# 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

# 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  $fill\ struct()$ .

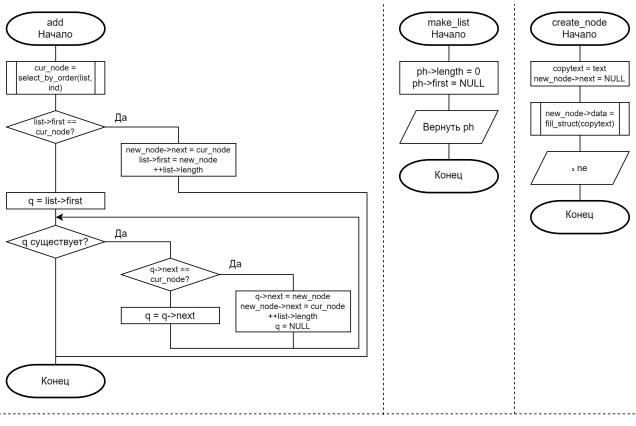
Функция *select\_by\_order()* перебирает все элементы списка до тех пор, пока не найдёт элемент с указанным индексом и возвращает указатель на его. Если элемента с таким индексом не существует, функция выведет соответствующее сообщение и вернёт указатель на первый элемент списка.

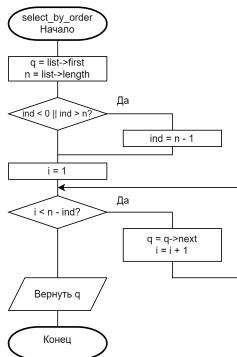
Функция *add()* вставляет в список элемент. Для этого строка преобразуется в элемент списка с помощью функции *create\_node()* и ищется элемент, перед которым нужно вставить, с помощью функции *select\_by\_order()*. В зависимости от положения элемента меняются указатели соседних элементов, тем самым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

# Описание переменных

№	Имя переменной	Тип	Назначение						
		Функция $Lis$	tOfAthlete *make_list()						
1	ph	ListOfAthlete *	Указатель на голову списка						
	Фун	кция NodeOfList	*create node(const char *text)						
1	text	char *	Исходная строка с данными						
2	new_node	NodeOfList *	Новый элемент списка						
3	copytext	char *	Копия исходной строки						
	Функция <i>Noc</i>	deOfList *select_b	y_order(const ListOfAthlete *list, int ind)						
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 voi	id add(ListOfAthle	ete *list, NodeOfList *new_node, int ind)						
1	list	ListOfAthlete *	Указатель на голову списка						
2	new_node	NodeOfList *	Элемент, который нужно вставить						
3	ind	int	Номер элемента, перед которым нужно						
3	mu	1111	вставить						
4	our node	NodeOfList *	Указатель на элемент, перед которым						
4	cur_node	NoucOILISt '	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;
}
               <del>++</del>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';
            str[tt] == '\0';</pre>
                             pole[ind++] = word;
word = str + tt + 1;
                      }
               str[tt] = '\0';
               user->name = pole[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()
      <u>ListOfAthlete *ph = NULL;</u>
```

```
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));
         (new node && copytext) {
          stropy (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;
          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);</pre>
     printf("+----+
           "+----+\n");
void pprint(const ListOfAthlete *list) {
     NodeOfList *cur node = list->first
```

