

Национальный исследовательский ядерный университет
«МИФИ» Институт интеллектуальных кибернетических
систем

Кафедра №12 «Компьютерные системы и технологии»

ОТЧЕТ

О выполнении лабораторной работы №3

«Работа с массивами данных»

Студент: Титов Иван Андреевич.

Преподаватель: Уваров М.П.

Группа: Б23-901

Москва – г .2023

І.Условие

Вариант № 14

Введение

Необходимо спроектировать и реализовать на языке C программу, осуществляющую по запросам пользователя ввод, обработку и вывод последовательности данных, которая представляется в виде массива.

Программа должна реализовывать следующую функциональность:

1. Инициализация массива (заполнение массива в цикле).
2. Вставка нового элемента в массив по индексу.
3. Удаление элемента массива по индексу.
4. Обработка данных (в соответствии с индивидуальным заданием).
5. Вывод текущего состояния массива.

Примечания:

1. Взаимодействие программы с пользователем должно быть выстроено с помощью диалогового меню, имеющего приблизительно следующий вид:
 - (a) Инициализация массива.
 - (b) Вставка нового элемента.
 - (c) Удаление элемента.
 - (d) Индивидуальное задание.
 - (e) Вывод содержимого массива.
 2. После выполнения любой из представленных операций программа должна автоматически выводить содержимое массива.
 3. Программа должна штатным образом завершаться при выборе пользователем соответствующего пункта диалогового меню или при обнаружении EOF — конца файла (в UNIX-подобных ОС инициируется нажатием клавиш `Ctrl + D`, в Windows — `Ctrl + Z`).
 4. Программа должна осуществлять проверку корректности вводимых данных и, в случае ошибок, выдавать соответствующие сообщения, после чего продолжать работу.
 5. Обработываемые последовательности должны быть представлены в виде массива элементов, которые имеют соответствующий тип данных.
-

6. Память под массивы обрабатываемых данных должна выделяться и освобождаться динамически, с использованием функций `calloc()`, `malloc()`, `realloc()` и `free()` из состава стандартной библиотеки.
7. Использование VLA (массивов переменной длины) не допускается.
8. При вставке элемента в массив по указанному индексу, элементы массива с индексом, превышающим или совпадающим с заданным, должны «сдвигаться вправо».
9. При удалении элемента массива с указанным индексом, элементы массива с большим индексом должны «сдвигаться влево».
10. При вставке в массив нового элемента по индексу, значение которого превышает максимально допустимое (оно соответствует длине массива), необходимо осуществлять вставку в конец массива.
11. При удалении элемента массива по индексу, значение которого превышает максимально допустимое (оно соответствует длине массива), необходимо вернуть ошибку.
12. Логически законченные части алгоритма решения задачи должны быть оформлены в виде отдельных функций с параметрами. Использование глобальных переменных не допускается.
13. Функции по обработке массивов не должны быть диалоговыми, т. е. они должны принимать все необходимые данные в качестве параметров и возвращать результат работы в виде соответствующих данных и кодов ошибок (исключения: функции инициализации и вывода массивов).
14. Исходные коды программы должны быть логичным образом разбиты на несколько файлов (необходимо использовать как *.c-файлы, так и *.h-файлы).
15. Согласно условиям индивидуального задания, может требоваться наличие нескольких исходных последовательностей. В таком случае, пользователь должен иметь возможность интерактивного взаимодействия с каждой из них.
16. Программа должна корректным образом работать с памятью, для проверки необходимо использовать соответствующие программные средства, например: `valgrind` (при тестировании и отладке программы необходимо запускать её командой вида `valgrind ./lab3`).

Индивидуальное задание

В исходной последовательности вещественных чисел найти те, старшая значащая цифра в которых равна 9. Сформировать из данных чисел новую последовательности, удалив их из исходной.

Правила изменения размера выделенной под массив области памяти

Размер выделенной под массив области памяти задаётся пользователем вручную через соответствующие пункты меню.

Любые операции по изменению размера выделенной области памяти выполняются только по явному запросу со стороны пользователя.

Цель: создать программу, которая удовлетворяет требованиям.

Требования:

1. Выводить в терминал меню, отображающее функциональность программы. Реализовать взаимосвязи функций, соответствующих пунктам меню.
2. Принимать на вход значения, соответствующие выбранным пунктам меню, а также значения, необходимые для реализации функционала функций (пунктов меню)

3. Реализовать необходимые функции в соответствии с предлагаемыми алгоритмами их реализации (к примеру, удаление – сдвигом элементов массива влево, вставка элемента массива – сдвигом элементов массива вправо)
4. Программа должна быть разбита на файлы с исходным кодом (.c) и заголовочные (.h)
5. Работа с памятью пользователем осуществляется непосредственно, т.е. изменение размера массива должно происходить вручную, в явном виде (как пункт меню)

Примечания к решению:

1. Функции, осуществляющие безопасный ввод, разбиты на 2 составляющие, реализующие каждая свой функционал:
 - а) InputD() – функция, осуществляющая ввод вещественного числа (типа данных double) (в рамках данной программы это является заполнением массива элементами).
 - б) InputST3() – функция, осуществляющая ввод чисел типа данных size_t, поддерживающая ввод до 3 цифр. Необходима для ввода пользователем величин, определяющих размер массива (переменная s), размер дополнения массива ячейками (переменная n), номер индекса (переменная ind).Аналогичные функции были реализованы в предыдущей лабораторной работе, поэтому здесь их коды приводиться не будут. Также стоит отметить, что используемая в InputD и InputST3() функция посимвольного ввода getch(), подключаемая отдельно «динамической библиотекой» «conio.c» (которая является копией <conio.h> из windows в unix-подобных системах) применяется в программе для выбора пунктов меню.
2. Работа с памятью в программе осуществляется следующим образом: задаётся размер массива, относительно которого и выделяется память для массива функцией malloc (учитывая размер типа данных double). Инициализация массива организована двумя способами: вручную и автоматизированных (генерация псевдослучайных значений). В любом из способов изначально задаётся размер массива в качестве переменной s. Ещё одна переменная – f. Она задаёт фактический размер массива, т.е. количество ячеек с используемыми значениями (в данном случае – вещественными числами). «Пустые» ячейки не учитываются. Таким образом разность s-f означает количество «пустых» (опять же, пустые лишь условно, так как они всегда заполнены нулевыми значениями) ячеек.

