

Отчет по курсовой работе №8

по курсу: языки и методы программирования

студент группы : М8О-105Б-21 Козлов Егор Сергеевич, № по списку: 9

Адреса www, e-mail, jabber, skype: iamaghoulzxc@gmail.com

Работа выполнена: "24 мая 2022 г."

Преподаватель: каф. 806 В.К.Титов

Входной контроль знаний с оценкой: _____

Отчет сдан " ____ " _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Линейные списки.

2. Цель работы: Составить и отладить программу на языке Си для обработки линейного списка заданной организации с отображением списка на динамические структуры

3. Задание (вариант 9): Вид списка: линейный двунаправленный; Нестандартное действие: добавить k экземпляров последнего элемента в начало списка

4. Оборудование(лабораторное):

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ ГБ

НМД _____ ГБ. Терминал _____ адрес _____. Принтер _____

Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Ryzen 7 5800 @ 8x 3.2 GHz, ОП 16384 МБ, НМД _____ ГБ. Монитор Встроенный

Другие устройства _____

5. Программное обеспечение(лабораторное):

Операционная система семейства UNIX, наименование _____ версия _____

Интерпретатор команд: _____ версия _____

Система программирования: _____ версия _____

Редактор текстов: _____ версия _____

Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождение и имена файлов и программ данных: _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Ubuntu версия 22.04

Интерпретатор команд: bash версия _____

Система программирования: C версия _____

Редактор текстов: Emacs версия _____

Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождение и имена файлов и программ данных: /usr/bin, а также /bin

6. Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)

При запуске программы вызывается меню со следующими функциями:

- 1: генерируется список из n элементов. Генерация происходит посредством создания n раз нового элемента с случайным значением. Указатель на следующий элемент последнего элемента списка указывает на новый элемент, указатель нового элемента на следующий указывает на NULL (никуда не указывает);
- 2: добавление нового элемента в конец списка. Действия аналогичны генерации списка, но значение элемента принимается из консоли и выполняется это действие один раз;
- 3: добавление элемента в начало списка. Создается новый элемент с значением, полученным от пользователя. Указатель нового элемента на следующий указывает на первый элемент прошлого списка, указатель на предыдущий элемент указывает на NULL (никуда не указывает);
- 4: удаление из списка веденного элемента. Проходим циклом по списку, пока не найдем элемент с нужным значением. С помощью указателей убираем его из списка и удаляем этот элемент из памяти;
- 5: вывод всех элементов списка. Выводим значения всех элементов, проходя список циклом;
- 6: вставка элемента после указанного. Проходим циклом по списку, пока не найдем элемент с указанным значением. Создаем новый элемент, указатель на следующий элемент у нового элемента указывает на элемент, который ранее шел после найденного, указатель на следующий элемент у найденного элемента указывает на только что созданный элемент;
- 7: очистка списка. Циклом проходим по списку и удаляем из памяти каждый элемент;
- 8: длина списка. Выводим длину списка, которая обновляется каждый раз, как только в нем появляется/исчезает элемент;
- 9: вывод меню;
- 10: особое действие над списком (вставка последнего элемента в начало списка k раз. Вставляем последний элемент списка k раз с помощью цикла и функции добавления элемента в начало списка;
- 0: выход из программы.

7. Сценарий выполнения работы [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты, либо соображения по тестированию].

kp8.cpp :

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 #include<time.h>
5
6 struct list;
7 typedef list *link;
8 typedef float item;
9
10 struct list {
11     item body;
12     link prev, next;
13 } *base, *newElement, *lastElement, *current;
14
15 int length = 0, i, n;
16
17 void randomize() {
18     long a = time(0);
19     srand(a);
20 }
21
22 void add(item data) {
23     if (!base) {
24         base = new list;
25         base->body = data;
26         base->next = nullptr;
27         base->prev = nullptr;
28         lastElement = base;
29     } else {
30         newElement = new list;
31         newElement->body = data;
32         lastElement->next = newElement;
33         newElement->next = nullptr;
34         newElement->prev = lastElement;
35         lastElement = newElement;
36     }
37     length++;
38 }
39
40 void addFirst(item data) {
```