```
| University | Age | Weight | Height "
     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 *));
    mas = (NOULL) {
   for (int i = 0; cur node != NULL; ++i) {
      mas[i] = cur node;
      mas[i] = cur node->next;
          }
     return mas:
void my swap(NodeOfList **mas, ListOfAthlete *list, int i, int j) {
    NodeOfList *q;
          (i == 0) {
list->first = mas[j];
     } else {
         mas[i - 1] \rightarrow 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;
    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;
    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) {
                     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) ||
               if (mas[i] = 1)
                            (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;
                             if care in the state in th
                                                           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"
                                            "| Name | University | Age | Weight "
"| Height | Res1 | Res2 | Res3 | Index |\n");
print line();
sorted(mas, list, ch);
print line();
lse {
                                             } 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:
              "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 \"!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");
      scanf("%i", &ch);
if (ch < 1 || 2 < ch) {
   printf("Invalid command!\n");
} else {
   find(list, ch);</pre>
       } else if (!stromp(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 (!stromp(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 (stramp(str, "!end") != 0);
free list(list);
return 0;
```

#### Контрольные примеры

No	Исходные	Результаты
312	данные	тезультаты

	input1.csv !print !add Dmitriev D.S.;TomSU;23;	Please enter the file no. The file has successful To display the data, enter To sort the data, enter To add new data, enter To end the program, enter To add the program, enter To end the program to the Total	ly been proce ter the comma the command the command " er the comman	nd "!pi "!find "!sort !add" d "!end	" d" +					
1	67.8;172;150;190 ;220 3 !print !end	Ivanov I.I.   Petrov P.P.   Sidorov S.S.   Kuznetsov K.K.   Smirnov A.A.   Fedorov F.F.   Volkov V.V.   Mikhailov M.M.	MSU   SPbSU   MIPT   HSE   MGU   NRU HSE   BSU   TSU   ITMO   RANEPA + te in format: ght;height;re item indicat	25   22   18   20   27   19   21   24   26   23   25	70.5   65.2   55.8   75.1   90.3   68.7   60.0   72.4   80.6   73.8	175   180   165   185   195   170   170   178   178   183	120   140   90   200   180   110   80   160   210   170	200   180   150   220   220   170   0   190   240   200	230   210   180   240   250   200   0   220   250   230	7.801     8.129     7.527     8.788     7.198     6.987     1.333     7.873     8.685     8.130
		Ivanov I.I.   Petrov P.P.   Sidorov S.S.   Kuznetsov K.K.   Smirnov A.A.   Fedorov F.F.   Volkov V.V.   Dmitriev D.S.   Mikhailov M.M.	University   MSU   SPbSU   MIPT   HSE   MGU   NRU HSE   BSU   TomSU   TSU   ITMO   RANEPA	+	Weight   70.5   65.2   55.8   75.1   90.3   68.7   60.0   67.8   72.4   80.6   73.8	Height   175   180   165   185   190   170   160   172   178   195   183	Res1   120   140   90   200   180   110   80   150   160   210   170	Res2   200   180   150   220   220   170   0   190   190   240   200	Res3	Index   
