

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной техники

ОТЧЕТ
по лабораторной работе № 8
по дисциплине «Программирование»
ТЕМА: ЛИНЕЙНЫЕ ОДНОСВЯЗНЫЕ СПИСКИ.

Студент гр. 3312

Шарапов И. Д.

Преподаватель

Аббас С. А.

Санкт-Петербург

2024

Содержание

Цель работы	3
Задание (Вариант 1)	3
Постановка задачи и описание решения.....	3
Описание переменных	4
Схема подалгоритма	5
Текст программы.....	5
Контрольные примеры.....	10
Содержимое файлов.....	12
Примеры выполнения программы.....	14
Выводы	17

Цель работы

Целью работы является изучение линейных односвязных списков в языке Си, а также работа с указателями на структуры.

Задание (Вариант 1)

Разработать подалгоритм и написать функцию, вставляющую в односвязный список получаемые данные перед заданным по номеру элементом. Номер элемента задаётся с конца списка. При недостаточном количестве элементов в списке данные вставить в начало списка.

Постановка задачи и описание решения

Для выполнения задания в код прошлой лабораторной работы было добавлено 4 функции.

Функция *make_list()* инициализирует список: выделяет память для головы и заполняет её поля.

Функция *create_node()* выделяет память для нового элемента списка и заполняет его с помощью функции *fill_struct()*.

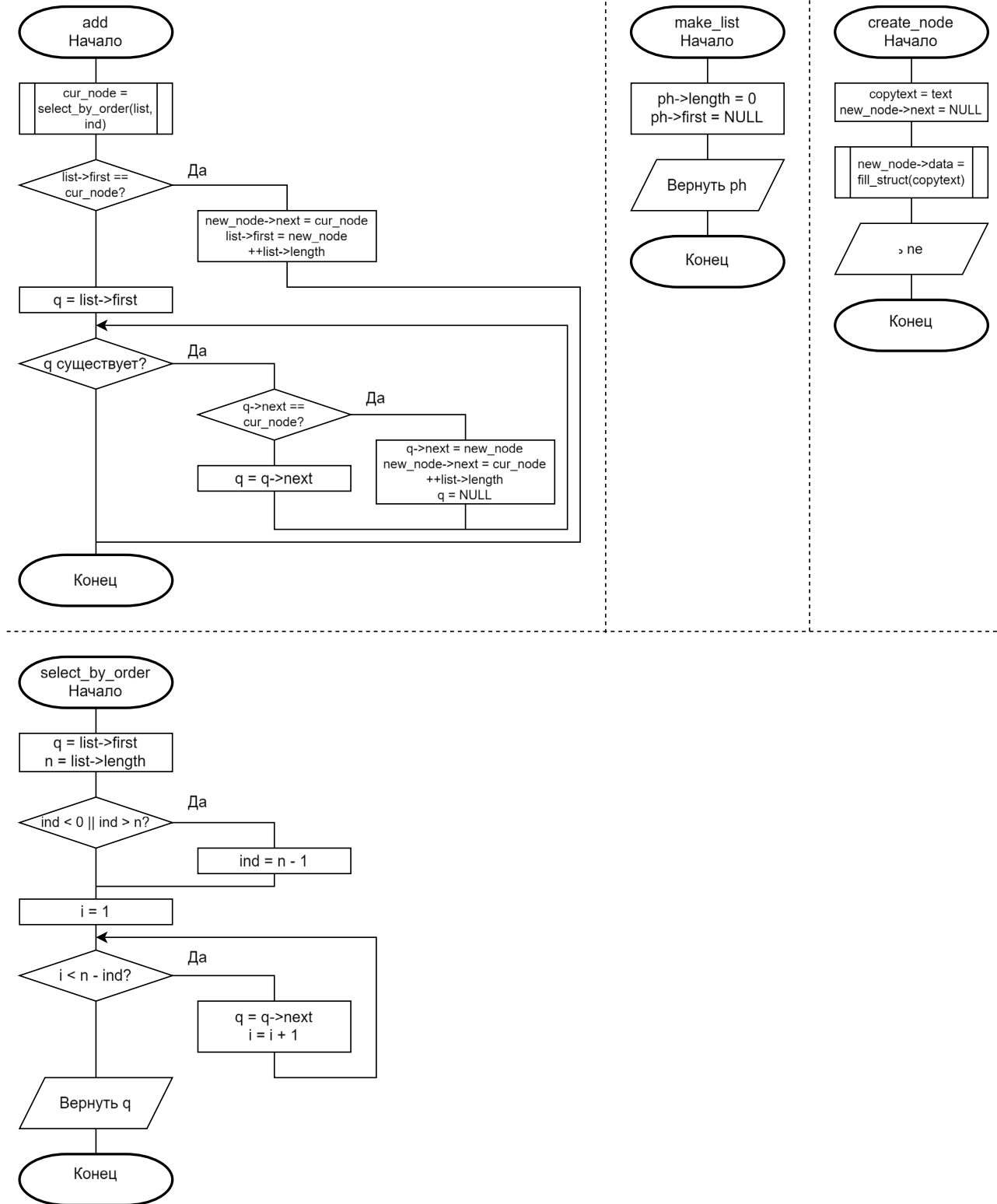
Функция *select_by_order()* перебирает все элементы списка до тех пор, пока не найдёт элемент с указанным индексом и возвращает указатель на его. Если элемента с таким индексом не существует, функция выведет соответствующее сообщение и вернёт указатель на первый элемент списка.