```

41     if (!base) {
42         base = new list;
43         base->body = data;
44         base->next = nullptr;
45         base->prev = nullptr;
46         lastElement = base;
47     } else {
48         newElement = new list;
49         newElement->body = data;
50         newElement->next = base;
51         newElement->prev = nullptr;
52         base->prev = newElement;
53         current = newElement;
54         newElement = base;
55         base = current;
56     }
57     length++;
58 }
59
60 void printList() {
61     if (!base) {
62         printf("\nList is empty\n");
63     } else {
64         current = base;
65         printf("[ ");
66         while (current) {
67             if (current->next) { printf("%2.2f, ", current->body); }
68             else { printf("%2.2f ", current->body); }
69             current = current->next;
70         }
71         printf("]");
72     }
73     printf("\n");
74 }
75
76 void mainAction(int m) {
77     for (i = 0; i < m; ++i) {
78         addFirst(lastElement->body);
79     }
80 }
81
82 int listLength() {
83     return length;
84 }
85
86 void generateList(int n) {
87     for (i = 0; i < n; ++i) {
88         add(item(rand() % 10000) / 100);
89     }
90 }
91
92 void insert(item m, item data) {
93     if (!base) {
94         printf("\nList is empty\n");
95     } else {
96         i = 0;
97         current = base;
98         while (current) {
99             if (current->body == m) {
100                 newElement = new list;
101                 newElement->body = data;
102                 newElement->prev = current;
103                 newElement->next = current->next;
104                 current->next = newElement;
105                 i = 1;
106                 break;
107             }
108             current = current->next;
109         }
110     }
111     if (!i) printf("Item with this value not exists!\n");
112 }
113
114 void deleteElement(item m) {
115     i = 0;
116     current = base;
117     while (current) {
118         if (current->body == m) {
119             if (current == base) {

```

```

120         base = base->next;
121         delete current;
122         i = 1;
123         break;
124     } else if (current == lastElement) {
125         lastElement->prev->next = nullptr;
126         lastElement = lastElement->prev;
127         delete current;
128         current = nullptr;
129         i = 1;
130         break;
131     } else {
132         newElement = current->next;
133         current->prev->next = current->next;
134         current->next->prev = current->prev;
135         delete current;
136         i = 1;
137         break;
138     }
139 }
140 current = current->next;
141 }
142 if (!i) printf("Item with this value not exists!\n");
143 }
144
145 void eraseList() {
146     current = base;
147     while (current) {
148         newElement = current->next;
149         delete current;
150         current = newElement;
151     }
152     base = 0;
153 }
154
155 int main() {
156     float m, m1;
157     int k = 9;
158     for (;;) {
159         if (k == 9) {
160             printf("\nInput from 1 to 10 or 0 for actions:\n"
161                 "1. Generation list.\n"
162                 "2. Addition new items in end of list.\n"
163                 "3. Addition new items in begin of list.\n"
164                 "4. Deleting from list.\n"
165                 "5. Print list.\n"
166                 "6. Inserting in list.\n"
167                 "7. Erase list.\n"
168                 "8. Length of list.\n"
169                 "9. Print this MENU\n"
170                 "10. Main Action\n"
171                 "0. Exit out program.\n"
172                 "=> ");
173         }
174
175         printf("\nInput number of MENU: ");
176         scanf("%d", &k);
177
178         if (!k) break;
179
180         if (k == 1) {
181             printf("\nInput number of new items of list: n = ");
182             scanf("%d", &n);
183             generateList(n);
184         } else if (k == 2) {
185             printf("\nInput value of adding item: m = ");
186             scanf("%f", &m);
187             add(m);
188         } else if (k == 3) {
189             printf("\nInput value of adding item: m = ");
190             scanf("%f", &m);
191             addFirst(m);
192         } else if (k == 4) {
193             scanf("%f", &m);
194             deleteElement(m);
195         }
196     }
197 }
198

```

```

199     } else if (k == 5) printList();
200
201     else if (k == 6) {
202
203         printf("\nInput value inserting item = ");
204         scanf("%f", &m);
205         printf("\nInput value item after which to insert = ");
206         scanf("%f", &m1);
207         insert(m, m1);
208
209     } else if (k == 7) eraseList();
210
211     else if (k == 8) {
212         printf("\nLenght of list = %d\n", listLength());
213
214     } else if (k == 10) {
215         if (!base) {
216             printf("\nList is empty!\n");
217             continue;
218         }
219
220         printf("\nInput value for main action = ");
221         scanf("%d", &n);
222         mainAction(n);
223
224     } else if (k > 10) printf("Have no such number of MENU\n");
225 }
226
227 return 0;
228 }

```

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

8. Распечатка протокола (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем).