II. Алгоритмическая часть

Программа представляет из себя многофайловый проект. При этом можно выделить две группы: ранее освоенные функции безопасного ввода (InputST3, InputD), аналогичные тем, что использовались в предыдущей лабораторной работе, новые функции в соответствующих им исходных файлах (Out.c, Del.c, BigDel.c, Paste.c, Input.c, InRan.c) и заголовочный файл (lab3H.h). Функционал функций (и соответственно их исходных файлов) следующий:

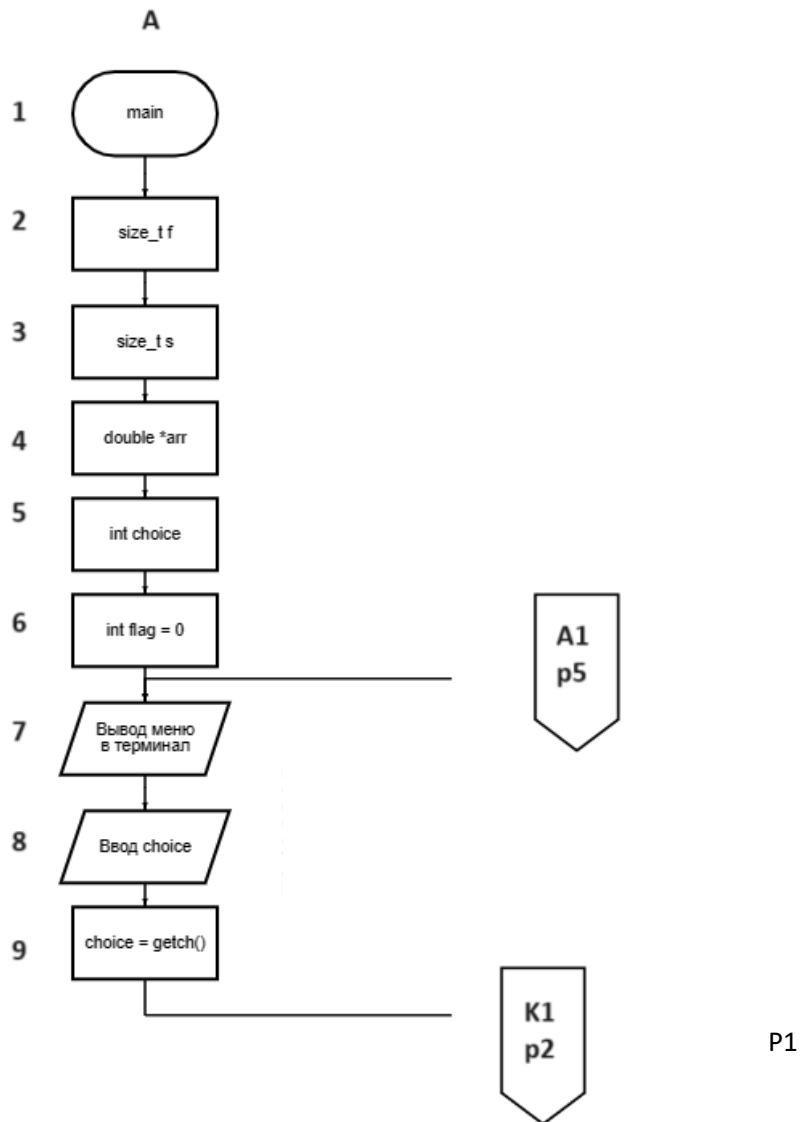
1. Out.c – вывод массива в терминал. На вход подаётся массив и его фактический размер. Соответственно выводятся только непустые значения.
2. Del.c – удаление числа из массива по индексу. На вход подаётся массив, индекс числа (ind) и фактический размер массива. Все числа в массиве, начиная с элемента под номером ind сдвигаются влево. Последняя освобождённая ячейка заполняется значением «0».
3. BigDel.c – реализация индивидуального задания – удаление из массива всех чисел со старшей значащей цифрой «9». На вход подаются массив, фактический размер массива (f) и полный размер массива (s). В процессе выполнения функции создаются два массива part1 (без чисел со значащей цифрой «9») и part2 (оставшиеся числа). Массив part2 выводится в терминал, но далее не используется. В теле функции main() основному используемому массиву arr присуждается значение массива part1.
4. Paste.c – вставка числа по индексу. На вход подаются массив и его фактический размер (f). Если массив достаточно велик, чтобы вместить ещё одно число, то все числа, начиная с f-ого приобретают значения предыдущего члена массива (т.е. сдвигаются вправо). Вставляемый элемент вводится вручную (функцией InputD()), как и его индекс в массиве.
5. Input.c – инициализация массива набором пользователем вручную. Функция принимает на вход указатель на переменную flag (которая обозначает, инициализирован массив или нет), а также полный размер массива (s).
6. InRan.c – инициализация массива путём генерации псевдослучайных чисел. Функция принимает на вход указатель на переменную flag (которая обозначает, инициализирован массив или нет), а также полный размер массива (s). Диапазон генерации обеспечивает числа с 3 значащими цифрами перед точкой и 3 цифрами – после.

