACE(tmp->next->data)) ptr = tmp->next->next;
66 }
67 }

68 list_print(ls);

69 return 0;
70 }

```

5. Описание текстовых примеров

Программа №1

Значения массива	Ожидаемые значения	Полученное значение В
123 1234 312 1, 2	231 3412 123 1	231 3412 123 1
Oll ekk vova 15 17 1	Oll ekk ovav 51 71 1	Oll ekk ovav 51 71 1

6. Выводы

В ходе выполнения данной работы на примере программы, выполняющей сдвиг символов в каждом слове строки (представленном на физическом

уровне списком), мы научились работать со структурами данных, в основе которых лежат списки