

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.03.01 | Информатика и вычислительная техника | | |
| Направленность (профиль) | |  | Системы автоматизированного проектирования | | |
| Форма обучения | |  | очная | | |
|  | |  |  | | |
| Факультет | |  | Информационных технологий и управления | | |
| Кафедра | |  | Систем автоматизированного проектирования и управления | | |
| Учебная дисциплина | |  | Программирование | | |
| Курс | I | | | Группа | 4ХХ |

Отчёт по контрольной работе №

Вариант №

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Исполнитель: |  |  |  |  |
| обучающийся группы 4ХХ |  |  |  | Фамилия Имя Отчество |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |
| Проверил(и): |  |  |  | Корниенко Иван Григорьевич |
|  |  | (дата, подпись) |  | Макарук Роман Валерьевич |
|  |  |  |  | Федин Алексей Константинович |

Содержание

[1 Задание №1 3](#_Toc51078351)

[1.1 Цель работы 3](#_Toc51078352)

[1.2 Постановка задачи 3](#_Toc51078353)

[1.3 Описание хода выполнения 3](#_Toc51078354)

[1.4 Блок-схема алгоритма решения задачи 4](#_Toc51078355)

[1.5 Исходный код полученного программного решения 5](#_Toc51078356)

[1.6 Тестирование 5](#_Toc51078357)

[1.7 Выводы по заданию №1 6](#_Toc51078358)

[2 Задание №2 7](#_Toc51078359)

[2.1 Цель работы 7](#_Toc51078360)

[2.2 Постановка задачи 7](#_Toc51078361)

[2.3 Описание хода выполнения 7](#_Toc51078362)

[2.4 Блок-схема алгоритма решения задачи 7](#_Toc51078363)

[2.5 Исходный код полученного программного решения 9](#_Toc51078364)

[2.6 Тестирование 9](#_Toc51078365)

[2.7 Выводы по заданию №2 9](#_Toc51078366)

[3 Задание №3 10](#_Toc51078367)

[3.1 Цель работы 10](#_Toc51078368)

[3.2 Постановка задачи 10](#_Toc51078369)

[3.3 Описание хода выполнения 10](#_Toc51078370)

[3.4 Блок-схема алгоритма решения задачи 10](#_Toc51078371)

[3.5 Исходный код полученного программного решения 11](#_Toc51078372)

[3.6 Тестирование 12](#_Toc51078373)

[3.7 Выводы по заданию №3 12](#_Toc51078374)

# 1 Задание №1

## 1.1 Цель работы

Пример: «изучение операторов *if* …».

## 1.2 Постановка задачи

Пример: «Разработать программу для вычисления арифметического выражения (1) и вывода полученного результата. Исходные данные *a, b* ввести с клавиатуры».

В этом же разделе приводятся формулы для расчёта и исходные данные для ввода в программу. Формулы обязательно необходимо набрать с помощью компонента из меню «Вставка -> Уравнение». Эти материалы доступны в файле с контрольными работами.

|  |  |
| --- | --- |
|  | (1) |

## 1.3 Описание хода выполнения

Пример: «Для выполнения задана необходимо дополнительно изучить …Чтобы правильно решить приведённое уравнение, необходимо дополнительно… Для упрощения процесса тестирования и отладки программы, было принято решение о инициализации переменных значениями, которые предполагают ввод с клавиатуры. При разработке блок-схемы было принято решение записать выражение (1) в виде нескольких уравнений (1.1) и (1.2).».

Для решения некоторых задач, в том числе для выполнения блок-схем, может потребоваться введение новых переменных, обозначение для которых описывается в данном разделе. Например: «Для решения поставленной задачи были введены следующие дополнительные обозначения: *A* – исходный массив символов; *Ho* – указатель на исходный массив символов *A*; *SA* – размер исходного массива символов *A*; *Hf* – указатель на предикат; *P* – результат работы предиката; *R* – отфильтрованный массив символов; *HR* – указатель на отфильтрованный массив; *SR* – размер нового (отфильтрованного) массива». Использование имён переменных, обозначающих перечисленные сущности в тексте программы строго запрещено.

|  |  |
| --- | --- |
|  | (1.1) |
|  | (1.2) |

## 1.4 Блок-схема алгоритма решения задачи

Приводится графическое изображение алгоритма, выполненное по правилам ГОСТ 19.701-90. При нанесении соединительных линий, пользоваться правилом: если линия (в том числе ломанная), содержит хотя бы одно нестандартное направление (справа налево или снизу вверх), то необходимо использовать стрелки. Если блок-схема не помещается на одной странице, необходимо воспользоваться правилами переноса. При этом, подпись под схемой для первой части меняется на «Рисунок 1, лист 1 – Блок-схема алгоритма решения задачи №1.1», а для последующих «Рисунок 1, лист 2» и т.д.

