ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
О ЛАБОРАТОРНОЙ РАБОТЕ №4.1

По дисциплине «Языки программирования»

ВАРИАНТ 3

Выполнил: ст. гр. ТКИ-142

Величко Иван Сергеевич

Проверил: к.т.н., доц. Васильева М. А.

(Проверил: к.т.н, доц. Балакина Е. П.)

Москва 2024

1. Формулировка задания

Создать одномерный массив из n целых чисел. Реализовать возможность заполнения массива, как случайными числами, так и с помощью клавиатуры по желанию пользователя. Пункты задания организовать в виде функций (методов). Составить блок-схему.

Таблица 1 – Исходные данные

|  |  |  |
| --- | --- | --- |
| Вариант | Задачи | Константы |
| 3 | 1. Найти сумму элементов, имеющих нечетные индексы.  2. Подсчитать количество элементов массива, значения которых больше заданного числа А и кратных 5.  3. Разделить все элементы массива с четными номерами на первый элемент (первый элемент отличен от 0) | [-10;20] |

1. Блок-схема алгоритма

Ниже представлены:

Блок-схема основного алгоритма (Рисунок 1).

Блок-схема основного алгоритма (Рисунок 1.2).

Блок-схема функции random\_input() (Рисунок 2).

Блок-схема функции input() (Рисунок 3).

Блок-схема функции input\_positive() (Рисунок 4).

Блок-схема функции keyboard\_input() (Рисунок 5).

Блок-схема функции sum() (Рисунок 6).

Блок-схема функции count() (Рисунок 7).

Блок-схема функции division() (Рисунок 8).

Блок-схема функции print\_array() (Рисунок 9).

Блок-схема функции check\_range() (Рисунок 10).

Блок-схема функции print\_array() (Рисунок 11).