Функция *add()* вставляет в список элемент. Для этого строка преобразуется в элемент списка с помощью функции *create_node()* и ищется элемент, перед которым нужно вставить, с помощью функции *select_by_order()*. В зависимости от положения элемента меняются указатели соседних элементов, тем самым новый элемент добавляется в список.

Описание переменных

№	Имя переменной	Тип	Назначение
Функция <i>ListOfAthlete *make_list()</i>			
1	ph	ListOfAthlete *	Указатель на голову списка
Функция <i>NodeOfList *create_node(const char *text)</i>			
1	text	char *	Исходная строка с данными
2	new_node	NodeOfList *	Новый элемент списка
3	copytext	char *	Копия исходной строки
Функция <i>NodeOfList *select_by_order(const ListOfAthlete *list, int ind)</i>			
1	list	ListOfAthlete *	Указатель на голову списка
2	ind	int	Номер элемента, который нужно найти
3	q	NodeOfList *	Временный элемент для перебора
4	n	int	Длина списка
Функция <i>void add(ListOfAthlete *list, NodeOfList *new_node, int ind)</i>			
1	list	ListOfAthlete *	Указатель на голову списка
2	new_node	NodeOfList *	Элемент, который нужно вставить
3	ind	int	Номер элемента, перед которым нужно вставить
4	cur_node	NodeOfList *	Указатель на элемент, перед которым нужно вставить
5	q	NodeOfList *	Временный элемент для перебора

Схема подалгоритма



Текст программы

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Athlete {
    char *name;
    char *university;
}
  
```

```

    int age;
    float weight;
    int height;
    int result[3];
    float index;
} Athlete;

typedef struct NodeOfList {
    Athlete *data;
    struct NodeOfList *next;
} NodeOfList;

typedef struct ListOfAthlete {
    int length;
    NodeOfList *first;
} ListOfAthlete;

int from_str_to_int(char *x) {
    int ans = 0;

    while (*x != '\0') {
        ans = ans * 10 + (*x - '0');
        ++x;
    }
    return ans;
}

float from_str_to_float(char *x) {
    float ans = 0, a = 10, b = 1;

    while (*x != '\0') {
        if (*x == '.' || *x == ',') {
            a = 1;
            b = 10;
        } else {
            ans = ans * a + (float) (*x - '0') / b;
            if (b > 1) b *= 10;
        }
        ++x;
    }
    return ans;
}

void from_str_to_int_mas(char *x, int *mas) {
    int ind = 0, j = 0;

    while (x[j] != '\0') {
        if (x[j] == ',') {
            x[j] = '\0';
            mas[ind++] = from_str_to_int(x);
            x += j + 1;
            j = -1;
        }
        ++j;
    }
    mas[ind] = from_str_to_int(x);
}

Athlete *fill_struct(char *str) {
    Athlete *user = NULL;
    char *word = str;
    int ind = 0, tt;
    char *pole[5];

    user = (Athlete *) malloc(sizeof(Athlete));
    if (user != NULL) {
        for (tt = 0; str[tt] != '\n' && str[tt] != '\0'; ++tt) {
            if (str[tt] == ',' && ind < 5) {
                str[tt] = '\0';
                pole[ind++] = word;
                word = str + tt + 1;
            }
        }
        str[tt] = '\0';
        user->name = pole[0];
        user->university = pole[1];
        user->age = from_str_to_int(pole[2]);
        user->weight = from_str_to_float(pole[3]);
        user->height = from_str_to_int(pole[4]);
        from_str_to_int_mas(word, user->result);
        user->index = (float) (user->result[0] + user->result[1] + user->result[2]) /
user->weight;
    }
    return user;
}

ListOfAthlete *make_list() {
    ListOfAthlete *ph = NULL;

```

```

ph = (ListOfAthlete *) malloc(sizeof(ListOfAthlete));
if (ph != NULL) {
    ph->length = 0;
    ph->first = NULL;
}
return ph;
}

NodeOfList *create_node(const char *text) {
    NodeOfList *new_node = NULL;
    char *copytext = NULL;

    new_node = (NodeOfList *) malloc(sizeof(NodeOfList));
    copytext = (char *) malloc((strlen(text) + 1) * sizeof(char));
    if (new_node && copytext) {
        strcpy(copytext, text);
        new_node->data = fill_struct(copytext);
        new_node->next = NULL;
    }
    return new_node;
}