Пример: «На рисунке 1 представлена блок-схема алгоритма решения задачи …».

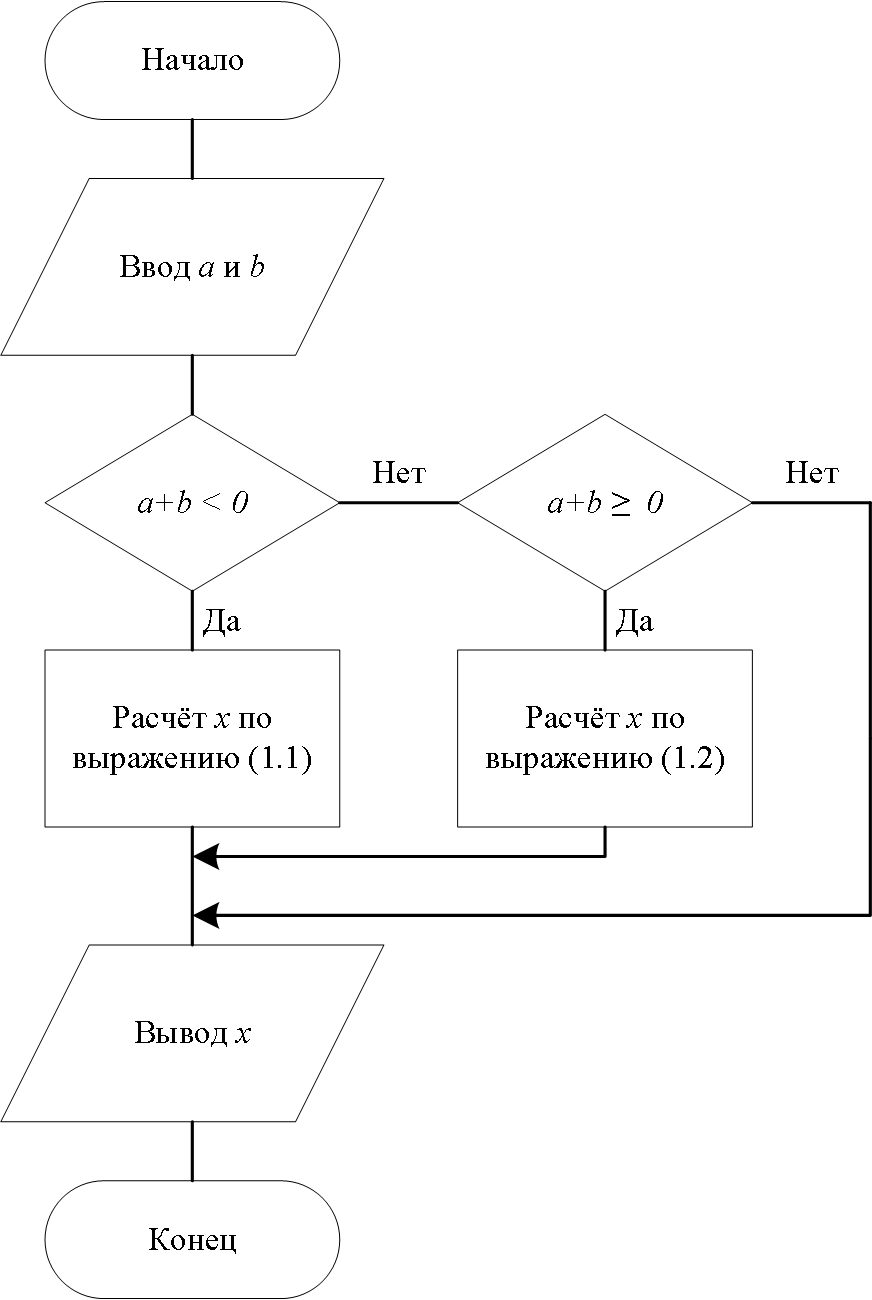


Рисунок 1 – Блок-схема алгоритма решения задачи №1.1

## 1.5 Исходный код полученного программного решения

В разделе приводится листинг программы. Обязательно с применением стиля «Исходный код»!