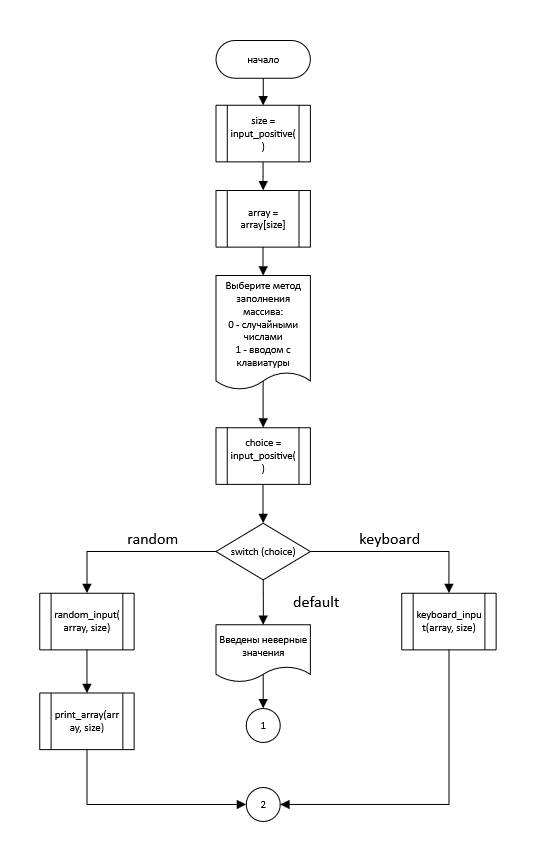


Рисунок 1 – Блок-схема основного алгоритма

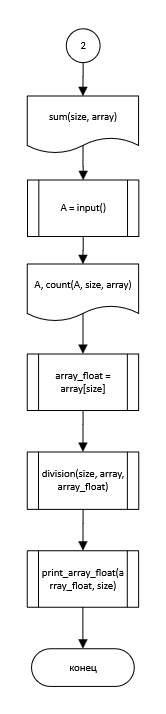


Рисунок 1.1 – Блок-схема основного алгоритма

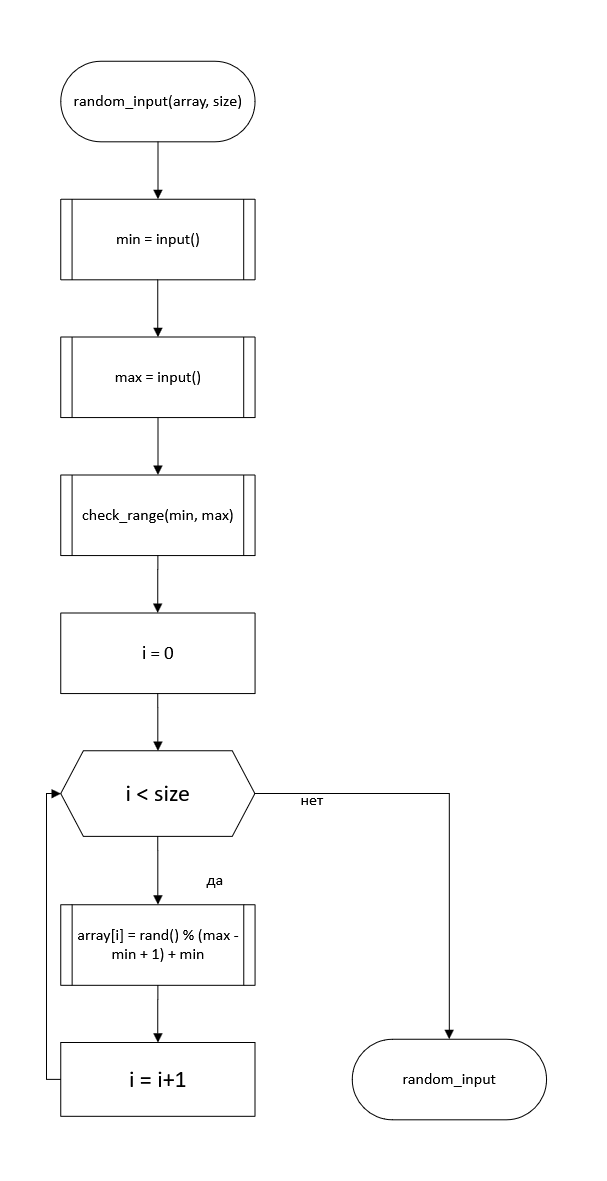


Рисунок 2 – Блок-схема функции random\_input()



Рисунок 3 – Блок-схема функции input()

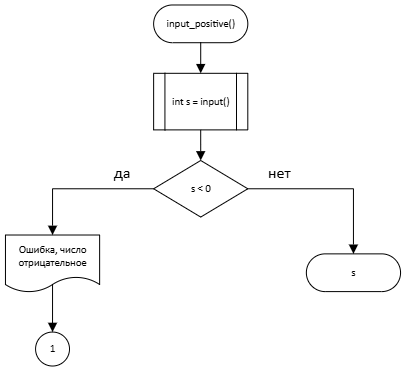


Рисунок 4– Блок-схема функции input\_positive()

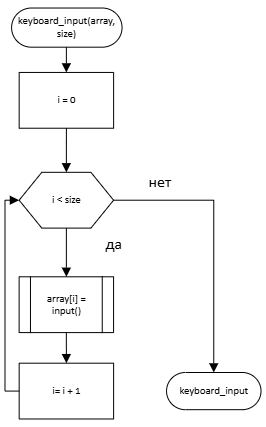


Рисунок 5 – Блок-схема функции keyboard\_input()

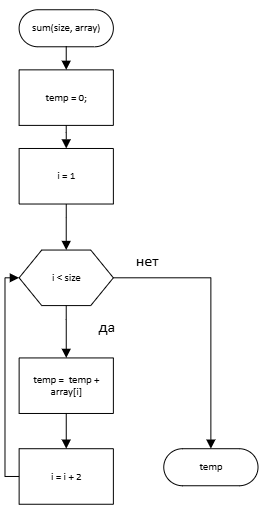


Рисунок 6 – Блок-схема функции sum()

