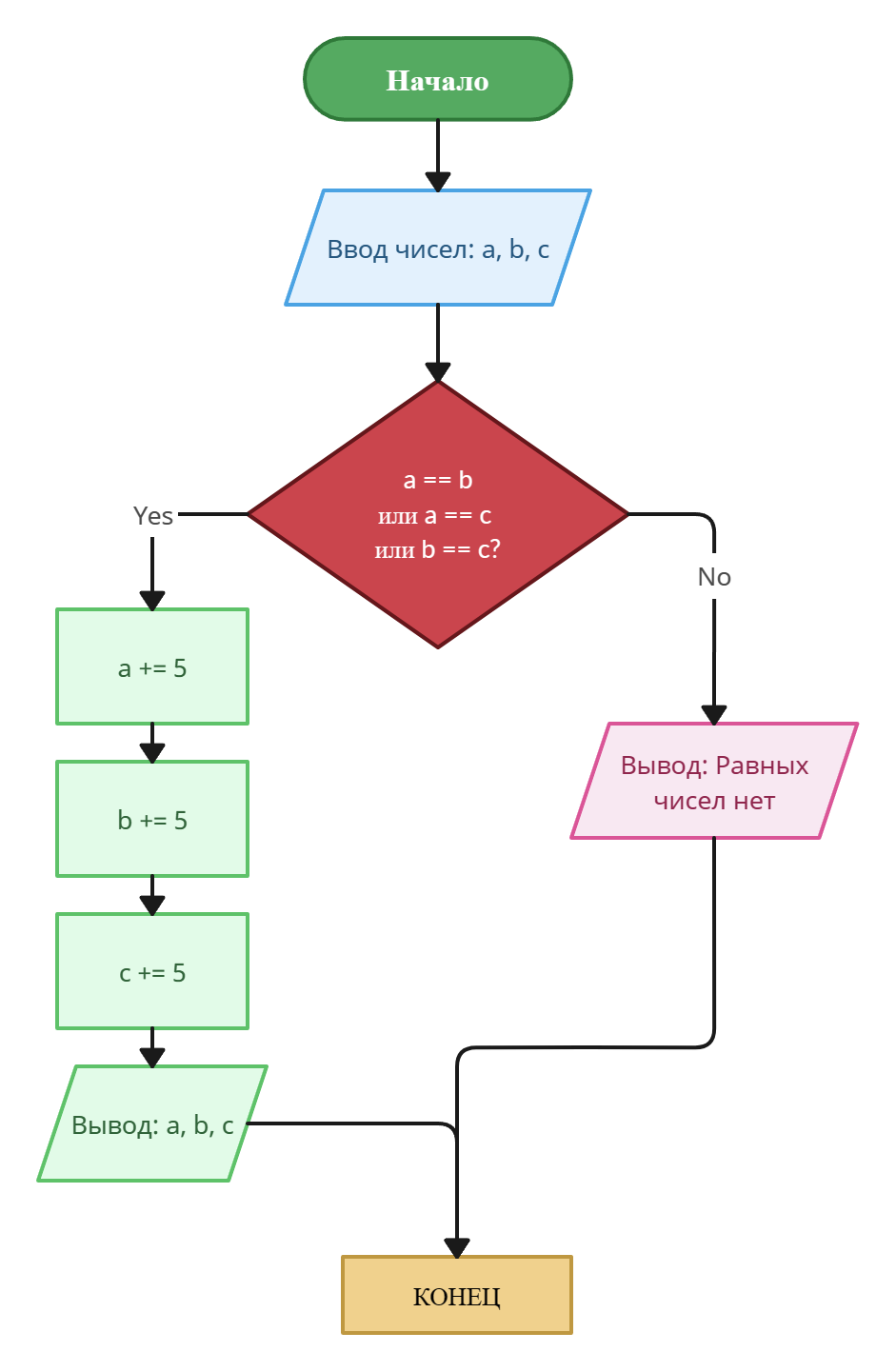
**Обсуждение условных и циклических конструкций, закрепление знаний о строках и массивах**

**Разбор домашнего задания по теме «Булевый тип данных и условия»**

Задача № 1. Составить алгоритм увеличения всех трех чисел, введенных с клавиатуры, на 5, если среди них есть хотя бы два равных числа. В противном случае выдать ответ «равных чисел нет».