NodeOfList *select_by_order(const ListOfAthlete *list, int ind) {
    NodeOfList *q = list->first;
    int n = list->length;

    if (ind < 0 || ind > n) {
        ind = n - 1;
        printf("The index is out of range!\n"
            "The item will be inserted at the top of the list.\n");
    }
    for (int i = 1; i < n - ind; ++i) q = q->next;
    return q;
}

void add(ListOfAthlete *list, NodeOfList *new_node, int ind) {
    NodeOfList *cur_node = select_by_order(list, ind);
    NodeOfList *q = NULL;

    if (list && new_node && cur_node) {
        if (list->first == cur_node) {
            new_node->next = cur_node;
            list->first = new_node;
            ++list->length;
        } else {
            q = list->first;
            while (q != NULL) {
                if (q->next == cur_node) {
                    q->next = new_node;
                    new_node->next = cur_node;
                    ++list->length;
                    q = NULL;
                } else q = q->next;
            }
        }
    }
}

void free_list(ListOfAthlete *list) {
    NodeOfList *cur_node = list->first;
    NodeOfList *next_node;

    free(list);
    while (cur_node != NULL) {
        next_node = cur_node->next;
        free(cur_node->data);
        free(cur_node);
        cur_node = next_node;
    }
}

void print_line() {
    printf("+");
    for (int i = 0; i < 22; printf("-"), ++i);
    printf("+-----+");
    printf("\n");
}

void print_node(Athlete *node) {
    printf("| %-20s | %-10s | %-3i | %0.1f ", node->name, node->university,
        node->age, node->weight);
    if (node->weight < 100) printf(" ");
    printf("| %-6i | %-4i | %-4i | %-4i ", node->height, node->result[0],
        node->result[1], node->result[2]);
    printf("| %0.3f |\n", node->index);
}

void pprint(const ListOfAthlete *list) {
    NodeOfList *cur_node = list->first;

```

```

    print line();
    printf("| Name          | University | Age | Weight | Height "
           "| Res1 | Res2 | Res3 | Index |\n");
    print line();
    while (cur node != NULL) {
        print node(cur node->data);
        cur node = cur node->next;
    }
    print line();
}

NodeOfList **get mas(const ListOfAthlete *list) {
    NodeOfList *cur node = list->first;
    NodeOfList **mas = NULL;

    mas = (NodeOfList **) malloc(list->length * sizeof(NodeOfList *));
    if (mas != NULL) {
        for (int i = 0; cur node != NULL; ++i) {
            mas[i] = cur node;
            cur node = cur node->next;
        }
    }
    return mas;
}

void my swap(NodeOfList **mas, ListOfAthlete *list, int i, int j) {
    NodeOfList *q;

    if (i == 0) {
        list->first = mas[j];
    } else {
        mas[i - 1]->next = mas[j];
    }
    mas[j - 1]->next = mas[i];
    q = mas[j]->next;
    mas[j]->next = mas[i]->next;
    mas[i]->next = q;
    q = mas[i];
    mas[i] = mas[j];
    mas[j] = q;
}

void sort_list(ListOfAthlete *list, int param) {
    NodeOfList **mas = get mas(list);
    int n = list->length;

    for (int i = 0; i < n; ++i) {
        for (int j = i; j < n; ++j) {
            if ((param == 1 && mas[i]->data->age > mas[j]->data->age) ||
                (param == 2 && mas[i]->data->weight > mas[j]->data->weight) ||
                (param == 3 && mas[i]->data->height > mas[j]->data->height) ||
                (param == 4 && mas[i]->data->index > mas[j]->data->index)) {
                my swap(mas, list, i, j);
            }
        }
    }
    free(mas);
}

char *m_strlwr(const char *str) {
    char *new str = NULL;
    new str = (char *) malloc((strlen(str) + 1) * sizeof(char));

    if (new str != NULL) {
        strcpy(new str, str);
        strlwr(new str);
    }
    return new str;
}

void sorted(int *mas, const ListOfAthlete *list, int param) {
    NodeOfList *cur node, *min node;
    int ind;

    for (int j = 0; j < list->length; ++j) {
        cur node = list->first;
        min node = NULL;
        for (int i = 0; cur node != NULL && i < list->length; ++i, cur node = cur node-
>next) {
            if (mas[i] == 1) {
                if ((min node == NULL) ||
                    ((param == 1 && min node->data->age > cur node->data->age) ||
                     (param == 2 && min node->data->weight > cur node->data->weight) ||
                     (param == 3 && min node->data->height > cur node->data->height) ||
                     (param == 4 && min node->data->index > cur node->data->index))) {
                        min node = cur node;
                        ind = i;
                    }
            }
        }
    }
}

```



