

Правительство Российской Федерации

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Национальный исследовательский университет
«Высшая школа экономики»**

Московский институт электроники и математики Национального
исследовательского университета «Высшая школа экономики»

Факультет прикладной математики и кибернетики

О Т Ч Е Т

По лабораторной работе № 13

По курсу «Программирование»

ФИО студента	Номер группы	Дата	Баллы
Борисов Павел Геннадьевич	ПИ-11		

Москва - 2014 г.

Постановка задания

Вводится строка из строчных латинских букв, слова разделены пробелами, признак конца ввода - точка. При выполнении задания предложение организовать в виде двухсвязного списка слов (не кольца). Программа должна содержать функции формирования исходного списка, вывода списка до и после модификации и реализации непосредственно варианта задания.

Вариант 3

Перед словом с максимальной длиной вставить его копию

Makefile

```
1 all: lab13
2
3 lab13: linkedlist.c linkedlist.h main.c
4     gcc --std=c99 -ggdb -o lab13 main.c linkedlist.h linkedlist.c
5
6 clean:
7     rm -f lab13
```

linkedList.c

```
1 #include "linkedList.h"
2
3 Word* createLinkedListFromInput()
4 {
5     Word *head = malloc(sizeof(Word));
6     head->next = NULL;
7     head->prev = NULL;
8
9     Word *prevWord = head->prev;
10    Word *currWord = head;
11    //gather the input
12    char *line = NULL;
13    int nbytes = 0;
14    getdelim(&line, &nbytes, '.', stdin); //this one will gather all input before but '.'
15    int currSym = 0;
16    while(1)
17    {
18        char *text;
19        sscanf(line+currSym, "%ms", &text);
20        currSym += strlen(text);
21        currWord->word = text;
22
23        //remove whitespace
24        while (line[currSym] == ' ' || line[currSym] == '\n' || line[currSym] == '\t') currSym++;
25
26        if (line[currSym] == '\0')
27        {
28            text[strlen(text) - 1] = '\0'; //remove unusual dot.
29            break;
```

```

30     }
31     //double-linked list;
32     prevWord = currWord;
33     currWord = malloc(sizeof(Word));
34     prevWord->next = currWord;
35     currWord->prev = prevWord;
36     currWord->next = NULL;
37 }
38 free(line);
39 return head;
40 }
41
42 void freeWord(Word *w)
43 {
44     free(w->word);
45     free(w);
46 }
47
48 void freeAll(Word *w)
49 {
50     Word* w_next = w->next;
51     if (w_next)
52         freeAll(w_next);
53
54     freeWord(w);
55 }
56
57 void printSentence(Word *w)
58 {
59     printf("%s", w->word);
60     if (w->next)
61     {
62         printf(" ");
63         printSentence(w->next);
64     }
65     else
66         printf(".\n");
67 }

```

linkedlist.h

```

1  /* Moscow Institute of Electronics and Mathematics
2     The Faculty of Applied Mathematics
3
4     Assignment #13
5     Language: C99
6     Compiler: gcc
7
8     Student: Pavel Borisov
9     Group: Applied Informatics 11
10 */
11
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <string.h>
15

```

```

16 struct Node {
17     struct Node* prev;
18     struct Node* next;
19     char* word;
20 };
21
22 typedef struct Node Word;
23
24 Word* createLinkedListFromInput();
25
26 void freeWord(Word* w);
27 void freeAll(Word *w);
28
29 void printSentence(Word *w); //print the sentence

```

0.1. main.c

```

1 #include "linkedlist.h"
2
3 Word* insertWordCopyBeforeMaxLength(Word *w);
4
5 Word* maxLength(Word *w);
6
7
8 int main()
9 {
10     printf("Enter the sentence, ending with .\n");
11     Word *sent = createLinkedListFromInput();
12     printf("You have entered:\n");
13     printSentence(sent);
14     printf("Result:\n");
15     sent = insertWordCopyBeforeMaxLength(sent);
16     printSentence(sent);
17     return 0;
18 }
19
20
21 Word *maxLength(Word *w)
22 {
23     Word *maxlenWord = w;
24     int length = strlen(maxlenWord->word);
25     Word *currWord = w->next;
26     while(currWord)
27     {
28         int curlen = strlen(currWord->word);
29         if (curlen > length)
30         {
31             length = curlen;
32             maxlenWord = currWord;
33         }
34         currWord = currWord->next;
35     }
36     return maxlenWord;
37 }
38
39 Word * insertWordCopyBeforeMaxLength(Word *w)

```

```

40 {
41     Word* maxWord = maxLength(w);
42     Word* newWord = malloc(sizeof(Word));
43     char *text = calloc(strlen(maxWord->word)+1, sizeof(char));
44     strcpy(text, maxWord->word);
45     newWord->word = text;
46
47     newWord->prev = maxWord->prev;
48     newWord->prev->next = newWord;
49
50     newWord->next = maxWord;
51     maxWord->prev = newWord;
52     if (newWord->prev) //not first
53         return w;
54     else //first
55         return newWord;
56 }

```

Тесты

pasha@primum ~/projects/miem/1/4/13 (git)-[master] % ./lab13

Enter the sentence, ending with .

Workers of all countries unite.

You have entered:

Workers of all countries unite.

Result:

Workers of all countries countries unite.