Блок-схема задачи № 1:



Пример кода задачи № 1:

#include <iostream>

int main() {

int num1, num2, num3;

// Ввод трех чисел с клавиатуры

std::cout << "Введите первое число: ";

std::cin >> num1;

std::cout << "Введите второе число: ";

std::cin >> num2;

std::cout << "Введите третье число: ";

std::cin >> num3;

// Проверка на наличие хотя бы двух равных чисел

if (num1 == num2 || num1 == num3 || num2 == num3) {

// Увеличение всех трех чисел на 5

num1 += 5;

num2 += 5;

num3 += 5;

std::cout << "Новые значения чисел: " << num1 << ", " << num2 << ", " << num3 << std::endl;

} else {

std::cout << "Равных чисел нет" << std::endl;

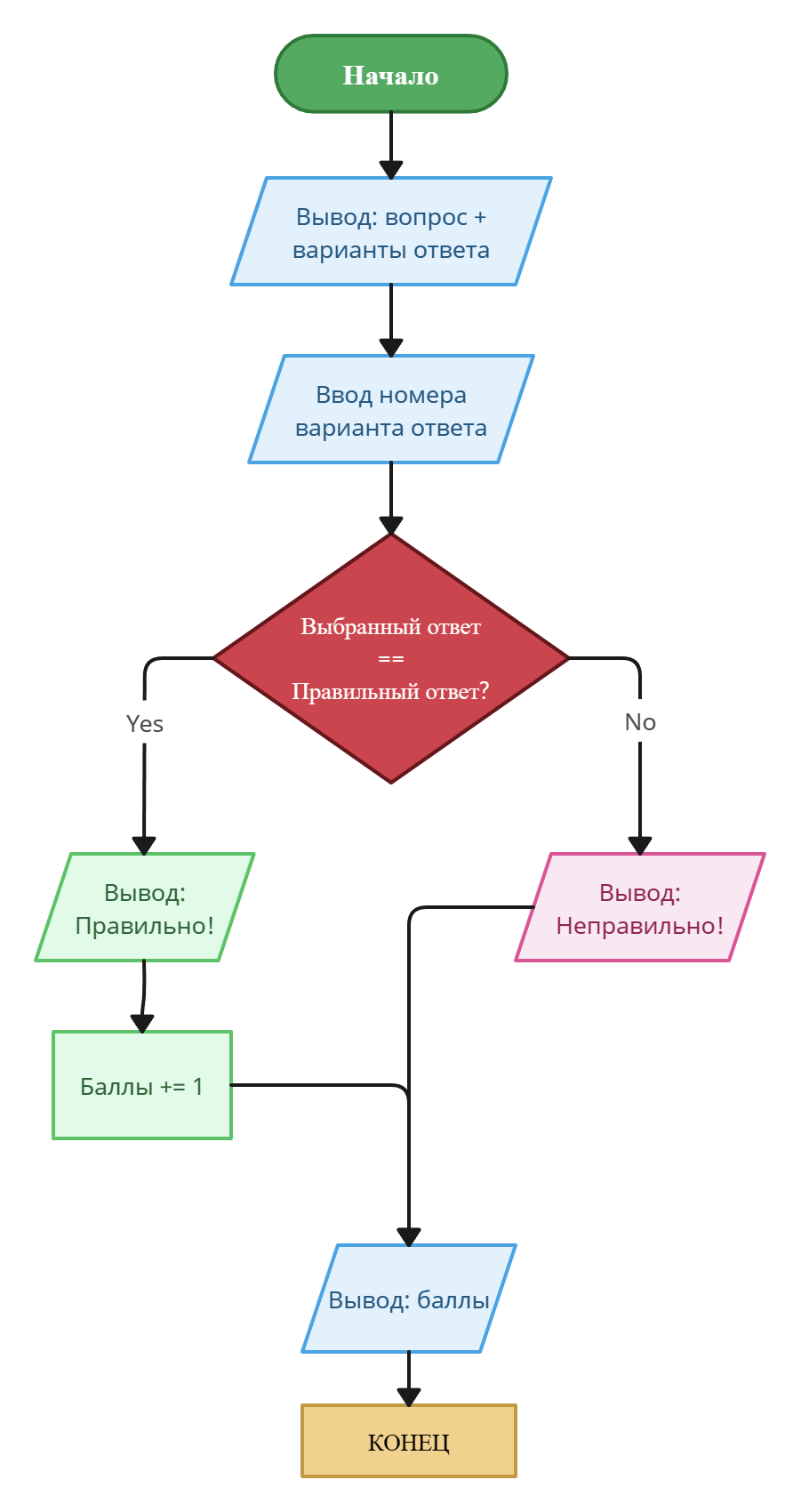
}

return 0;