```

    }
    if (min node != NULL) {
        mas[ind] = 2;
        print node(min node->data);
    }
}

for (int j = 0; j < list->length; ++j) {
    if (mas[j] == 2) mas[j] = 1;
}
}

void find(ListOfAthlete *list, int param) {
    NodeOfList *cur node = list->first;
    char x[128], *str, *new str;
    int mas[list->length], fl = 0, ch;

    printf("Enter the search string:\n");
    getchar();
    fgets(x, sizeof(x), stdin);
    x[strlen(x) - 1] = '\0';
    strlwr(x);
    for (int i = 0; cur node != NULL && i < list->length; ++i) {
        if (param == 1) str = cur node->data->name;
        else str = cur node->data->university;
        new str = m_strlwr(str);
        if (strstr(new str, x) != NULL) {
            if (fl == 0) {
                print line();
                printf("| Name | University | Age | Weight | Height | Res1 | Res2 | Res3 | Index |\n");
                print line();
            }
            print node(cur node->data);
            fl = 1;
            mas[i] = 1;
        } else {
            mas[i] = 0;
        }
        free(new str);
        cur node = cur node->next;
    }
    if (fl == 0) {
        printf("No matches found!\n");
    } else {
        print line();
        do {
            printf("Select a field to sort by or exit:\n"
                "1 = age\n"
                "2 = weight\n"
                "3 = height\n"
                "4 = index\n"
                "0 = exit\n"
                "Enter only one number!\n");
            scanf("%i", &ch);
            if (ch < 0 || 4 < ch) {
                printf("Invalid command!\n");
            } else if (ch != 0) {
                print line();
                printf("| Name | University | Age | Weight | Height | Res1 | Res2 | Res3 | Index |\n");
                print line();
                sorted(mas, list, ch);
                print line();
            } else {
                printf("To display the data, enter the command \\"!print\\"\\n"
                    "To find athletes, enter the command \\"!find\\"\\n"
                    "To sort the data, enter the command \\"!sort\\"\\n"
                    "To add new data, enter the command \\"!add\\"\\n"
                    "To end the program, enter the command \\"!end\\"\\n");
            }
        } while (ch != 0);
    }
}

int main() {
    ListOfAthlete *list;
    NodeOfList *cur node = NULL, *last node = NULL;
    char filename[128], text[1024], str[128];
    FILE *f;
    int ch;

    printf("Please enter the file name:\n");
    scanf("%s", filename);
    f = fopen(filename, "r");
    while (f == NULL) {
        printf("Something went wrong!\n"
            "Perhaps such a file does not exist.\n");
    }
}

```

```

        "Please enter the file name again:\n");
        scanf("%s", filename);
        f = fopen(filename, "r");
    }

    list = make_list();
    while (fgets(text, sizeof(text), f)) {
        if (list->length == 0) {
            cur node = create_node(text);
            list->first = cur node;
            last node = cur node;
        } else {
            cur node = create_node(text);
            last node->next = cur node;
            last node = cur node;
        }
        ++list->length;
    }
    printf("The file has successfully been processed!\n");
    fclose(f);

    printf("To display the data, enter the command \n"!print\n"
        "To find athletes, enter the command \n"!find\n"
        "To sort the data, enter the command \n"!sort\n"
        "To add new data, enter the command \n"!add\n"
        "To end the program, enter the command \n"!end\n");

    do {
        scanf("%s", str);
        if (!strcmp(str, "!end")) {
            printf("Goodbye!\n");
        } else if (!strcmp(str, "!print")) {
            pprint(list);
        } else if (!strcmp(str, "!find")) {
            printf("Select a field to find by:\n"
                "1 = name\n"
                "2 = university\n"
                "Enter only one number!\n");
            scanf("%i", &ch);
            if (ch < 1 || 2 < ch) {
                printf("Invalid command!\n");
            } else {
                find(list, ch);
            }
        } else if (!strcmp(str, "!sort")) {
            printf("Select a field to sort by:\n"
                "1 = age\n"
                "2 = weight\n"
                "3 = height\n"
                "4 = index\n"
                "Enter only one number!\n");
            scanf("%i", &ch);
            if (ch < 1 || 4 < ch) {
                printf("Invalid command!\n");
            } else {
                sort_list(list, ch);
                printf("The data has been successfully sorted!\n");
                pprint(list);
            }
        } else if (!strcmp(str, "!add")) {
            printf("Enter data of the athlete in format:\n"
                "name;university;age;weight;height;result1,result2,result3\n");
            getchar();
            fgets(text, sizeof(text), stdin);
            printf("Enter the number of the item indicated at the end\n"
                "of the list before which you want to insert the athlete:\n");
            scanf("%i", &ch);
            --ch;
            add(list, create_node(text), ch);
            printf("The item has been successfully inserted!\n");
        } else {
            printf("Unknown command!\n");
        }
    } while (strcmp(str, "!end") != 0);