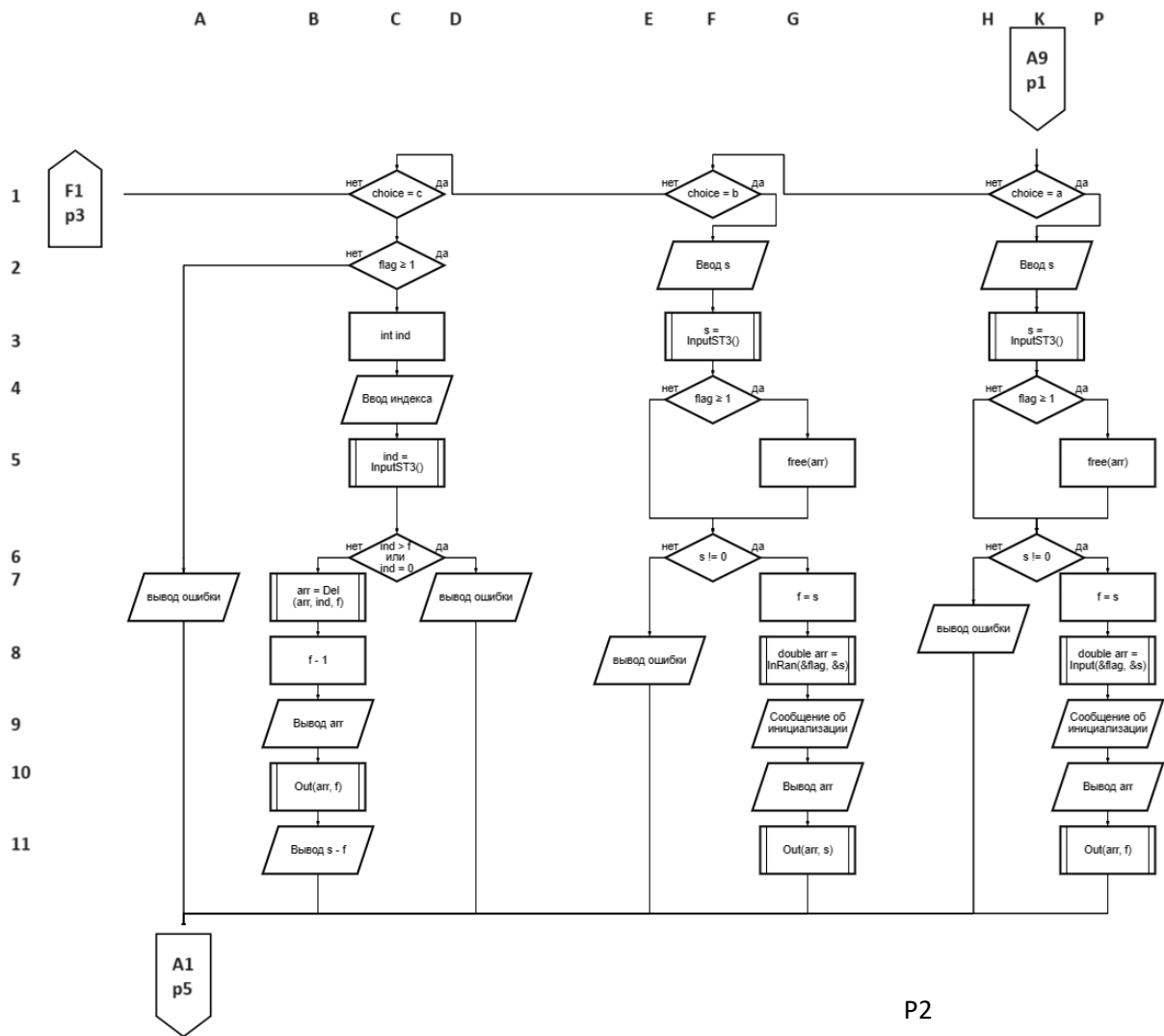
Переменные `element` (элемент для вставки), `arg` имеют тип данных `double`, так как по индивидуальному заданию они должны быть вещественными числами.

Переменные `ind` (индекс элемента для вставки/удаления), `s` (полный размер массива), `f` (фактический размер массива), `n` (количество ячеек для дополнения массива) имеют тип данных `size_t`, так как они определяют размер массива или положение в нём.

