

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет Информационных Технологий

Кафедра прикладной математики

Отчет защищен с оценкой _____

Преподаватель _____ Н. Д. Бубнова

«___» _____ 2015 г.

Отчет
по лабораторной работе №3

III семестр

по дисциплине «Введение в алгоритмы и основы
технологий разработки программ»

Студент группы ПИ-41 _____ В. О. Цисык
Преподаватель _____ Н. Д. Бубнова

БАРНАУЛ 2015

Задание:

Задан список чисел. Образовать из него новый, исключив из исходного минимальное число элементов так, чтобы список стал неубывающим

Описание алгоритма:

Программа начинает свое выполнение с того, что запрашивает у пользователя файл с исходными данными. Если файл с заданным именем существует, то создается пустой список, в который по очереди записываются исходные данные. Список с исходными данными передается в функцию `list_LIS()`.

Если список пуст (файл был пустым), то функция возвращает пустой список. Иначе, просматривается весь список. Во вспомогательный массив `len` записывается длина длиннейшей неубывающей подпоследовательности в списке до определенного элемента, `len[i]` - длина подпоследовательности на *i*-ом элементе списка. В массив `prev` записываются "позиции" элементов, при которых подпоследовательность имеет наибольшую длину.

После обработки списка, мы находим "элемент при котором длина подпоследовательности - наибольшая. По массиву `prev` восстанавливаем элементы и записываем их в список. Возвращаем список в вызывающую функцию. Ответ получен. Результат записывается в файл.

Код программы:

```
/* lab3.c */
/* Задан список чисел. Образовать из него новый, исключив из исходного
   минимальное число элементов так, чтобы список стал неубывающим */

#include <stdio.h>
#include "list.h"
#define LINELENGTH 1024
int main()
{
    FILE *fp;
    char line[LINELENGTH];
    char *p;
    int c, n;
    struct List *test = NULL;
    struct List *answer = NULL;
    test = list_init();
    printf("введите имя файла:");
    scanf("%s", line);
    if((fp = fopen(line, "r")) == NULL){
        fprintf(stderr, "Нет такого файла\n");
        return 1;
    }
}
```

```

/* читаем строку за строкой */
    p = line;
while(fgets (line, LINELENGTH, fp)){
    while(sscanf(p, " %d%n", &c, &n) == 1 ){
        list_insert(test, c);
        p +=n;
    }
}

answer = list_LIS(test);
fclose(fp);

while ((c = getchar())!= '\n' && c != EOF);/* отсеять всякий мусор */
printf("Сохранить в файл? [y/n]:");
scanf("%c", &c);/* пропустить пробелы и получить ответ */
switch (c){
case 'y': case 'Y':
printf("Записано в \"output.txt\\n\\n");
fp = fopen("output.txt", "a");
fprintf(fp, "-----\\n");
    while(answer->head != NULL){
        fprintf(fp, "%d ", answer->head->number);
        answer->head = answer->head->next;
    }
fprintf(fp, "\\n");
fclose(fp);
break;
}

return 0;
}

```

```

/* list.c */
#include <stdlib.h>
#include <assert.h>
#include <stdio.h>
#include "list.h"

/* Создаем новый список */
struct List *list_init(void)
{
    struct List* theList = malloc(sizeof(struct List));
    assert(theList != NULL);

    theList->head = NULL;
    theList->tail = NULL;
    return theList;
}

```

```

/* Добавляем в конец списка */
void list_insert(struct List *test, int number)

```

```

{
    struct Node *newNode = malloc(sizeof(struct Node));
    assert(newNode != NULL);
    newNode->number = number;
    newNode->next = NULL;
    if(test->head == NULL)
        test->head = test->tail = newNode;
    else{
        test->tail->next = newNode;
        test->tail = newNode;
    }
}

/* Добавляем в начало списка */
void list_push(struct List *test, int number)
{
    struct Node *newNode = malloc(sizeof(struct Node));
    newNode->number = number;
    newNode->next = NULL;
    if(test->head == NULL)
        test->head = test->tail = newNode;
    else{
        newNode->next = test->head;
        test->head = newNode;
    }
}

/* Обрабатываем список */
struct List *list_LIS(struct List *test)
{
    int *prev = NULL;
    int *len = NULL;
    struct Node *tmp = test->head;
    struct Node *first = NULL;
    struct List *answer = NULL;
    answer = list_init();
    int i, j, n;
    int pos, length;
    length = i = 0;

    /* пуст ли список? */

    if(tmp == NULL)
        return answer;
    while(tmp != NULL){

        /* выделить память под очередные элементы массива*/
        prev = realloc(prev, (++length) * sizeof(int));
        len = realloc(len, (++length) * sizeof(int));
        /* запомнить позицию головы */
        first = test->head;
        prev[i] = -1;

```

```

        len[i] = j = 0;

        /* находим количество элементов меньше определенного */
        while(first != tmp){
            if(first->number <= tmp->number &&
               len[j] + 1 > len[i]){
                len[i] = len[j] + 1;
                prev[i] = j;
            }
            first = first->next;
            j++;
        }
        /* printf("%d --- %d --- %d\n", len[i], tmp->number, prev[i]);*/
        tmp = tmp->next;
        i++;
    }
    /* находим самую длинную последовательность чисел... */
    n = length;
    pos = 0;
    length = len[0];
    for(i = 0; i < n; i++){
        if(len[i] > length){
            pos = i;
            length = len[i];
        }
    }

    tmp = test->head;
    i = 0;
    /* ...и пишем ее в список */
    while(pos != -1){
        for(i = 0; i < pos; i++){
            tmp = tmp->next;
        }
        list_push(answer, tmp->number);
        pos = prev[pos];
        tmp = test->head;
    }

    return answer;
}

/* list.h */

struct Node{
    int number;
    struct Node *next;
};

struct List{
    struct Node *head;
    struct Node *tail;
};

```

```
struct List *list_init(void);  
void list_insert(struct List *test, int number);  
  
struct List *list_LIS(struct List *test);  
void list_push(struct List *test, int number);
```

Результаты тестирования:

```
Лабораторная работа №3 : bash - Konsole
File Edit View Bookmarks Settings Help
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ ./lab3
Введите имя файла:test.txt
Сохранить в файл? [y/n]:y
Запущено в "output.txt"
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ cat output.txt
-----
2
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ cat test.txt
2
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ ./lab3
Введите имя файла:test.txt
Сохранить в файл? [y/n]:y
Запущено в "output.txt"
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ cat output.txt
-----
2
-----
2 2 2 2 4 4 4 6
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ cat test.txt
2 2 4 2 8 1 2 4 2 9 4 2 4 6 3 5
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ ./lab3
Введите имя файла:test.txt
Сохранить в файл? [y/n]:y
Запущено в "output.txt"
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ cat out.put.txt
cat: out.put.txt: Нет такого файла или каталога
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ cat output.txt
-----
2
-----
2 2 2 2 4 4 4 6
-----

vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ cat test.txt
vlad@desktop:~/AltSTU/Курс 2/Введение в алгоритмы и основы технолог
разработ пр/Лабораторная работа №3$ █
```