    free list(list);
    return 0;
}

```

Контрольные примеры

№	Исходные данные	Результаты
---	--------------------	------------

1	<pre>input1.csv !print !add Dmitriev D.S.;TomSU;23; 67.8;172;150;190 ;220 3 !print !end</pre>	<pre>Please enter the file name: The file has successfully been processed! To display the data, enter the command "!print" To find athletes, enter the command "!find" To sort the data, enter the command "!sort" To add new data, enter the command "!add" To end the program, enter the command "!end" +-----+-----+-----+-----+-----+-----+-----+-----+ Name University Age Weight Height Res1 Res2 Res3 Index +-----+-----+-----+-----+-----+-----+-----+-----+ Ivanov I.I. MSU 25 70.5 175 120 200 230 7.801 Petrov P.P. SPbSU 22 65.2 180 140 180 210 8.129 Sidorov S.S. MIPT 18 55.8 165 90 150 180 7.527 Kuznetsov K.K. HSE 20 75.1 185 200 220 240 8.788 Smirnov A.A. MGU 27 90.3 190 180 220 250 7.198 Fedorov F.F. NRU HSE 19 68.7 170 110 170 200 6.987 Volkov V.V. BSU 21 60.0 160 80 0 0 1.333 Mikhailov M.M. TSU 24 72.4 178 160 190 220 7.873 Novikov N.N. ITMO 26 80.6 195 210 240 250 8.685 Morozov M.I. RANEPA 23 73.8 183 170 200 230 8.130 +-----+-----+-----+-----+-----+-----+-----+-----+ Enter data of the athlete in format: name;university;age;weight;height;result1,result2,result3 Enter the number of the item indicated at the end of the list before which you want to insert the athlete: The item has been successfully inserted! +-----+-----+-----+-----+-----+-----+-----+-----+ Name University Age Weight Height Res1 Res2 Res3 Index +-----+-----+-----+-----+-----+-----+-----+-----+ Ivanov I.I. MSU 25 70.5 175 120 200 230 7.801 Petrov P.P. SPbSU 22 65.2 180 140 180 210 8.129 Sidorov S.S. MIPT 18 55.8 165 90 150 180 7.527 Kuznetsov K.K. HSE 20 75.1 185 200 220 240 8.788 Smirnov A.A. MGU 27 90.3 190 180 220 250 7.198 Fedorov F.F. NRU HSE 19 68.7 170 110 170 200 6.987 Volkov V.V. BSU 21 60.0 160 80 0 0 1.333 Dmitriev D.S. TomSU 23 67.8 172 150 190 220 8.260 Mikhailov M.M. TSU 24 72.4 178 160 190 220 7.873 Novikov N.N. ITMO 26 80.6 195 210 240 250 8.685 Morozov M.I. RANEPA 23 73.8 183 170 200 230 8.130 +-----+-----+-----+-----+-----+-----+-----+-----+ Goodbye!</pre>
2	<pre>input2.txt input2.csv !add Sorokin S.O.;MIIT;17;57. 9;165;100;160;1 90 15 !print !end</pre>	<pre>Please enter the file name: Something went wrong! Perhaps such a file does not exist. Please enter the file name again: The file has successfully been processed! To display the data, enter the command "!print" To find athletes, enter the command "!find" To sort the data, enter the command "!sort" To add new data, enter the command "!add" To end the program, enter the command "!end" Enter data of the athlete in format: name;university;age;weight;height;result1,result2,result3 Enter the number of the item indicated at the end of the list before which you want to insert the athlete: The index is out of range! The item will be inserted at the top of the list. The item has been successfully inserted! +-----+-----+-----+-----+-----+-----+-----+-----+ Name University Age Weight Height Res1 Res2 Res3 Index +-----+-----+-----+-----+-----+-----+-----+-----+ Sorokin S.O. MIIT 17 57.9 165 100 160 190 7.772 Kozlov K.V. UralSU 28 85.2 188 220 240 250 8.333 Orlov O.D. MIIT 17 57.5 163 100 160 0 4.522 Nikitin N.P. SFU 30 95.0 200 230 250 250 7.684 Kovalev K.N. KPI 16 50.6 155 70 120 150 6.719 Ilyin I.S. SPbPU 20 63.9 172 130 180 210 8.138 Sergeev S.I. TomSU 22 67.2 168 0 0 0 0.000 Stepanov S.V. BSTU 18 58.4 166 120 170 200 8.390 Gusev G.A. KAI 26 76.8 182 200 220 240 8.594 Popov P.V. NSU 25 71.7 176 180 210 230 8.647 Vasiliev V.S. TSU 19 66.0 170 110 160 190 6.970 +-----+-----+-----+-----+-----+-----+-----+-----+ Goodbye!</pre>

3

input3.csv

!add

Sokolov

S.V.;TSU;20;66.6;171;110;160;190

4

!print

!add

Vorobyov

V.I.;MSU;28;86.2;186;220;240;250

50

7

!print

!end

Please enter the file name:
The file has successfully been processed!
To display the data, enter the command "!print"
To find athletes, enter the command "!find"
To sort the data, enter the command "!sort"
To add new data, enter the command "!add"
To end the program, enter the command "!end"
Enter data of the athlete in format:
name;university;age;weight;height;result1,result2,result3
Enter the number of the item indicated at the end
of the list before which you want to insert the athlete:
The item has been successfully inserted!

Name	University	Age	Weight	Height	Res1	Res2	Res3	Index
Pavlov P.A.	MSU	24	70.2	177	160	200	230	8.405
Kiselev K.A.	SPbSU	20	65.8	175	140	180	210	8.055