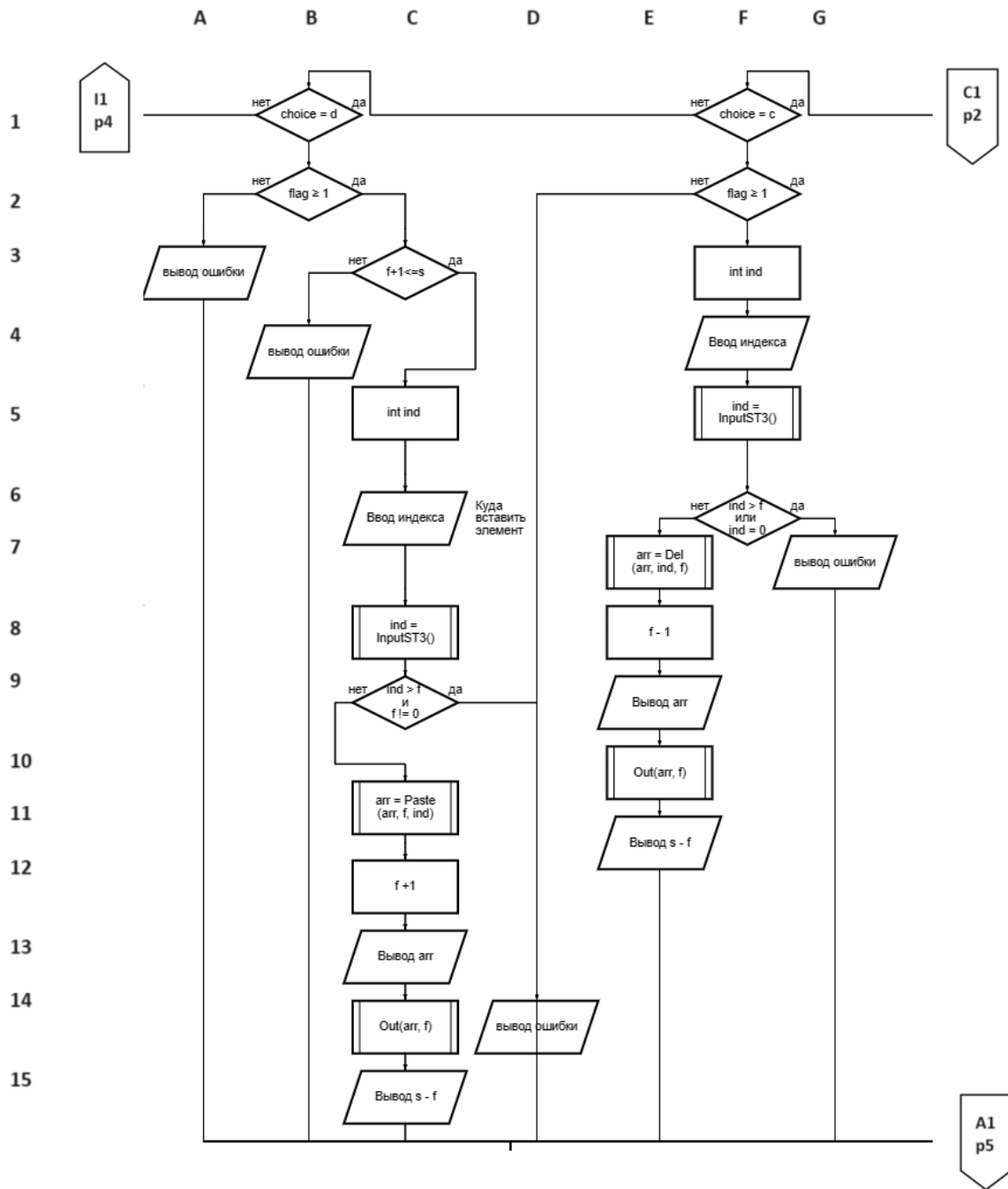
Блок-схема

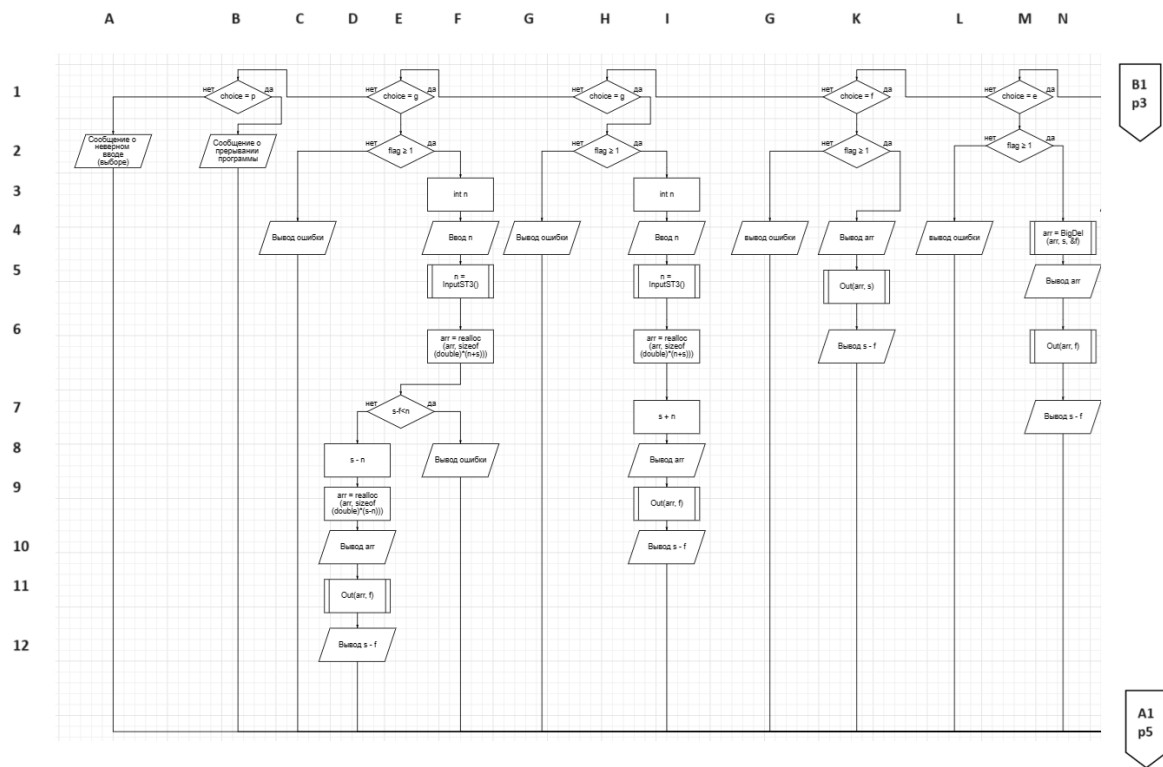
1. Main()