}

Задача № 2. Составить программу-тест, которая по очереди выводит на экран вопросы (вопросы выбираются вами) с вариантами ответов. В тесте должно быть минимум 3 вопроса. Правильность ответа на вопросы должна проверяться с помощью конструкции «if». В конце работы программа выдает количество заработанных баллов по результатам ответа.

Блок-схема задачи № 2:

****

Пример кода задачи № 2:

#include <iostream>

int main() {

int score = 0; // Переменная для хранения баллов пользователя

// Вопрос 1

std::cout << "Вопрос 1: Какой язык программирования вы изучаете?" << std::endl;

std::cout << "A. C++" << std::endl;

std::cout << "B. Python" << std::endl;

std::cout << "C. Java" << std::endl;

char answer1;

std::cout << "Введите букву вашего ответа: ";

std::cin >> answer1;

if (answer1 == 'A' || answer1 == 'a') {

std::cout << "Правильно!" << std::endl;

score += 1;

} else {

std::cout << "Неправильно." << std::endl;

}

// Вопрос 2

std::cout << "\nВопрос 2: Сколько букв в слове 'программирование'?" << std::endl;

std::cout << "A. 10" << std::endl;

std::cout << "B. 13" << std::endl;

std::cout << "C. 15" << std::endl;

char answer2;

std::cout << "Введите букву вашего ответа: ";

std::cin >> answer2;

if (answer2 == 'B' || answer2 == 'b') {

std::cout << "Правильно!" << std::endl;

score += 1;

} else {

std::cout << "Неправильно." << std::endl;

}

// Вопрос 3

std::cout << "\nВопрос 3: Как называется столица Франции?" << std::endl;

std::cout << "A. Рим" << std::endl;

std::cout << "B. Берлин" << std::endl;

std::cout << "C. Париж" << std::endl;

char answer3;

std::cout << "Введите букву вашего ответа: ";

std::cin >> answer3;

if (answer3 == 'C' || answer3 == 'c') {

std::cout << "Правильно!" << std::endl;

score += 1;

} else {

std::cout << "Неправильно." << std::endl;

}

// Вывод результатов

std::cout << "\nВы набрали " << score << " баллов из 3 возможных." << std::endl;

return 0;

}

**Разбор домашнего задания по теме «Цикл “while”»**

Задача № 1. Написать программу, которая будет считывать с клавиатуры числа в переменную, пока не будет введено число 0.

Пример кода задачи № 1:

#include <iostream>

int main() {

int number;

std::cout << "Введите числа (для завершения введите 0):" << std::endl;

do {

std::cin >> number;

// Здесь вы можете выполнить любые действия с введенным числом, если необходимо

} while (number != 0);

std::cout << "Программа завершена." << std::endl;

return 0;

}

Задача № 2. Напишите программу «Почемучка». Программа должна постоянно задавать вопрос «Почему?» и запрашивать у пользователя ответ до тех пор, пока пользователь не ответит «Потому что».

Пример кода задачи № 2:

#include <iostream>

#include <string>

int main() {

std::string userResponse;

std::cout << "Почему?" << std::endl;

while (true) {

std::cout << "Введите ваш ответ: ";

std::getline(std::cin, userResponse);

if (userResponse == "Потому что") {

std::cout << "Правильно! Программа завершена." << std::endl;

break;

} else {

std::cout << "Почему?" << std::endl;

}

}

return 0;

}

**Разбор домашнего задания по теме «Цикл “for”»**

Задача № 1. Вывести на экран ряд чисел от 1 000 000 до 2 000 000.