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Название: src.c \*

\* Задание: Первая программа в осеннем семестре \*

\* Автор: А. К. Федин, СПбГТИ (ТУ), 2021 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\* Название: src.c

\* Задание: Первая программа в осеннем семестре

\* Автор: А. К. Федин, СПбГТИ (ТУ), 2021

\*/

// Директивы препроцессора

#include <stdio.h>

#include <stdlib.h>

// Начальная точка для выполнения программы

int main (void) {

// Инструкции и выражения

puts(“Hello , world !”); // Запись строки в поток stdout

return EXIT\_SUCCESS;

} // Макрос EXIT\_SUCCESS используется для возвращения

// операционной системе кода успешного завершения программы

## 1.6 Тестирование

Тестирование программы было проведено на следующих исходных данных: … Результат тестирования приведён на рисунке/ах 2(-3).

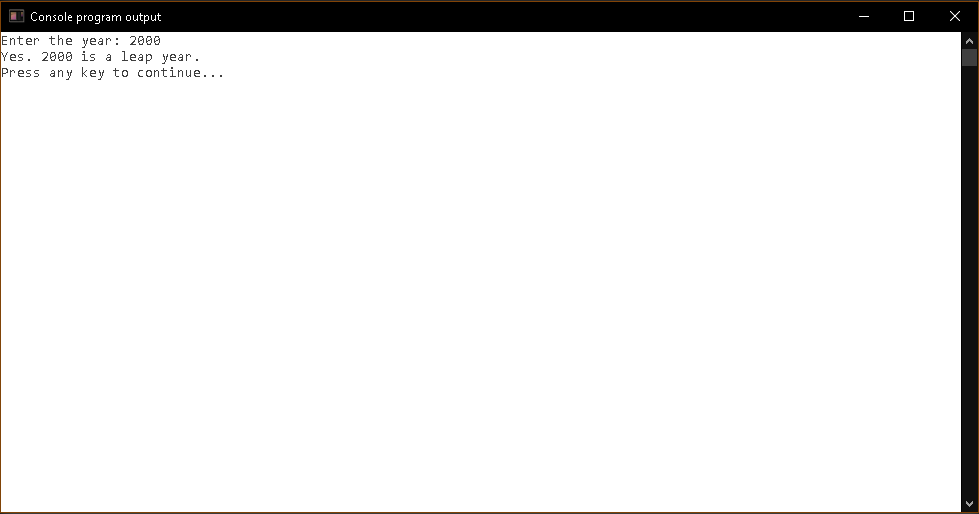


Рисунок 2 – Экранная копия результата работы разработанной программы задания №1.1

## 1.7 Выводы по заданию №1

Например: в ходе выполнения задания №1 контрольной работы №1 были изучены основы работы с оператором *if*, а также получены практические навыки его применения. При решении задачи был обнаружен недостаток знаний элементарных функций, а конкретно гиперболических функций. Для решения данной проблемы была изучена специализированная литература.

Используем слова: изучено, проанализировано, получены результаты, сделан вывод и т.д. и т.п.

# 2 Задание №2

## 2.1 Цель работы

## 2.2 Постановка задачи

## 2.3 Описание хода выполнения

## 2.4 Блок-схема алгоритма решения задачи

На рисунке 3 представлена блок-схема алгоритма решения задачи №1.2.