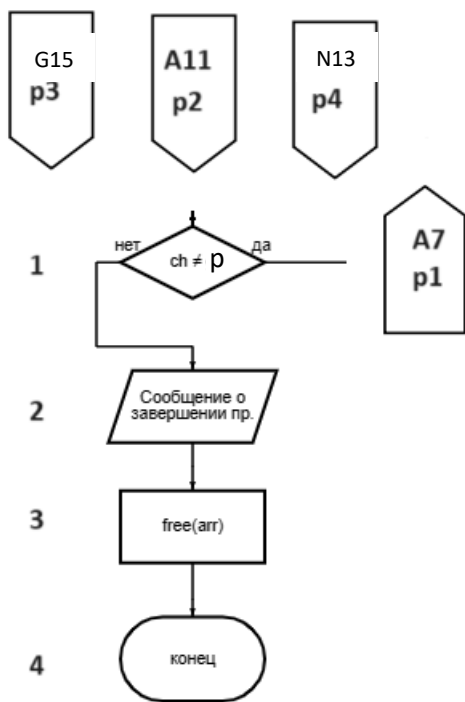


P2

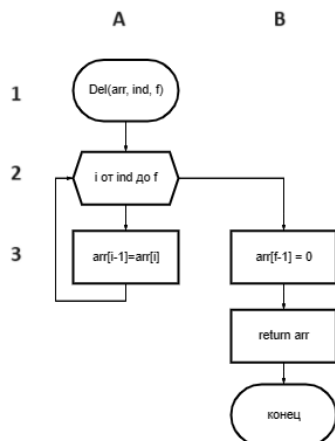




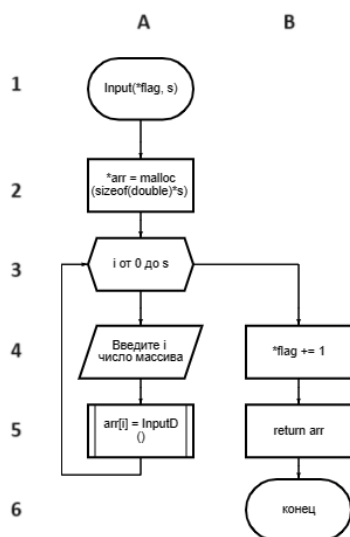
A



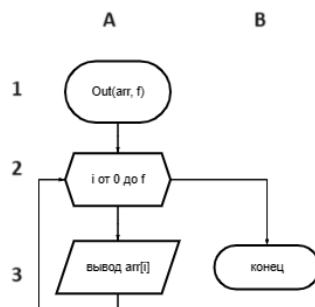
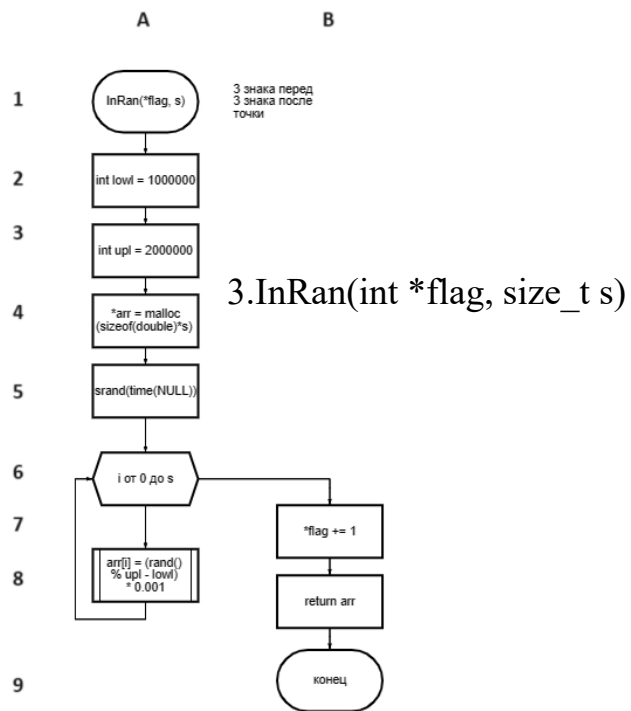
p5



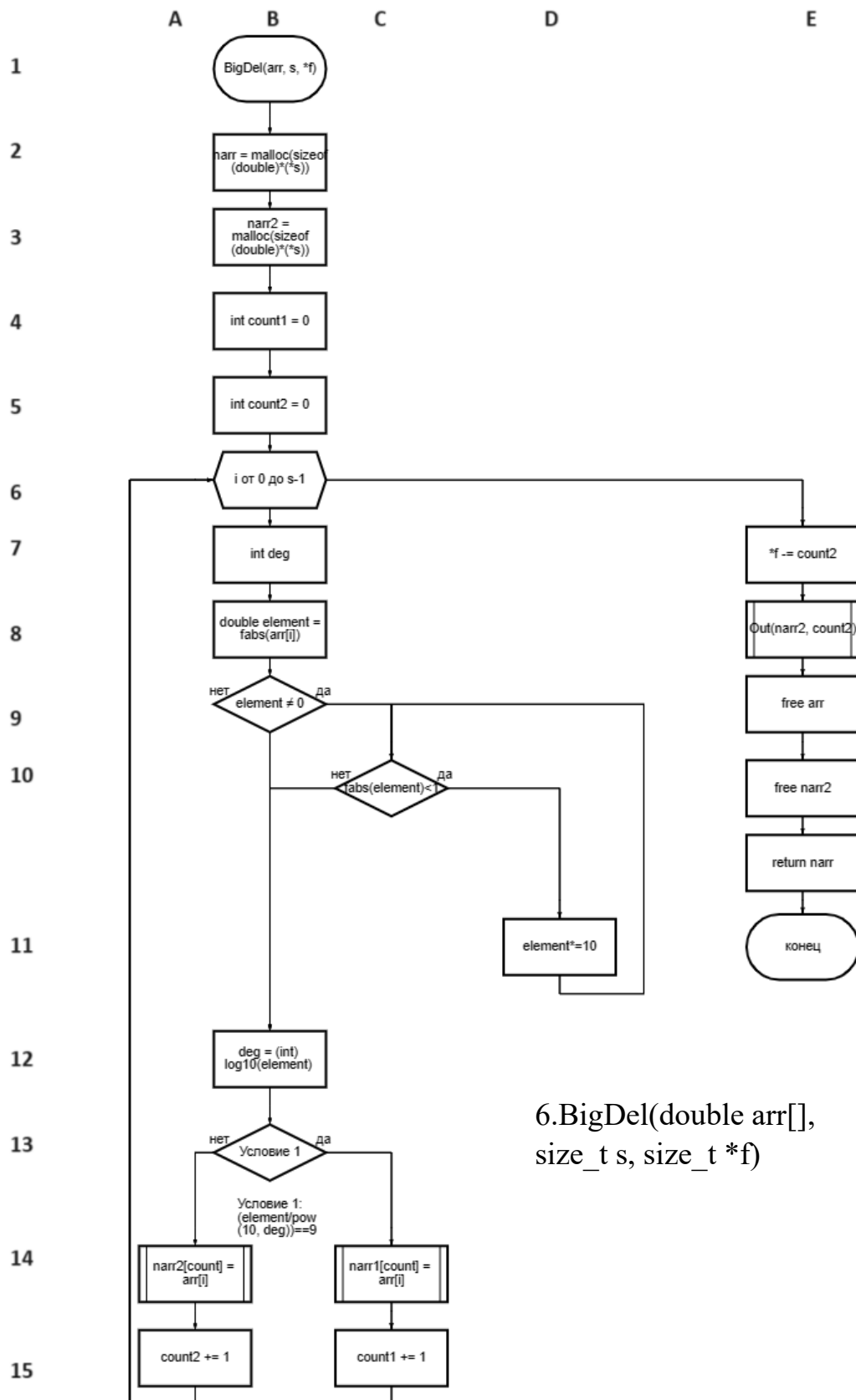
2.Del(double arr[], size_t ind, size_t f)