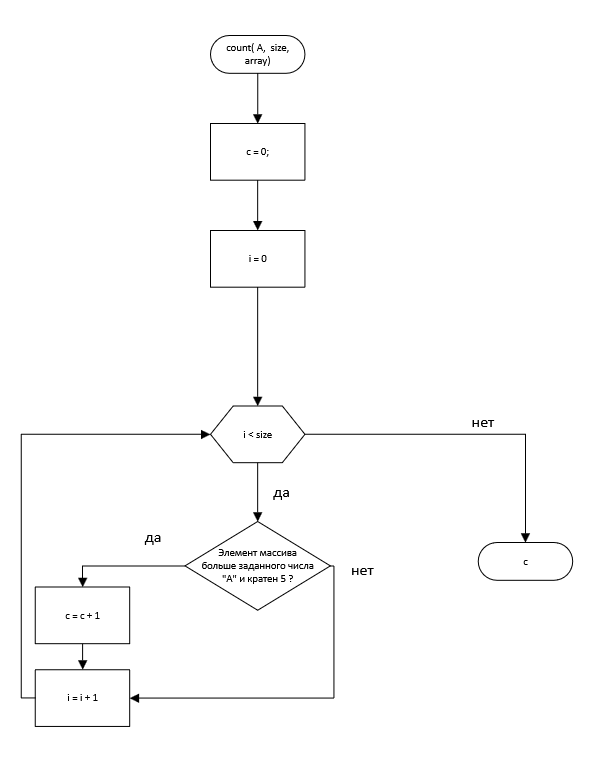


Рисунок 7 – Блок-схема функции count()

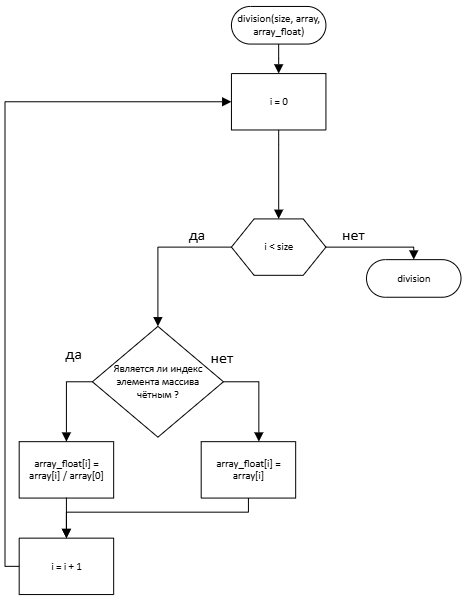


Рисунок 8 – Блок-схема функции division()

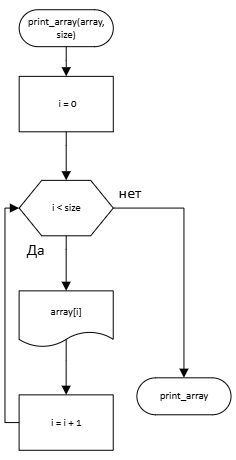


Рисунок 9 – Блок-схема функции print\_array()

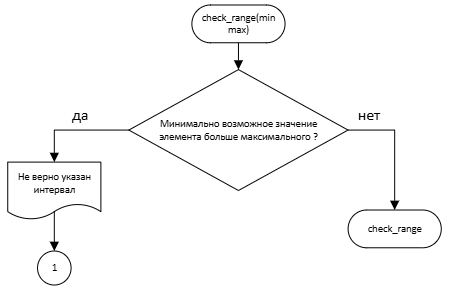


Рисунок 10 – Блок-схема функции check\_range()

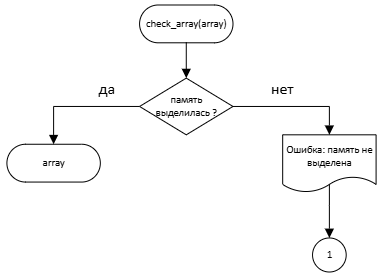


Рисунок 11 – Блок-схема функции check\_array()

1. Текст программы на языке C

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <errno.h>

/\*\*

\* @brief принимает из stdin, а затем возвращает значение переменной value

\* @return числовое значение полученное из stdin

\*/

int input(void);

/\*\*

\* @brief принимает из stdin, а затем возвращает значение переменной value, проверяет является ли число положительным

\* @return положительное числовое значение

\*/

size\_t input\_positive(void);

/\*\*

\* @brief заполняет массив случайно сгенерироваными числами

\* @param array указатель на массив

\* @param size размер массива

\*/

void random\_input(int\* const array, const size\_t size);

/\*\*

\* @brief заполняет массив из стандартного потока ввода

\* @param array указатель на массив

\* @param size размер массива

\*/

void keyboard\_input(int\* array, const size\_t size);

/\*\*

\* @brief расчитывает сумму элементов, имеющих нечетные индексы.