Bogdanov B.B.	MIPT	18	55.5	162	90	150	180	7.568
Danilov D.D.	HSE	22	75.3	183	200	220	240	8.765
Nesterov N.N.	MGU	27	90.8	188	180	220	250	7.159
Romanov R.I.	NRU HSE	19	68.2	168	110	170	200	7.038
Sokolov S.V.	TSU	20	66.6	171	110	160	190	6.907
Zaitsev Z.I.	BSU	21	60.3	163	80	130	160	6.136
Konovalev K.M.	TSU	25	72.6	179	120	190	220	7.300
Frolov F.P.	ITMO	26	101.2	192	210	240	250	6.917
Belov B.V.	UralSU	28	85.8	190	220	240	250	8.275

Enter data of the athlete in format:
name;university;age;weight;height;result1,result2,result3
Enter the number of the item indicated at the end
of the list before which you want to insert the athlete:
The item has been successfully inserted!

Name	University	Age	Weight	Height	Res1	Res2	Res3	Index
Pavlov P.A.	MSU	24	70.2	177	160	200	230	8.405
Kiselev K.A.	SPbSU	20	65.8	175	140	180	210	8.055
Bogdanov B.B.	MIPT	18	55.5	162	90	150	180	7.568
Danilov D.D.	HSE	22	75.3	183	200	220	240	8.765
Vorobyov V.I.	MSU	28	86.2	186	220	240	250	8.237
Nesterov N.N.	MGU	27	90.8	188	180	220	250	7.159
Romanov R.I.	NRU HSE	19	68.2	168	110	170	200	7.038
Sokolov S.V.	TSU	20	66.6	171	110	160	190	6.907
Zaitsev Z.I.	BSU	21	60.3	163	80	130	160	6.136
Konovalev K.M.	TSU	25	72.6	179	120	190	220	7.300
Frolov F.P.	ITMO	26	101.2	192	210	240	250	6.917
Belov B.V.	UralSU	28	85.8	190	220	240	250	8.275

Goodbye!

Содержимое файлов

input1.csv ×	
1	Ivanov I.I.;MSU;25;70.5;175;120;200;230
2	Petrov P.P.;SPbSU;22;65.2;180;140;180;210
3	Sidorov S.S.;MIPT;18;55.8;165;90;150;180
4	Kuznetsov K.K.;HSE;20;75.1;185;200;220;240
5	Smirnov A.A.;MGU;27;90.3;190;180;220;250
6	Fedorov F.F.;NRU HSE;19;68.7;170;110;170;200
7	Volkov V.V.;BSU;21;60.0;160;80;0;0
8	Mikhailov M.M.;TSU;24;72.4;178;160;190;220
9	Novikov N.N.;ITMO;26;80.6;195;210;240;250
10	Morozov M.I.;RANEPA;23;73.8;183;170;200;230

input2.csv ×	
1	Kozlov K.V.;UralSU;28;85.2;188;220;240;250
2	Orlov O.D.;MIIT;17;57.5;163;100;160;0
3	Nikitin N.P.;SFU;30;95.0;200;230;250;250
4	Kovalev K.N.;KPI;16;50.6;155;70;120;150
5	Ilyin I.S.;SPbPU;20;63.9;172;130;180;210
6	Sergeev S.I.;TomSU;22;67.2;168;0;0;0
7	Stepanov S.V.;BSTU;18;58.4;166;120;170;200
8	Gusev G.A.;KAI;26;76.8;182;200;220;240
9	Popov P.V.;NSU;25;71.7;176;180;210;230
10	Vasiliev V.S.;TSU;19;66.0;170;110;160;190

input3.csv ×	
1	Pavlov P.A.;MSU;24;70.2;177;160;200;230
2	Kiselev K.A.;SPbSU;20;65.8;175;140;180;210
3	Bogdanov B.B.;MIPT;18;55.5;162;90;150;180
4	Danilov D.D.;HSE;22;75.3;183;200;220;240
5	Nesterov N.N.;MGU;27;90.8;188;180;220;250
6	Romanov R.I.;NRU HSE;19;68.2;168;110;170;200
7	Zaitsev Z.I.;BSU;21;60.3;163;80;130;160
8	Konovarov K.M.;TSU;25;72.6;179;120;190;220
9	Frolov F.P.;ITMO;26;101.2;192;210;240;250
10	Belov B.V.;UralSU;28;85.8;190;220;240;250

Примеры выполнения программы

D:\VUZ\LAB_08_01\cmake-build-debug\LAB_08_01.exe

Please enter the file name:

input1.csv

The file has successfully been processed!

To display the data, enter the command "!print"

To find athletes, enter the command "!find"

To sort the data, enter the command "!sort"

To add new data, enter the command "!add"

To end the program, enter the command "!end"

!print

Name	University	Age	Weight	Height	Res1	Res2	Res3	Index
Ivanov I.I.	MSU	25	70.5	175	120	200	230	7.801
Petrov P.P.	SPbSU	22	65.2	180	140	180	210	8.129
Sidorov S.S.	MIPT	18	55.8	165	90	150	180	7.527
Kuznetsov K.K.	HSE	20	75.1	185	200	220	240	8.788
Smirnov A.A.	MGU	27	90.3	190	180	220	250	7.198
Fedorov F.F.	NRU HSE	19	68.7	170	110	170	200	6.987
Volkov V.V.	BSU	21	60.0	160	80	0	0	1.333
Mikhailov M.M.	TSU	24	72.4	178	160	190	220	7.873
Novikov N.N.	ITMO	26	80.6	195	210	240	250	8.685
Morozov M.I.	RANEPA	23	73.8	183	170	200	230	8.130

!add

Enter data of the athlete in format:

name;university;age;weight;height;result1,result2,result3

Dmitriev D.S.;TomSU;23;67.8;172;150;190;220

Enter the number of the item indicated at the end

of the list before which you want to insert the athlete:

3

The item has been successfully inserted!

!print

Name	University	Age	Weight	Height	Res1	Res2	Res3	Index
Ivanov I.I.	MSU	25	70.5	175	120	200	230	7.801
Petrov P.P.	SPbSU	22	65.2	180	140	180	210	8.129
Sidorov S.S.	MIPT	18	55.8	165	90	150	180	7.527
Kuznetsov K.K.	HSE	20	75.1	185	200	220	240	8.788
Smirnov A.A.	MGU	27	90.3	190	180	220	250	7.198
Fedorov F.F.	NRU HSE	19	68.7	170	110	170	200	6.987
Volkov V.V.	BSU	21	60.0	160	80	0	0	1.333
Dmitriev D.S.	TomSU	23	67.8	172	150	190	220	8.260
Mikhailov M.M.	TSU	24	72.4	178	160	190	220	7.873
Novikov N.N.	ITMO	26	80.6	195	210	240	250	8.685
Morozov M.I.	RANEPA	23	73.8	183	170	200	230	8.130

!end

Goodbye!

Process finished with exit code 0