```
isitmuse@isitmuse:~/lab/secondSem/kp8$ cat head
```

```
|||||КУРСОВАЯ РАБОТА №8||||| | |
||||||ЛИНЕЙНЫЕ СПИСКИ||||||
|||ВЫПОЛНИЛ СТУДЕНТ ГРУППЫ|||
|||М80-105Б-21 КОЗЛОВ ЕГОР|||
\\\\\\\\\\\\\\\\\\\\\\|\\\\\\\\\\\\\\\\\\\\\\
isitmuse@isitmuse:~/lab/secondSem/kp8$ cat kp8.cpp
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>

struct list;
typedef list *link;
typedef float item;

struct list {
    item body;
    link prev, next;
} *base, *newElement, *lastElement, *current;

int length = 0, i, n;

void randomize() {
    long a = time(0);
    srand(a);
}

void add(item data) {
    if (!base) {
        base = new list;
        base->body = data;
        base->next = nullptr;
        base->prev = nullptr;
        lastElement = base;
    } else {
        newElement = new list;
        newElement->body = data;
        lastElement->next = newElement;
        newElement->next = nullptr;
        newElement->prev = lastElement;
        lastElement = newElement;
    }
    length++;
}

void addFirst(item data) {
    if (!base) {
        base = new list;
        base->body = data;
        base->next = nullptr;
        base->prev = nullptr;
        lastElement = base;
    } else {
        newElement = new list;
        newElement->body = data;
        newElement->next = base;
        newElement->prev = nullptr;
        base->prev = newElement;
```

```

        current = newElement;
        newElement = base;
        base = current;
    }
    length++;
}

void printList() {
    if (!base) {
        printf("\nList is empty\n");
    } else {
        current = base;
        printf("[ ");
        while (current) {
            if (current->next) { printf("%.2f, ", current->body); }
            else { printf("%.2f ", current->body); }
            current = current->next;
        }
        printf("]");
    }
    printf("\n");
}

void mainAction(int m) {
    for (i = 0; i < m; ++i) {
        addFirst(lastElement->body);
    }
}

int listLength() {
    return length;
}

void generateList(int n) {
    for (i = 0; i < n; ++i) {
        add(item(rand() % 10000) / 100);
    }
}

void insert(item m, item data) {
    if (!base) {
        printf("\nList is empty\n");
    } else {
        i = 0;
        current = base;
        while (current) {
            if (current->body == m) {
                newElement = new list;
                newElement->body = data;
                newElement->prev = current;
                newElement->next = current->next;
                current->next = newElement;
                i = 1;
                break;
            }
            current = current->next;
        }
        if (!i) printf("Item with this value not exists!\n");
    }
}

void deleteElement(item m) {
    i = 0;

```

```

current = base;
while (current) {
    if (current->body == m) {
        if (current == base) {
            base = base->next;
            delete current;
            i = 1;
            break;
        } else if (current == lastElement) {
            lastElement->prev->next = nullptr;
            lastElement = lastElement->prev;
            delete current;
            current = nullptr;
            i = 1;
            break;
        } else {
            newElement = current->next;
            current->prev->next = current->next;
            current->next->prev = current->prev;
            delete current;
            i = 1;
            break;
        }
    }
    current = current->next;
}
if (!i) printf("Item with this value not exists!\n");
}

void eraseList() {
    current = base;
    while (current) {
        newElement = current->next;
        delete current;
        current = newElement;
    }
    base = 0;
}

int main() {
    float m, m1;
    int k = 9;
    for (;;) {
        if (k == 9) {
            printf("\nInput from 1 to 10 or 0 for actions:\n"
                "1. Generation list.\n"
                "2. Addition new items in end of list.\n"
                "3. Addition new items in begin of list.\n"
                "4. Deleting from list.\n"
                "5. Print list.\n"
                "6. Inserting in list.\n"
                "7. Erase list.\n"
                "8. Lenght of list.\n"
                "9. Print this MENU\n"
                "10. Main Action\n"
                "0. Exit out program.\n"
                "=> ");
        }

        printf("\nInput number of MENU: ");
        scanf("%d", &k);

        if (!k) break;
    }
}

```



```

    if (k == 1) {
        printf("\nInput number of new items of list: n = ");
        scanf("%d", &n);
        generateList(n);