\* @param array указатель на массив

\* @param size размер массива

\* @return сумму элементов, имеющих нечетные индексы

\*/

int sum(const size\_t size, int\* const array);

/\*\*

\* @brief расчитывает количество элементов массива, значения которых больше заданного числа А и кратных 5.

\* @param array указатель на массив

\* @param size размер массива

\* @return количество элементов массива, значения которых больше заданного числа А и кратных 5.

\*/

int count(const int A, const size\_t size, int\* array);

/\*\*

\* @brief делит все элементы массива с четными номерами на первый элемент (первый элемент отличен от 0).

\* @param array указатель на массив

\* @param size размер массива

\* @param array\_float указатель на массив чисел с плававющей точкой

\*/

void division(const size\_t size, int\* array, float\* array\_float);

/\*\*

\* @brief выводит содержимое массива

\* @param array указатель на массив

\* @param size размер массива

\*/

void print\_array(int\* array, const size\_t size);

/\*\*

\* @brief выводит содержимое массива чисел с плавающей точкой

\* @param array указатель на массив

\* @param size размер массива

\*/

void print\_array\_float(float\* array\_float, const size\_t size);

/\*\*

\* @brief проверяет правильность интервала

\* @param min минимально возможное значение элемента

\* @param max максимально возможное значение элемента

\*/

void check\_range(const int min, const int max);

/\*\*

\* @brief проверяет выделение памяти

\* @param array указатель на массив

\*/

void check\_array(int\* const array);

enum method {

random,

keyboard

};

int main() {

puts("Введите размер массива:");

size\_t size = input\_positive();

int\* array = (int\*)malloc(size \* sizeof(int));

check\_array(array);

printf("Выберите способ заполнения массива:%d - случайными числами\n%d - вводом с клавиатуры: ", (int)random, (int)keyboard);

enum method choice = (enum method)input\_positive();

switch (choice) {

case random:

random\_input(array, size);

puts("Массив: ");

print\_array(array, size);

break;

case keyboard:

keyboard\_input(array, size);

break;

default:

free(array);

errno = EIO;

perror("Введены неверные значения\n");

exit(1);

}

printf("Сумма всех элементов с нечетными индексами: %d\n", sum(size, array));

printf("Введите число A (с которым будут сравниваться элементы массива): ");

int A = input();

printf("Количество элементов массива значения которых больше %d и кратны 5: %d\n", A, count(A, size, array));

float\* array\_float = (float\*)malloc(size \* sizeof(float));

check\_array(array\_float);

division(size, array, array\_float);

free(array);

puts("Массив после деления всех элементов с четными номерами на первый элемент: ");

print\_array\_float(array\_float, size);

free(array\_float);

return 0;

}

int input() {

int value = 0;

int s = scanf\_s("%d", &value);

while (getchar() != '\n');

if (s != 1) {

errno = EIO;

perror("Ошибка, не числовое значение\n");

exit(1);

}

return value;

}

size\_t input\_positive() {

int s = input();

if (s < 0) {

errno = EIO;

perror("Ошибка, число отрицательное\n");

exit(EXIT\_FAILURE);

}

return (size\_t)s;

}

void random\_input(int\* const array, const size\_t size) {

srand(time(NULL));

puts("Введите минимальное значение элемента массива:");

int min = input();

puts("Введите максимальное значение элемента массива:");

int max = input();

check\_range(min, max);

for (size\_t i = 0; i < size; i++) {

array[i] = rand() % (max - min + 1) + min;

}

}

void keyboard\_input(int\* array, const size\_t size) {

for (size\_t i = 0; i < size; i++) {

array[i] = input();

}

}

int sum(const size\_t size, int\* const array) {

int temp = 0;

for (size\_t i = 1; i < size; i += 2) {

temp += array[i];

}

return temp;

}

int count(const int A, const size\_t size, int\* array) {

printf("Введите число A (с которым будут сравниваться элементы массива): ");;

int c = 0;

for (size\_t i = 0; i < size; i++) {

if ((array[i] > A) && (array[i] % 5 == 0)) {

c++;

}

}

return c;

}

void division(const size\_t size, int\* array, float\* array\_float) {

for (size\_t i = 0; i < size; i++) {

if (i % 2 == 0) {

array\_float[i] = (float)array[i] / (float)array[0];