```

D:\VUZ\LAB_08_01\cmake-build-debug\LAB_08_01.exe
Please enter the file name:
input2.txt
Something went wrong!
Perhaps such a file does not exist.
Please enter the file name again:
input2.csv
The file has successfully been processed!
To display the data, enter the command "!print"
To find athletes, enter the command "!find"
To sort the data, enter the command "!sort"
To add new data, enter the command "!add"
To end the program, enter the command "!end"
!add
Enter data of the athlete in format:
name;university;age;weight;height;result1,result2,result3
Sorokin S.O.;MIIT;17;57.9;165;100;160;190
Enter the number of the item indicated at the end
of the list before which you want to insert the athlete:
15
The index is out of range!
The item will be inserted at the top of the list.
The item has been successfully inserted!
!print
+-----+-----+-----+-----+-----+-----+-----+-----+
| Name          | University | Age | Weight | Height | Res1 | Res2 | Res3 | Index |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Sorokin S.O.  | MIIT      | 17  | 57.9   | 165    | 100   | 160   | 190   | 7.772 |
| Kozlov K.V.   | UralSU    | 28  | 85.2   | 188    | 220   | 240   | 250   | 8.333 |
| Orlov O.D.    | MIIT      | 17  | 57.5   | 163    | 100   | 160   | 0      | 4.522 |
| Nikitin N.P.  | SFU       | 30  | 95.0   | 200    | 230   | 250   | 250   | 7.684 |
| Kovalev K.N.  | KPI       | 16  | 50.6   | 155    | 70    | 120   | 150   | 6.719 |
| Ilyin I.S.    | SPbPU     | 20  | 63.9   | 172    | 130   | 180   | 210   | 8.138 |
| Sergeev S.I.  | TomSU     | 22  | 67.2   | 168    | 0     | 0     | 0     | 0.000 |
| Stepanov S.V. | BSTU      | 18  | 58.4   | 166    | 120   | 170   | 200   | 8.390 |
| Gusev G.A.    | KAI       | 26  | 76.8   | 182    | 200   | 220   | 240   | 8.594 |
| Popov P.V.    | NSU       | 25  | 71.7   | 176    | 180   | 210   | 230   | 8.647 |
| Vasiliev V.S. | TSU       | 19  | 66.0   | 170    | 110   | 160   | 190   | 6.970 |
+-----+-----+-----+-----+-----+-----+-----+-----+
!end
Goodbye!