2	input2.txt input2.csv !add Sorokin S.O.;MIIT;17;57. 9;165;100;160;1 90 15 !print !end	Sorokin S.O.   Kozlov K.V.   Orlov O.D.   Nikitin N.P.   Kovalev K.N.   Ilyin I.S.   Sergeev S.I.   Stepanov S.V.   Gusev G.A.   Popov P.V.	s not exist. ame again: ly been proce ter the command the command ' the command ' the command the command the command to it format: ght;height;re item indicat h you want to ge! ed at the top ssfully inser ' University ' University ' KAI   SPBPU   TomSU   BSTU   KAI   NSU   TSU	nd "!p "!find "!sort !add" d "!en sult1, ed at inser  of th ted! +   Age +   17   28   17   30   16   20   22   18   26   25   19	"" result2,r the end t the ath e list.    Weight   57.9   85.2   57.5   95.0   63.9   67.2   58.4   76.8   71.7	Height He	Res1   100   220   100   230   70   130   0   120   200   180   110	Res2   160   240   160   250   120   180   0   170   220   210   160	Res3   190   250   0   250   150   210   0   200   240   230   190	Index

Please enter the file name: input3.csv The file has successfully been processed! !add To display the data, enter the command "!print" To find athletes, enter the command "!find" Sokolov To sort the data, enter the command "!sort" To add new data, enter the command "!add" S.V.;TSU;20;66. To end the program, enter the command "!end" Enter data of the athlete in format: 6;171;110;160;1 name;university;age;weight;height;result1,result2,result3 90 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 | University | Age | Weight | Height | Res1 | Res2 | Res3 | Index | !add | MSU | Pavlov P.A. | 24 | 70.2 | 177 | 160 | 200 | 230 | 8.405 | Vorobyov | 140 | 180 | 210 | 8.055 | | Kiselev K.A. l SPbSU | 20 | 65.8 l 175 | Bogdanov B.B. | MIPT | 18 | 55.5 | 162 | 90 | 150 | 180 | 7.568 | V.I.;MSU;28;86. | 200 | 220 | 240 | 8.765 | | HSE | 22 | 75.3 | 183 | Danilov D.D. | Nesterov N.N. l MGU | 27 | 90.8 188 | 180 | 220 | 250 | 7.159 | 2;186;220;240;2 | Romanov R.I. | NRU HSE | 19 | 68.2 | 168 | 110 | 170 | 200 | 7.038 | 50 | TSU | 110 | 160 | 190 | 6.907 | | Sokolov S.V. | 20 | 66.6 | 171 | Zaitsev Z.I. | BSU | 21 | 60.3 | 163 | 80 | 130 | 160 | TSU | 25 | 72.6 | 179 | 120 | 190 | 220 | 7.300 | | Konovalov K.M. | 210 | 240 | 250 | 6.917 | 3 I Frolov F.P. I ITMO | 26 | 101.2 | 192 !print | Belov B.V. | UralSU | 28 | 85.8 | 190 | 220 | 240 | 250 | 8.275 | !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! | University | Age | Weight | Height | Res1 | Res2 | Res3 | Index | | 24 | 70.2 | 177 | 160 | 200 | 230 | 8.405 | | Kiselev K.A. | SPbSU | 20 | 65.8 | 175 | 140 | 180 | 210 | 8.055 | | 90 | 150 | 180 | 7.568 | | MIPT | 18 | 55.5 | 162 l Bogdanov B.B. | HSE | 22 | 75.3 | 200 | 220 | Danilov D.D. | 183 240 8.765 | MSU | Vorobyov V.I. | 28 | 86.2 | 186 | 220 | 240 | 250 | 8.237 | | MGU | 180 | 220 | 250 | 7.159 | | Nesterov N.N. | 27 | 90.8 | 188 Romanov R.I. | NRU HSE | 19 | 68.2 | 168 | 110 | 170 | 200 | 110 | 160 | 190 | 6.907 | | Sokolov S.V. l TSU | 20 | 66.6 | 171 | 80 | 130 | 160 | 6.136 | | Zaitsev Z.I. l BSU | 21 | 60.3 | 163 | Konovalov K.M. | TSU | 25 | 72.6 | 179 | 120 | 190 | 220 | ITMO | 210 | 240 | 250 | 6.917 | | 26 | 101.2 | 192 | Frolov F.P. | 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 Kuznetsov K.K.; HSE; 20; 75.1; 185; 200; 220; 240 4 5 Smirnov A.A.; MGU; 27; 90.3; 190; 180; 220; 250 Fedorov F.F.; NRU HSE; 19; 68.7; 170; 110; 170; 200 6 7 Volkov V.V.; BSU; 21; 60.0; 160; 80; 0; 0 Mikhailov M.M.; TSU; 24; 72.4; 178; 160; 190; 220 8 9 Novikov N.N.;ITM0;26;80.6;195;210;240;250 Morozov M.I.; RANEPA; 23; 73.8; 183; 170; 200; 230 10