}

else

array\_float[i] = array[i];

}

}

void print\_array(int\* array, const size\_t size) {

for (size\_t i = 0; i < size; i++) {

printf("%d ", array[i]);

}

}

void print\_array\_float(float\* array\_float, const size\_t size) {

for (size\_t i = 0; i < size; i++) {

printf("%0.2f ", array\_float[i]);

}

}

void check\_range(const int min, const int max) {

if (min > max) {

errno = EIO;

perror("Не верно указан интервал\n");

exit(1);

}

}

void check\_array(int\* const array) {

if (!array) {

perror("Ошибка: массив не выделен");

exit(EXIT\_FAILURE);

}

}

1. Результаты выполнения программы

Ниже представлены:

* Примеры правильной работы программы (Рисунок 12, Рисунок 13).
* Результаты тестирования программы (Рисунок 14, Рисунок 15, Рисунок 16, Рисунок 17, Рисунок 18, Рисунок 19, Рисунок 20, Рисунок 21, Рисунок 22, Рисунок 23).

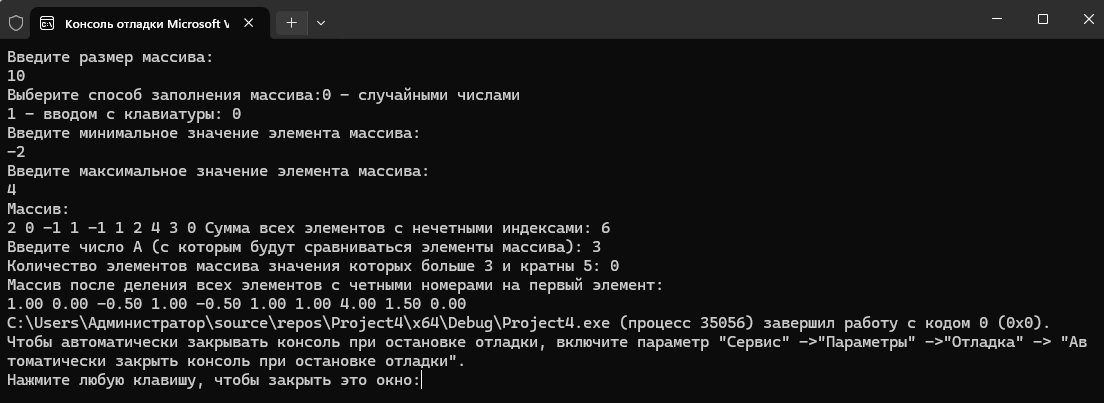


Рисунок 12 – Выполнение программы при выборе метода заполнения случайными числами

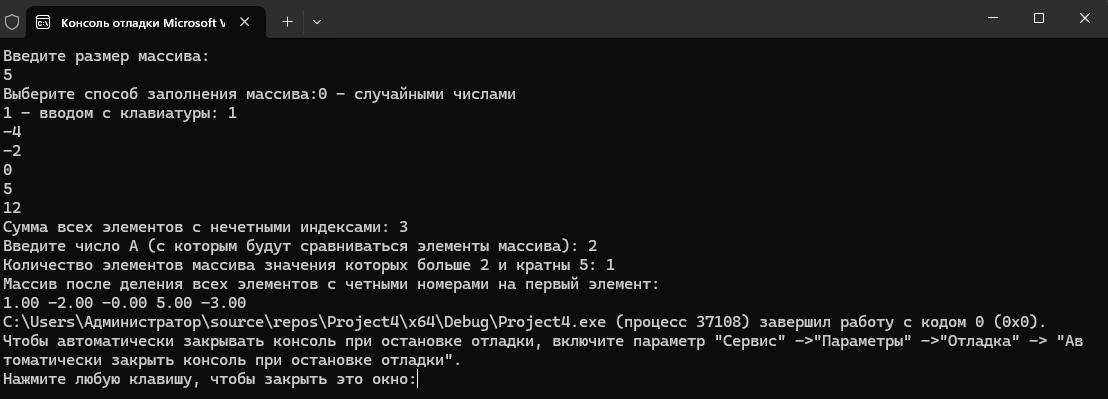


Рисунок 13 – Выполнение программы при выборе метода заполнения с помощью из stdin