Process finished with exit code 0

```



```

D:\VUZ\LAB_08_01\cmake-build-debug\LAB_08_01.exe
Please enter the file name:
input3.csv
The file has successfully been processed!
To display the data, enter the command "!print"
To find athletes, enter the command "!find"
To sort the data, enter the command "!sort"
To add new data, enter the command "!add"
To end the program, enter the command "!end"
!add
Enter data of the athlete in format:
name;university;age;weight;height;result1,result2,result3
Sokolov S.V.;TSU;20;66.6;171;110;160;190
Enter the number of the item indicated at the end
of the list before which you want to insert the athlete:
4
The item has been successfully inserted!
!print
+-----+-----+-----+-----+-----+-----+-----+-----+
| Name           | University | Age | Weight | Height | Res1 | Res2 | Res3 | Index |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Pavlov P.A.    | MSU        | 24  | 70.2   | 177    | 160  | 200  | 230  | 8.405 |
| Kiselev K.A.   | SPbSU      | 20  | 65.8   | 175    | 140  | 180  | 210  | 8.055 |
| Bogdanov B.B.  | MIPT       | 18  | 55.5   | 162    | 90   | 150  | 180  | 7.568 |
| Danilov D.D.   | HSE        | 22  | 75.3   | 183    | 200  | 220  | 240  | 8.765 |
| Nesterov N.N.  | MGU        | 27  | 90.8   | 188    | 180  | 220  | 250  | 7.159 |
| Romanov R.I.   | NRU HSE    | 19  | 68.2   | 168    | 110  | 170  | 200  | 7.038 |
| Sokolov S.V.   | TSU        | 20  | 66.6   | 171    | 110  | 160  | 190  | 6.907 |
| Zaitsev Z.I.   | BSU        | 21  | 60.3   | 163    | 80   | 130  | 160  | 6.136 |
| Kononov K.M.   | TSU        | 25  | 72.6   | 179    | 120  | 190  | 220  | 7.300 |
| Frolov F.P.    | ITMO       | 26  | 101.2  | 192    | 210  | 240  | 250  | 6.917 |
| Belov B.V.     | UralSU     | 28  | 85.8   | 190    | 220  | 240  | 250  | 8.275 |
+-----+-----+-----+-----+-----+-----+-----+-----+
!add
Enter data of the athlete in format:
name;university;age;weight;height;result1,result2,result3
Vorobyov V.I.;MSU;28;86.2;186;220;240;250
Enter the number of the item indicated at the end
of the list before which you want to insert the athlete:
7
The item has been successfully inserted!
!print
+-----+-----+-----+-----+-----+-----+-----+-----+
| Name           | University | Age | Weight | Height | Res1 | Res2 | Res3 | Index |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Pavlov P.A.    | MSU        | 24  | 70.2   | 177    | 160  | 200  | 230  | 8.405 |
| Kiselev K.A.   | SPbSU      | 20  | 65.8   | 175    | 140  | 180  | 210  | 8.055 |
| Bogdanov B.B.  | MIPT       | 18  | 55.5   | 162    | 90   | 150  | 180  | 7.568 |
| Danilov D.D.   | HSE        | 22  | 75.3   | 183    | 200  | 220  | 240  | 8.765 |
| Vorobyov V.I.  | MSU        | 28  | 86.2   | 186    | 220  | 240  | 250  | 8.237 |
| Nesterov N.N.  | MGU        | 27  | 90.8   | 188    | 180  | 220  | 250  | 7.159 |
| Romanov R.I.   | NRU HSE    | 19  | 68.2   | 168    | 110  | 170  | 200  | 7.038 |
| Sokolov S.V.   | TSU        | 20  | 66.6   | 171    | 110  | 160  | 190  | 6.907 |
| Zaitsev Z.I.   | BSU        | 21  | 60.3   | 163    | 80   | 130  | 160  | 6.136 |
| Kononov K.M.   | TSU        | 25  | 72.6   | 179    | 120  | 190  | 220  | 7.300 |
| Frolov F.P.    | ITMO       | 26  | 101.2  | 192    | 210  | 240  | 250  | 6.917 |
| Belov B.V.     | UralSU     | 28  | 85.8   | 190    | 220  | 240  | 250  | 8.275 |
+-----+-----+-----+-----+-----+-----+-----+-----+
!end
Goodbye!

```


Выводы

В результате выполнения работы изучены особенности работы с линейными односвязными списками в языке Си. А также получены практические навыки в работе с указателями на структуры.