Пример кода задачи № 1:

#include <iostream>

int main() {

for (int i = 1000000; i <= 2000000; ++i) {

std::cout << i << " ";

}

return 0;

}

Задача № 2. Напишите программу, которая выведет таблицу умножения на 7.

Пример кода задачи № 2:

#include <iostream>

int main() {

for (int i = 1; i <= 10; ++i) {

std::cout << "7 \* " << i << " = " << 7 \* i << std::endl;

}

return 0;

}

**Разбор домашнего задания по теме «Строки»**

Задача № 1. Маша в свободное время решила поиграть в игру "Найди слово в слове". Например, она нашла слово "нос" в "носорог". Она захотела написать программу, которая бы проверяла, есть ли слово "сон" в вводимой пользователем строке. Помогите ей справиться с этой задачей!

Пример кода задачи № 1:

#include <iostream>

#include <string>

int main() {

std::string inputString;

std::cout << "Введите строку: ";

std::getline(std::cin, inputString);

if (inputString.find("сон") != std::string::npos) {

std::cout << "Слово 'сон' найдено в строке." << std::endl;

} else {

std::cout << "Слово 'сон' не найдено в строке." << std::endl;

}

return 0;

}

Задача № 2. Костя и Паша ввели две строки. У них появилась идея, чтобы строка Кости вставлялась в середину строки Паши. После этого должна выводиться готовая скрещенная строка. Напишите программу для решения этой задачи!

Пример кода задачи № 2:

#include <iostream>

#include <string>

int main() {

std::string kostaString, pashaString;

std::cout << "Введите строку Кости: ";

std::getline(std::cin, kostaString);

std::cout << "Введите строку Паши: ";

std::getline(std::cin, pashaString);

// Вычисляем позицию, в которую нужно вставить строку Кости

int insertPosition = pashaString.length() / 2;

// Вставляем строку Кости в середину строки Паши

pashaString.insert(insertPosition, kostaString);

std::cout << "Скрещенная строка: " << pashaString << std::endl;

return 0;

}

**Разбор домашнего задания по теме «Массивы и ГПСЧ»**

Задача № 1. Создайте массив с 10 элементами. Заполните его с помощью ГПСЧ. Попробуйте вывести четвертый, десятый, сотый или любой другой элемент, который вы не создавали. Почему при выходе за пределы массива мы получаем «мусор», а не нули?

Пример кода задачи № 1:

#include <iostream>

#include <cstdlib>

#include <ctime>

int main() {

const int arraySize = 10;

int myArray[arraySize];

// Инициализация генератора случайных чисел с текущим временем

std::srand(static\_cast<unsigned int>(std::time(nullptr)));

// Заполнение массива случайными числами

for (int i = 0; i < arraySize; ++i) {

myArray[i] = std::rand() % 100; // Заполняем числами от 0 до 99

}

// Вывод элементов массива

for (int i = 0; i < arraySize; ++i) {

std::cout << "myArray[" << i << "] = " << myArray[i] << std::endl;

}

return 0;

}

Когда вы обращаетесь к элементу массива за его пределами, вы получаете непредсказуемое значение, которое зависит от текущего состояния памяти и компилятора. Это называется неопределенным поведением (undefined behavior).

В C++ массивы обычно располагаются в непрерывной области памяти, и каждому элементу массива соответствует определенный участок этой памяти. Когда вы обращаетесь к элементу за пределами массива, вы фактически читаете или записываете в участок памяти, который не является частью массива. Это может привести к различным последствиям, таким как чтение значений, оставшихся в этой памяти от предыдущих операций, или даже к ошибкам выполнения программы.

Задача № 2. Используйте созданный массив. Выведете среднее арифметическое значение элементов этого массива.

Пример кода задачи № 2:

#include <iostream>

int main() {

const int arraySize = 10;

int myArray[arraySize];

// Заполняем массив случайными числами

for (int i = 0; i < arraySize; ++i) {

myArray[i] = std::rand() % 100; // Заполняем числами от 0 до 99

}

// Вычисляем сумму элементов массива

int sum = 0;

for (int i = 0; i < arraySize; ++i) {

sum += myArray[i];

}

// Вычисляем среднее арифметическое

double average = sum / arraySize;

std::cout << "Среднее арифметическое значение элементов массива: " << average << std::endl;

return 0;

}