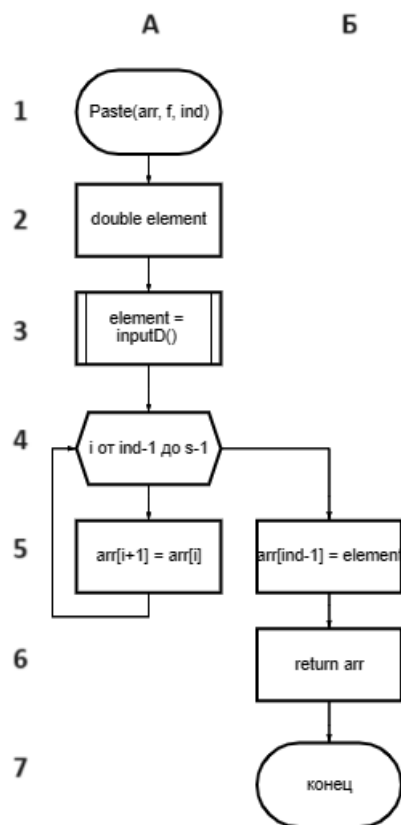
4.Input(int *flag, size_t s)



5.Out(double arr[], size_t f)



6. `BigDel(double arr[],
size_t s, size_t *f)`



7.Paste(double arr[], size_t f)

III.Код

1. lab3H.h

```
#ifndef LAB3_H
#define LAB3_H
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
#define Esc 27
#define Enter 10
#define BackSp 127
#define minus 45
#define dot 46
#define zero 48
#define nine 57
#define one 49
#define BackSp 127

int getch(void);

double InputD();

size_t InputST3();

int Out(double arr[], size_t s);

double* Del(double arr[], size_t ind, size_t f);

double* BigDel(double arr[], size_t s, size_t *f);

double* Paste(double arr[], size_t f);

double* InRan(int *flag, size_t s);

double* Input(int *flag, size_t s);

#endif
```

2. lab3.c

```
#include "lab3H.h"
int main(){
    size_t s;
    size_t f;
    double *arr;
    char choice;
    int flag = 0;
    do{
        printf("\nВведите символ, чтобы выбрать соответствующий пункт меню\n");
        printf("a) - ввод массива\nb) - создание случайного массива\nc) - удаление элемента массива\n");
        printf("d) - вставка элемента массива по индексу\ne) - удаление всех элементов со старшей значимой цифрой 9\n");
        printf("f) - вывод массива\ng) - добавить памяти\nh) - выход\n");
        choice = getch();
        switch (choice){
            case 'a': //Ввод массива с терминала
```

```

число элементов массива\n");
printf("Инициализирован ввод массива\nВведите
s = InputST3();
if (flag >= 1){
    free(arr);
}
if (s != 0){
    arr = Input(&flag, s);
    f = s;
    printf("\nВаш массив:\n");
    Out(arr, s);
    break;
}
else{
    printf("\nИндекс введён
некорректно\n");
    break;
}
case 'b': //задание случайного массива
printf("Инициализирован ввод случайного
массива\nВведите число элементов массива\n");
s = InputST3();
if (flag >= 1){
    free(arr);
}
if (s != 0){
    arr = InRan(&flag, s);
    f = s;
    printf("\nВаш массив:\n");
    Out(arr, s);
    break;
}
else{
    printf("\nИндекс
некорректно\n");
    break;
}
case 'c': //удаление элемента массива по индексу
if (flag >= 1){
    printf("Инициализировано удаление
элемента массива\nВведите индекс массива (номер элемента)\n");
    size_t ind = InputST3();
    if (ind > f || ind == 0){
        printf("\nИндекс
некорректно\n");
        break;
    }
    else{
        arr = Del(arr, ind, f);
        f--;
        printf("\nВаш массив:\n");
        Out(arr, f);
        printf("\nСвободных
        ячеек:
        %zu\n", s-f);
        break;
    }
}
else{
    printf("Массив не обнаружен\n");
    break;
}
case 'd': //вставка элемента по индексу
if (flag >= 1){