#### input2.csv ×

- 1 Kozlov K.V.; UralSU; 28; 85.2; 188; 220; 240; 250
- 2 Orlov 0.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 Konovalov K.M.; TSU; 25; 72.6; 179; 120; 190; 220
- 9 Frolov F.P.;ITM0;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			Ċ		İ	Height	I	Res1	I	Res2	ĺ	Res3	ĺ	Index	Ī
Ivanov	I.I.	, - 	MSU	Ī	25	I	70.5		175	Ċ	120	1	200	Ċ	230	÷	7.801	
Petrov	P.P.	İ	SPbSU	i	22	i	65.2	i	180	i	140	i	180	i	210	i	8.129	i
Sidoro	v S.S.	l	MIPT	Ī	18	I	55.8	Ī	165	Ī	90	Ī	150	Ī	180	Ī	7.527	Τ
Kuznet	sov K.K.	l	HSE	Ī	20		75.1	I	185	Ī	200	Ī	220	Ī	240	I	8.788	Ι
Smirno	v A.A.		MGU	Ī	27	1	90.3	Ī	190	Ī	180	1	220	Ī	250	Ī	7.198	Ι
Fedoro	v F.F.		NRU HSE	I	19		68.7	I	170	I	110	1	170	I	200	I	6.987	1
Volkov	V.V.		BSU	I	21	I	60.0	I	160	1	80	1	0	1	0	I	1.333	1
Mikhai	lov M.M.		TSU	I	24		72.4	I	178	I	160	1	190	1	220	I	7.873	$\mathbf{I}$
Noviko	v N.N.		ITMO	I	26		80.6	I	195		210	1	240	1	250	I	8.685	1
Morozo	v M.I.	l	RANEPA	I	23		73.8	I	183	I	170	1	200	I	230	I	8.130	1
+		+-		+-		+-		+-		+		+-		+-		+-		+-

!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			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

+	-+-		- +		-+		- 4		. +		- +		- +		4		- +
Name	 -+-	University	İ	Age	İ	Weight		-	-	Res1					-	Index	
									Ċ								
Sorokin S.O.		MIIT	ı	17		57.9		165	ı	100		160	ı	190		7.772	
Kozlov K.V.	-	UralSU		28	-	85.2	1	188	1	220	1	240	1	250		8.333	
Orlov O.D.	-	MIIT		17	-	57.5	1	163	1	100	1	160	1	0		4.522	-
Nikitin N.P.	1	SFU	1	30		95.0	1	200	1	230	1	250	1	250		7.684	1
Kovalev K.N.	-	KPI	١	16	-	50.6	1	155	1	70	1	120	1	150	1	6.719	1
Ilyin I.S.	1	SPbPU	1	20	-	63.9	1	172	1	130	1	180	1	210	1	8.138	1
Sergeev S.I.	1	TomSU	١	22	-	67.2	1	168	1	0	1	0	1	0	1	0.000	1
Stepanov S.V.	1	BSTU	1	18	1	58.4	1	166	1	120	1	170	1	200	1	8.390	1
Gusev G.A.	1	KAI	1	26	1	76.8	1	182	1	200	1	220	1	240	1	8.594	1
Popov P.V.	1	NSU	I	25	1	71.7	1	176	1	180	1	210	1	230	1	8.647	1
Vasiliev V.S.	1	TSU	١	19	1	66.0	1	170	1	110	1	160	1	190	1	6.970	1
+	-+-		+		-+		- + -		+		-+-		+		4.		-+

!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

_								<b>.</b> .		ъ.		ъ.		ъ.		ъ.		
	Name		University		Age			Ċ		Ċ		÷		Ċ		Ċ		Ċ
T						T.						,						
١	Pavlov P.A.	l	MSU	ı	24	ı	70.2	ı	177	ı	160	ı	200	ı	230	ı	8.405	ı
	Kiselev K.A.		SPbSU		20		65.8	-	175		140		180		210		8.055	
-	Bogdanov B.B.		MIPT	I	18	1	55.5	1	162	1	90	1	150	1	180	1	7.568	1
-	Danilov D.D.		HSE	I	22	1	75.3		183	1	200	1	220	1	240	1	8.765	1
-	Nesterov N.N.		MGU	I	27	1	90.8	1	188	I	180	1	220	1	250	1	7.159	
-	Romanov R.I.		NRU HSE	I	19	1	68.2	1	168	1	110	1	170	1	200	1	7.038	1
-	Sokolov S.V.	l	TSU	Ī	20	1	66.6	1	171	Ī	110	1	160	1	190	1	6.907	
-	Zaitsev Z.I.	l	BSU	Ī	21	1	60.3	-	163	Ī	80	1	130	I	160	I	6.136	1
-	Konovalov K.M.	l	TSU	Ī	25	1	72.6	1	179	Ī	120	1	190	1	220	1	7.300	
-	Frolov F.P.	I	ITMO	Ī	26	1	101.2		192	Ī	210	1	240	1	250	I	6.917	1
-	Belov B.V.	I	UralSU	I	28	1	85.8	I	190	I	220	1	240	I	250	I	8.275	I
+		+-		+-		- +		- 4 -		4.		. 4 .		. +		. +		- +

! 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		+   Height	: :			
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
Konovalov 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.