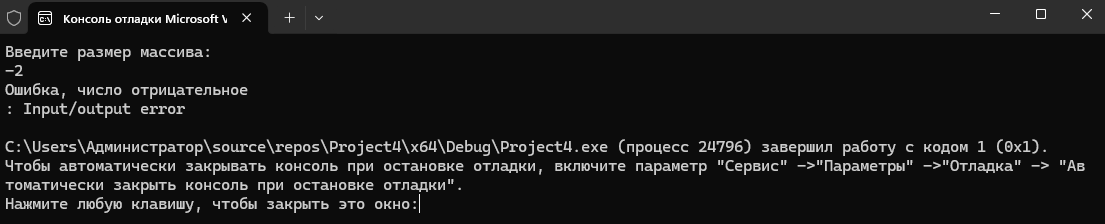


Рисунок 14 – Ввод отрицательного числа в качестве размера массива

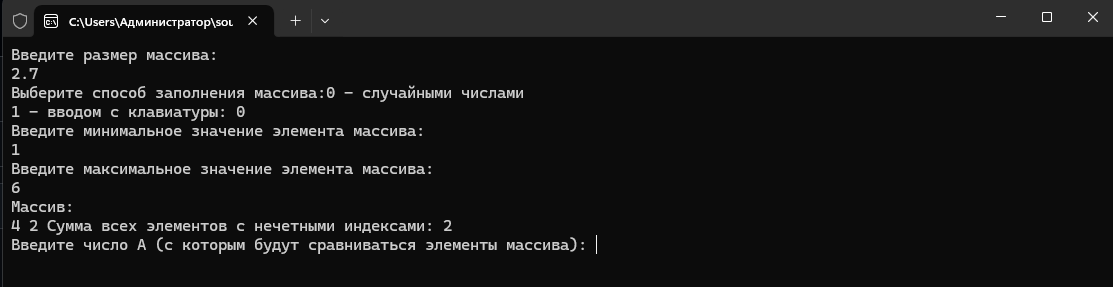


Рисунок 15 – Ввод числа с плавающей точкой в качестве размера массива

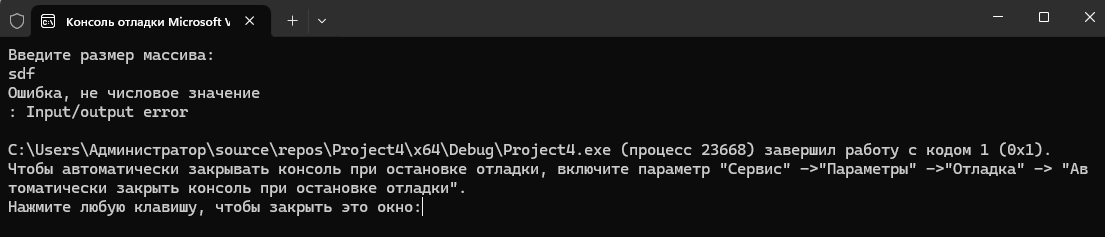


Рисунок 16 – Ввод символьных значений в качестве размера массива

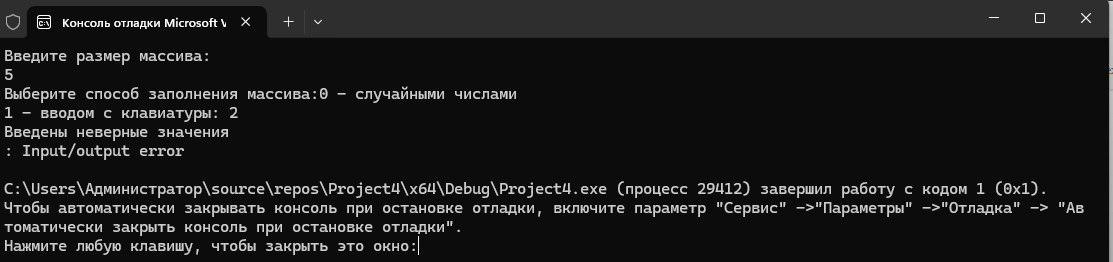


Рисунок 17 – Ввод некорректного значения при выборе метода заполнения массива(не 0 и 1)