```



```

        элемента в массив");

        куда вставить число\n");

        введён некорректно\n");

        %zu\n", s-f);

        слишком маленький\n");

    }
    else{
        printf("Массив не обнаружен\n");
        break;
    }
    case 'e': //удаление элементов с первой "9"
        if (flag >= 1){
            printf("Инициализировано          удаление
элементов массива со стар. знач. цифрой '9'");
            arr = BigDel(arr, s, &f);
            printf("\nВаш массив:\n");
            Out(arr, f);
            printf("\nСвободных ячеек: %zu\n", s-
f);

            break;
        }
        else{
            printf("Массив не обнаружен\n");
            break;
        }
    case 'f': //Вывод массива
        if (flag >= 1){
            printf("\nВаш массив:\n");
            Out(arr, f);
            printf("\nСвободных ячеек: %zu\n", s-
f);

            break;
        }
        else{
            printf("Массив не обнаружен\n");
            break;
        }
    case 'g': //расширение массива
        if (flag >= 1){
            printf("Введите число ячеек, которые
необходимо добавить к массиву\n");

            int n;
            n = InputST3();

            printf("Инициализирована вставка
            if(f+1<=s){
                int ind;
                printf("\nВведите индекс,

                ind = InputST3();
                if (ind-1 > f && f!=0){
                    printf("\nИндекс

                    break;
                }
                arr = Paste(arr, f, ind);
                f += 1;
                printf("\nВаш массив:\n");
                Out(arr, f);
                printf("\nСвободных ячеек:

                break;
            }
            else{
                printf("\nОшибка! Ваш массив

                break;
            }
        }
    }
}
else{
    printf("Массив не обнаружен\n");
    break;
}
}
case 'f': //Вывод массива
    if (flag >= 1){
        printf("\nВаш массив:\n");
        Out(arr, f);
        printf("\nСвободных ячеек: %zu\n", s-
f);

        break;
    }
    else{
        printf("Массив не обнаружен\n");
        break;
    }
}
case 'g': //расширение массива
    if (flag >= 1){
        printf("Введите число ячеек, которые
        необходимо добавить к массиву\n");

        int n;
        n = InputST3();

```

```

sizeof(double)*(n+s));

f);

массив!\n");

arr = realloc(arr,

s += n;
printf("\nВаш массив:\n");
Out(arr, f);
printf("\nСвободных ячеек: %zu\n", s-

printf("\nПоздравляем, вы увеличили

break;

}
else{
printf("Массив не обнаружен\n");
break;

}
case 'h'://уменьшение массива
if (flag >= 1){
if (s==f){
printf("Свободных для

break;

}
printf("Введите число ячеек, которые

int n;
n = InputST3();
if (s-f<n){
printf("\nВы не можете

break;

}
arr = realloc(arr, sizeof(double)*(s-

s -= n;
printf("\nВаш массив:\n");
Out(arr, f);
printf("\nСвободных ячеек: %zu\n", s-

printf("\nВаш массива уменьшен\n");
break;

}
else{
printf("Массив не обнаружен\n");
break;

}
case 'p'://выход из программы
printf("Осуществлён выход из программы\n");
break;
default:
printf("Введено некорректное значение\n");

}
}while(choice != 'p');
free(arr);
return 0;
}

```

3. Del.c

```

#include "lab3H.h"
double* Del(double arr[], size_t ind, size_t f){
for (int i = ind; i < f; i++){
arr[i-1] = arr[i];
}
arr[f-1] = 0;
}

```

```

    return arr;
}

```

4. Paste.c

```

#include "lab3H.h"
double* Paste(double arr[], size_t f, size_t ind){
double element;
printf("\nВведите число, которое нужно вставить\n");
element = InputD();
for (int i = f; i > ind-1; i--){
    arr[i] = arr[i-1];
}
arr[ind-1] = element;
return arr;
}

```

5. BigDel.c

```

#include "lab3H.h"
double* BigDel(double arr[], size_t s, size_t *f){
    int count1 = 0;
    int count2 = 0;
    double* narr = (double*) malloc(sizeof(double)*(s));
    double* narr2 = (double*) malloc(sizeof(double)*(s));
    for (int i = 0; i < *f; i++){
        int deg;
        double element = fabs(arr[i]);
        if (element != 0){
            while (element < 1){
                element *= 10;
            }
            deg = (int)log10(element);
            if ((int)(element/pow(10, deg)) != 9){
                narr[count1] = arr[i];
                count1 += 1;
            }
            else{
                narr2[count2] = arr[i];
                count2 += 1;
            }
        }
        *f -= count2;
        printf("\nВаша подпоследовательность из чисел со старшей значащей цифрой 9:\n");
        Out(narr2, count2);
        free(arr);
        free(narr2);
        return narr;
    }
}

```

6. Input.c

```

#include "lab3H.h"
double* Input(int *flag, size_t s){
    double *arr = (double*) malloc(sizeof(double)*s);
    for (int i = 0; i < s; i++){
        printf("\nВведите %d число массива\n", i+1);
        arr[i] = InputD();
    }
    *flag += 1;
    return arr;
}

```

7. InRan.c

```

#include "lab3H.h"
double* InRan(int *flag, size_t s){

```

```
int low1 = 1000000; int up1 = 2000000;
double *arr = (double*) malloc(sizeof(double)*s);
srand(time(NULL));
for (int i = 0; i < s; i++){
    arr[i] = (rand() % up1 - low1) * 0.001;
}
*flag += 1;
return arr;
}
```