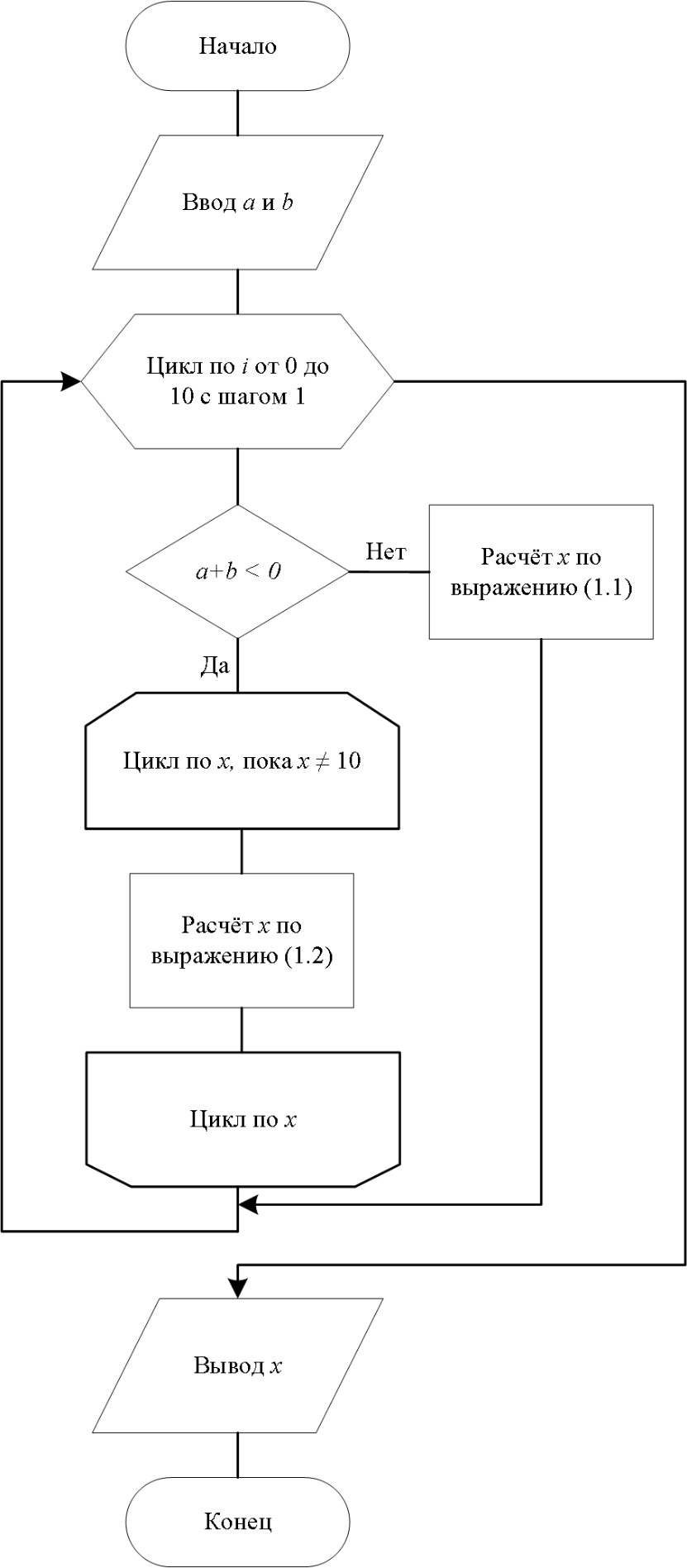


Рисунок 3 – Блок-схема алгоритма решения задачи №1.2

## 2.5 Исходный код полученного программного решения

## 2.6 Тестирование

Рисунок 4 – Экранная копия результата работы разработанной программы задания 1.2

## 2.7 Выводы по заданию №2

# 3 Задание №3

## 3.1 Цель работы

## 3.2 Постановка задачи

## 3.3 Описание хода выполнения

Пример: «Для выполнения задана необходимо дополнительно изучить …Чтобы правильно решить приведённое уравнение, необходимо дополнительно… Для упрощения процесса тестирования и отладки программы, было принято решение о инициализации переменных значениями, которые предполагают ввод с клавиатуры. При разработке блок-схемы было принято решение записать выражение (1) в виде нескольких уравнений (1.1) и (1.2).».

Для решения некоторых задач, в том числе для выполнения блок-схем, может потребоваться введение новых переменных, обозначение для которых описывается в данном разделе. Например: «Для решения поставленной задачи были введены следующие дополнительные обозначения: *A* – исходный массив символов; *Ho* – указатель на исходный массив символов *A*; *SA* – размер исходного массива символов *A*; *Hf* – указатель на предикат; *P* – результат работы предиката; *R* – отфильтрованный массив символов; *HR* – указатель на отфильтрованный массив; *SR* – размер нового (отфильтрованного) массива». Использование имён переменных, обозначающих перечисленные сущности в тексте программы строго запрещено.

## 3.4 Блок-схема алгоритма решения задачи

Приводится графическое изображение алгоритма, выполненное по правилам ГОСТ 19.701-90. При нанесении соединительных линий, пользоваться правилом: если линия (в том числе ломанная), содержит хотя бы одно нестандартное направление (справа налево или снизу вверх), то необходимо использовать стрелки. Если блок-схема не помещается на одной странице, необходимо воспользоваться правилами переноса. При этом, подпись под схемой для первой части меняется на «Рисунок 1, лист 1 – Блок-схема алгоритма решения задачи №1.1», а для последующих «Рисунок 1, лист 2» и т.д.