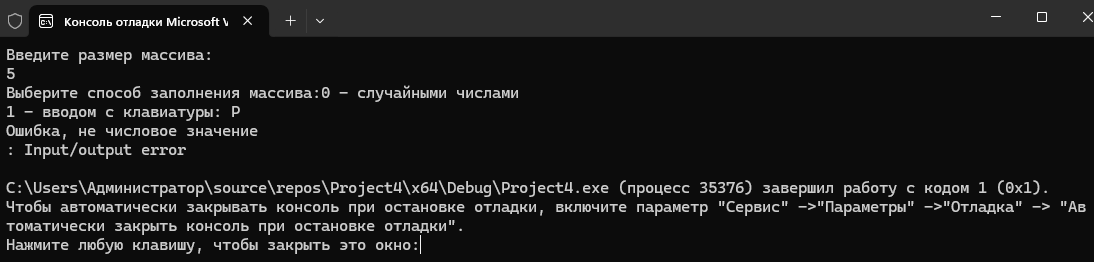


Рисунок 18– Ввод некорректного значения при выборе метода заполнения массива(символьное значение)

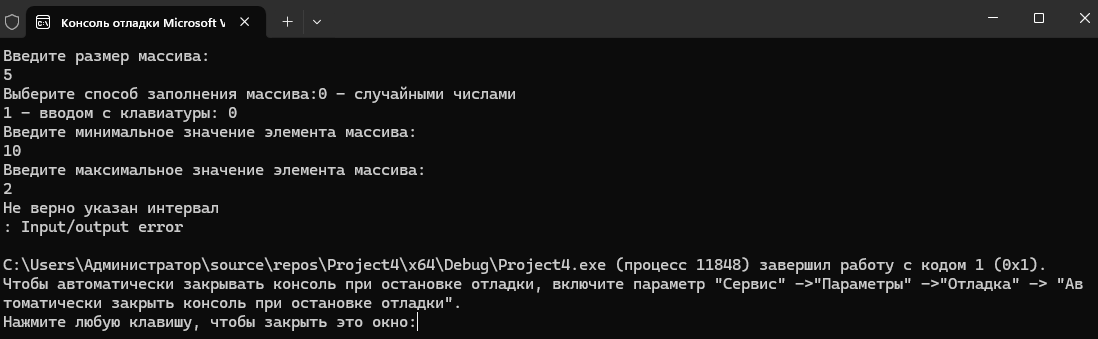


Рисунок 19 – Некорректные значения интервала

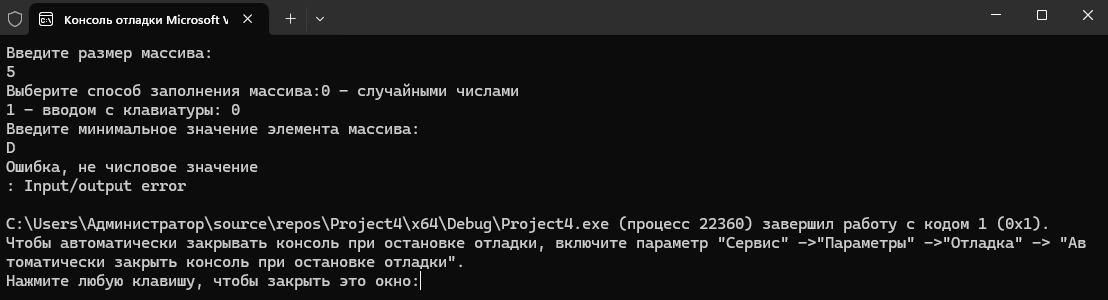


Рисунок 20 – Ввод символьного значения в качестве наименьшего возможного значения элемента массива при случайном заполнении массива

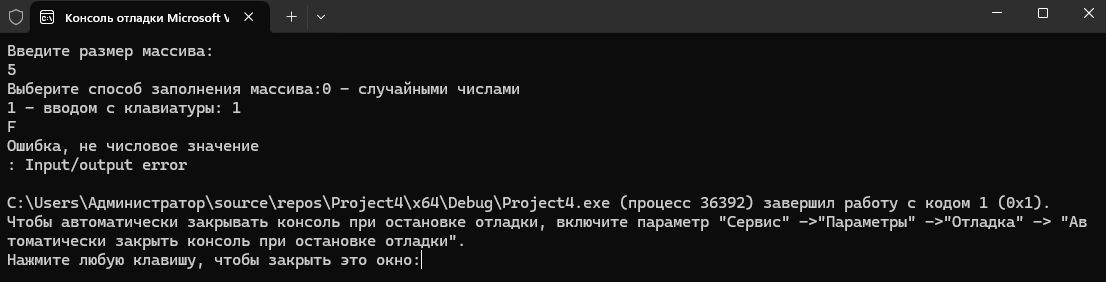


Рисунок 21 – Ввод символьного значения элемента массива при заполнении массива из stdin