IV.Проверка работоспособности

Входные значения	Выходные значения
a/3/0.9 /2/-0.9	<div>Инициализирован ввод массива</div> <div>Введите число элементов массива</div> <div>3</div> <div>Введите 1 число массива</div> <div>0.9</div> <div>Введите 2 число массива</div> <div>2</div> <div>Введите 3 число массива</div> <div>-0.9</div> <div>Ваш массив:</div> <div><div>0.900000</div><div>2.000000</div><div>-0.900000</div></div> <div>Введите символ, чтобы выбрать соответствующий пункт меню</div> <div>a) - ввод массива</div> <div>b) - создание рандомного массива</div> <div>c) - удаление элемента массива</div> <div>d) - вставка элемента массива по индексу</div> <div>e) - удаление всех элементов со старшей значимой цифрой 9</div> <div>f) - вывод массива</div> <div>g) - добавить памяти</div> <div>h) - убрать памяти</div> <div>p) - выход</div>
b/20	<div>Инициализирован ввод рандомного массива</div> <div>Введите число элементов массива</div> <div>20</div> <div>Ваш массив:</div> <div><div>503.882000</div><div>-765.273000</div><div>802.425000</div><div>165.770000</div><div>-937.141000</div><div>-874.501000</div><div>-718.256000</div><div>-188.038000</div><div>504.678000</div><div>-27.958000</div><div>4</div></div> <div><div>12.553000</div><div>185.558000</div><div>-798.744000</div><div>164.880000</div><div>48.866000</div><div>-47.893000</div><div>381.847000</div><div>-767.627000</div><div>275.483000</div><div>-725.642000</div><div></div></div> <div>Введите символ, чтобы выбрать соответствующий пункт меню</div> <div>a) - ввод массива</div> <div>b) - создание рандомного массива</div> <div>c) - удаление элемента массива</div> <div>d) - вставка элемента массива по индексу</div> <div>e) - удаление всех элементов со старшей значимой цифрой 9</div> <div>f) - вывод массива</div> <div>g) - добавить памяти</div> <div>h) - убрать памяти</div> <div>p) - выход</div>
c/3	<div>Инициализировано удаление элемента массива</div> <div>Введите индекс массива (номер элемента)</div> <div>3</div> <div>Ваш массив:</div> <div><div>914.967000</div><div>581.923000</div><div>255.726000</div><div>247.614000</div><div>-158.301000</div><div>777.832000</div><div>676.790000</div><div>332.785000</div><div>-393.253000</div><div>487.336000</div><div>-</div></div> <div><div>523.174000</div><div>831.343000</div><div>-358.466000</div><div>-65.158000</div><div>-943.053000</div><div>207.742000</div><div>605.408000</div><div>-823.350000</div><div>4.459000</div><div></div><div></div></div> <div>Свободных ячеек: 1</div>
d/4/0.23	<div>Инициализирована вставка элемента в массив</div> <div>Введите индекс, куда вставить число</div> <div>4</div> <div>Введите число, которое нужно вставить</div> <div>0.23</div> <div>Ваш массив:</div> <div><div>914.967000</div><div>581.923000</div><div>255.726000</div><div>0.230000</div><div>247.614000</div><div>-158.301000</div><div>777.832000</div><div>676.790000</div><div>332.785000</div><div>-393.253000</div><div>4</div></div> <div><div>87.336000</div><div>-523.174000</div><div>831.343000</div><div>-358.466000</div><div>-65.158000</div><div>-943.053000</div><div>207.742000</div><div>605.408000</div><div>-823.350000</div><div>4.459000</div><div></div></div> <div>Свободных ячеек: 0</div>
e	<div>Инициализировано удаление элементов массива со стар. знач. цифрой '9'</div> <div>Ваша подпоследовательность из чисел со старшей значащей цифрой 9:</div> <div><div>914.967000</div><div>-943.053000</div></div> <div>Ваш массив:</div> <div><div>581.923000</div><div>255.726000</div><div>0.230000</div><div>247.614000</div><div>-158.301000</div><div>777.832000</div><div>676.790000</div><div>332.785000</div><div>-393.253000</div><div>487.336000</div><div>-</div></div> <div><div>523.174000</div><div>831.343000</div><div>-358.466000</div><div>-65.158000</div><div>207.742000</div><div>605.408000</div><div>-823.350000</div><div>4.459000</div><div></div><div></div></div> <div>Свободных ячеек: 2</div>
f	<div>Ваш массив:</div> <div><div>581.923000</div><div>255.726000</div><div>0.230000</div><div>247.614000</div><div>-158.301000</div><div>777.832000</div><div>676.790000</div><div>332.785000</div><div>-393.253000</div><div>487.336000</div><div>-</div></div> <div><div>523.174000</div><div>831.343000</div><div>-358.466000</div><div>-65.158000</div><div>207.742000</div><div>605.408000</div><div>-823.350000</div><div>4.459000</div><div></div><div></div></div> <div>Свободных ячеек: 2</div>

g/10	<div>Введите число ячеек, которые необходимо добавить к массиву 10 Ваш массив: 581.923000 255.726000 0.230000 247.614000 -158.301000 777.832000 676.790000 332.785000 -393.253000 487.336000 - 523.174000 831.343000 -358.466000 -65.158000 207.742000 605.408000 -823.350000 4.459000 Свободных ячеек: 12 Поздравляем, вы увеличили массив!</div>
p	<div>Введите символ, чтобы выбрать соответствующий пункт меню a) - ввод массива b) - создание рандомного массива c) - удаление элемента массива d) - вставка элемента массива по индексу e) - удаление всех элементов со старшей значимой цифрой 9 f) - вывод массива g) - добавить памяти h) - убрать памяти p) - выход Осуществлѐн выход из программы</div>
h/10	<div>Введите число ячеек, которые необходимо убрать из массива 12 Ваш массив: -232.720000 -31.785000 -880.279000 -745.690000 -679.301000 37.399000 893.479000 -654.247000 322.734000 223.154000 - 282.592000 -544.474000 465.451000 -411.067000 557.861000 558.934000 759.547000 25.874000 Свободных ячеек: 0 Ваш массива уменьшен Введите символ, чтобы выбрать соответствующий пункт меню a) - ввод массива b) - создание рандомного массива c) - удаление элемента массива d) - вставка элемента массива по индексу e) - удаление всех элементов со старшей значимой цифрой 9 f) - вывод массива g) - добавить памяти h) - убрать памяти p) - выход</div>

V. Выводы

В ходе работы над лабораторной работой №3 хоть и не были усовершенствованы функции безопасного ввода, но были сделаны существенные продвижения на других направлениях: были освоены массивы, примитивная работа с памятью, освоена работа с многофайловыми программами, отлажена работа с пользовательским меню.

Также существенные продвижения касаются и алгоритмической части. Возрастающие объемы данных и соответственно более громоздкий алгоритм заставляют более грамотно подходить к организации и структуризации различных функций и действий. К примеру, в данной лабораторной работе для функции `main()` необходимо было разделить на части (для более эффективного чтения блок-схем и лучшего представления), связанные межстраничными соединителями.

Функционал всех необходимых функций (и соответствующих им пунктов меню) был реализован в полной мере, требования лабораторной работы можно считать удовлетворёнными.