    } else if (k == 2) {
        printf("\nInput value of adding item: m = ");
        scanf("%f", &m);
        add(m);

    } else if (k == 3) {
        printf("\nInput value of adding item: m = ");
        scanf("%f", &m);
        addFirst(m);

    } else if (k == 4) {
        scanf("%f", &m);
        deleteElement(m);

    } else if (k == 5) printList();

    else if (k == 6) {

        printf("\nInput value inserting item = ");
        scanf("%f", &m);
        printf("\nInput value item after which to insert = ");
        scanf("%f", &m1);
        insert(m, m1);

    } else if (k == 7) eraseList();

    else if (k == 8) {
        printf("\nLenght of list = %d\n", listLength());

    } else if (k == 10) {
        if (!base) {
            printf("\nList is empty!\n");
            continue;
        }

        printf("\nInput value for main action = ");
        scanf("%d", &n);
        mainAction(n);

    } else if (k > 10) printf("Have no such number of MENU\n");
}

```

```

return 0;

```

```

}isitmuse@isitmuse:~/lab/secondSem/kp8$ g++ -o kp8 kp8.cpp
isitmuse@isitmuse:~/lab/secondSem/kp8$ ./kp8

```

Input from 1 to 10 or 0 for actions:

1. Generation list.
2. Addition new items in end of list.
3. Addition new items in begin of list.
4. Deleting from list.
5. Print list.
6. Inserting in list.
7. Erase list.
8. Lenght of list.
9. Print this MENU
10. Main Action

```

0. Exit out program.
=>
Input number of MENU: 1

Input number of new items of list: n = 10

Input number of MENU: 5
[ 93.83, 8.86, 27.77, 69.15, 77.93, 83.35, 53.86, 4.92, 66.49, 14.21 ]

Input number of MENU: 2

Input value of adding item: m = 13.72

Input number of MENU: 5
[ 93.83, 8.86, 27.77, 69.15, 77.93, 83.35, 53.86, 4.92, 66.49, 14.21, 13.72 ]

Input number of MENU: 3

Input value of adding item: m = 23.43

Input number of MENU: 5
[ 23.43, 93.83, 8.86, 27.77, 69.15, 77.93, 83.35, 53.86, 4.92, 66.49, 14.21, 13.72 ]

Input number of MENU: 4
27.77

Input number of MENU: 5
[ 23.43, 93.83, 8.86, 69.15, 77.93, 83.35, 53.86, 4.92, 66.49, 14.21, 13.72 ]

Input number of MENU: 5
[ 23.43, 93.83, 8.86, 69.15, 77.93, 83.35, 53.86, 4.92, 66.49, 14.21, 13.72 ]

Input number of MENU: 9

Input from 1 to 10 or 0 for actions:
1. Generation list.
2. Addition new items in end of list.
3. Addition new items in begin of list.
4. Deleting from list.
5. Print list.
6. Inserting in list.
7. Erase list.
8. Lenght of list.
9. Print this MENU
10. Main Action
0. Exit out program.
=>
Input number of MENU: 6

Input value inserting item = 77.93

Input value item after which to insert = 83.55

Input number of MENU: 5
[ 23.43, 93.83, 8.86, 69.15, 77.93, 83.55, 83.35, 53.86, 4.92, 66.49, 14.21, 13.72 ]

Input number of MENU: 8

Lenght of list = 12

Input number of MENU: 10

Input value for main action = 5

```

Input number of MENU: 5

[13.72, 13.72, 13.72, 13.72, 13.72, 23.43, 93.83, 8.86, 69.15, 77.93, 83.55, 83.35, 53.86, 4.92, 66.49

Input number of MENU: 8

Lenght of list = 17

Input number of MENU: 0

isitmuse@isitmuse:~/lab/secondSem/kp8\$

9. Дневник отладки должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечание автора по существу работы _____

11. Выводы _____ Я научился работать с линейными списками.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом _____

Подпись студента _____