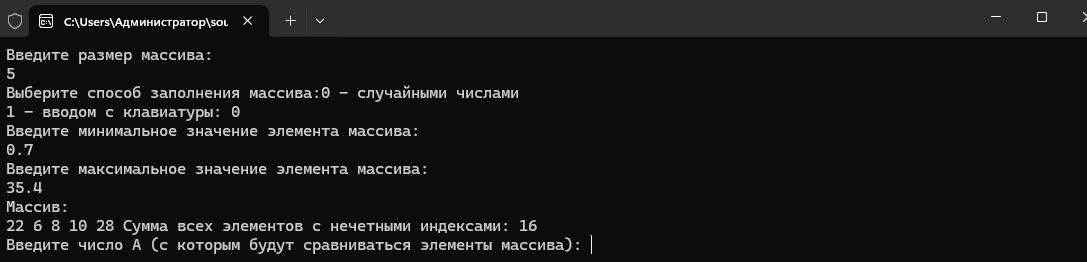


Рисунок 22 – Работа программы с числами с плавающей точкой при заполнении массива случайными числами

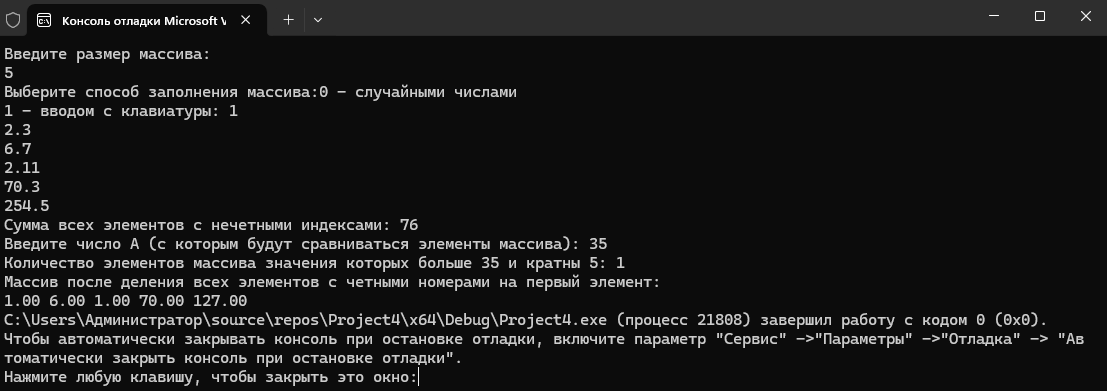


Рисунок 23 – Работа программы с числами с плавающей точкой при заполнении массива из stdin

1. Отметка о выполнении задания в веб-хостинге системы контроля версий

Отметка о выполнение задания представлена ниже (Рисунок 24).

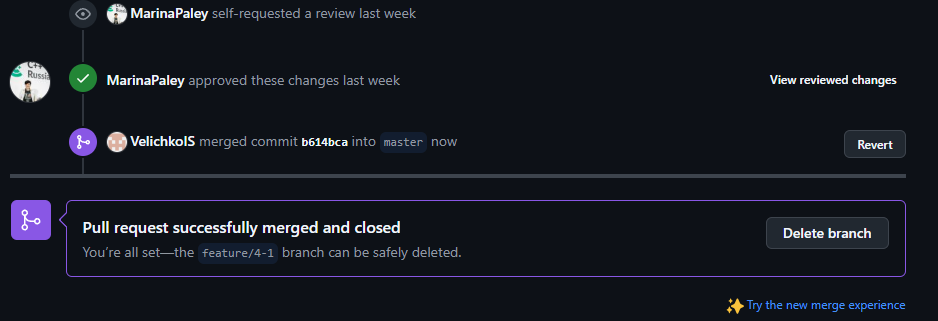


Рисунок 24 – Отметка о выполнении задания