Пример: «На рисунке 5 представлена блок-схема алгоритма решения задачи №1.3».

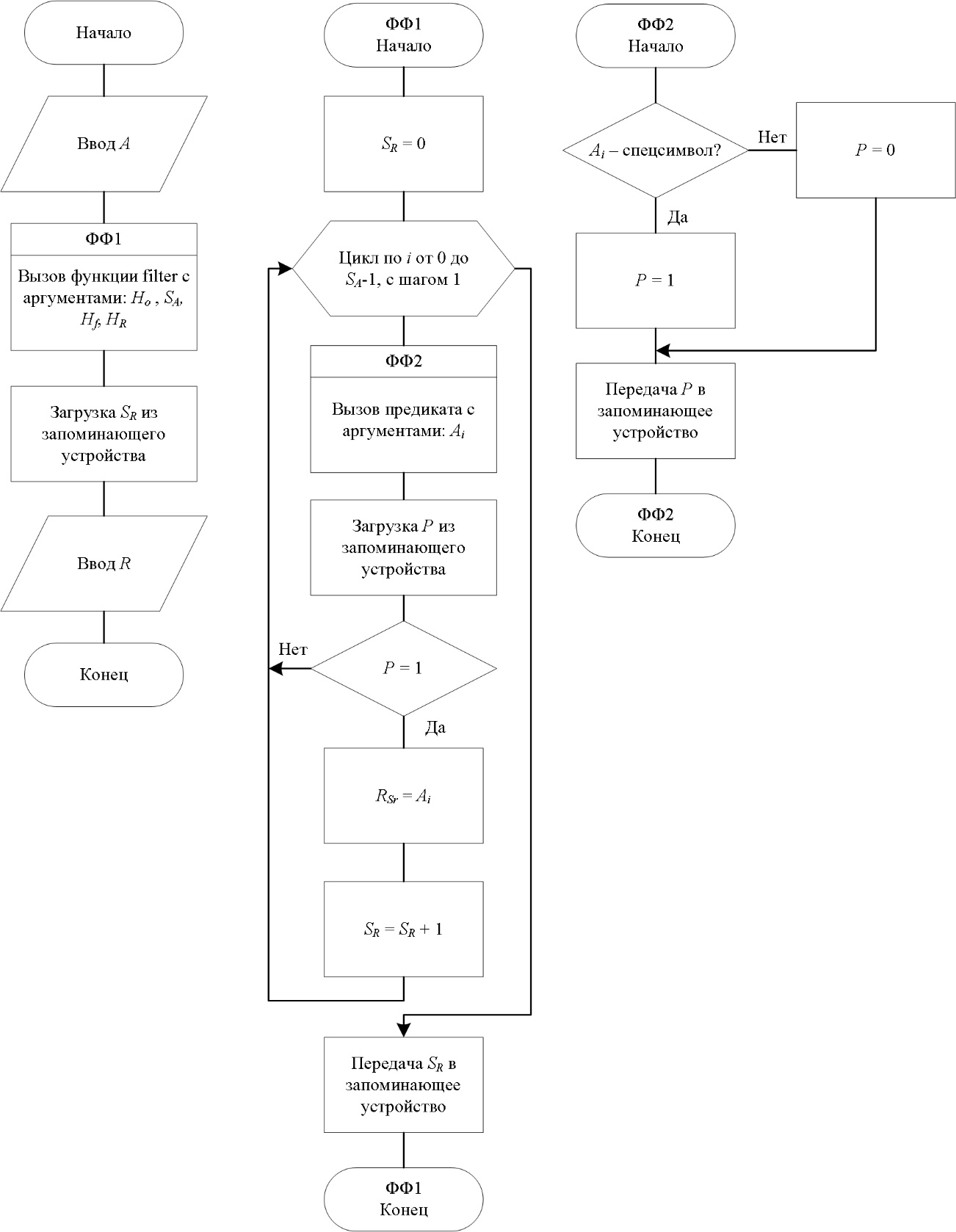


Рисунок 5 – Блок-схема алгоритма решения задачи №1.3

## 3.5 Исходный код полученного программного решения

## 3.6 Тестирование

Рисунок 6 – Экранная копия результата работы разработанной программы задания №1.3

## 3.7 Выводы по заданию №3