Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

ОТЧЕТ

по дисциплине «Операционные системы»

**Лабораторная работа № 5**

**Выполнил:**

Cтудент гр. 5130902/30002 Полин М.Д.

**Проверил**

Ст. преподаватель Я.А. Селиверстов

Санкт-Петербург

2024 г.

## **Задание**

**Лабораторная 5.  Контроль заданий на Степике.**

1.2. Раздел Ассемблер 1.

1. Напишите код, который обменивает значения в регистрах RSI и RDX. Т. е. если перед выполнением вашего кода в регистре RSI хранится число 1, а в регистре RDX хранится число 2, то после выполнения кода в регистре RSI должно храниться число 2, а в регистре RDX число 1.

Вам разрешено пользоваться следующими регистрами общего назначения: RAX, RBX, RCX, RDX, RBP, RDI, RSI, R8 - R15.

В задании не предполагается использование стека, даже если вы знаете, что это такое.

1. Сложите два числа в регистрах RSI и RDX, результат должен быть в регистре RSI. Вам разрешено пользоваться следующими регистрами общего назначения: RAX, RBX, RCX, RDX, RBP, RDI, RSI, R8 - R15. В задании не предполагается использование стека, даже если вы знаете, что это такое.
2. В регистре RSI вам дано целое число - градусы по шкале Фаренгейта (TF*TF*) Напишите код, который получит в регистре RSI соответствующее значение по шкале Цельсия (TC*TC*). Формула для перевода из шкалы Фаренгейта в шкалу Цельсия: TC=5×(TF−32)/9

**ВАЖНО:**﻿ в качестве операнда инструкций div и mul не может выступать просто число (a. k. a. immediate), т. е. вам придется загрузить делитель/множитель в регистр и использовать регистр.

Так как мы не рассматривали операции с нецелыми числами, нужно отбросить не целую часть результата и вернуть только целую. Кроме того гарантируется, что входное число и ответ будут беззнаковыми числами (более формально гарантируется, что  32≤TF≤220

Вам разрешено пользоваться следующими регистрами общего назначения: RAX, RBX, RCX, RDX, RBP, RDI, RSI, R8 - R15.

В задании не предполагается использование стека, даже если вы знаете, что это такое.

1. В регистрах RSI и RDX вам даны два числа, ваша задача поменять их местами, как и в одном из предыдущих заданий. Но добавляется условие, что все остальные регистры общего назначения должны остаться неизменными. Т. е. если вы используете какой-то регистр общего назначения кроме RSI и RDX, то вы должны сохранить и затем восстановить сохраненное значение регистра (ну или не пользоваться этими регистрами вовсе).

Вам разрешено пользоваться следующими регистрами общего назначения: RAX, RBX, RCX, RDX, RBP, RDI, RSI, R8 - R15. В задании предполагается использование стека, но вы должны восстановить стек в исходное состояние.

1.3 Язык ассемблера 2

1. А теперь ваша задача написать функцию swap. Функция принимает на вход два указателя на 64-битные числа (в регистрах RDI и RSI) и должна обменять значения в памяти.

**ВАЖНО:** ﻿функция принимает указатели на значения, которые нужно обменять, а не сами значения, т. е. не забудьте, что чтобы добраться до значений указатели нужно разыменовать.

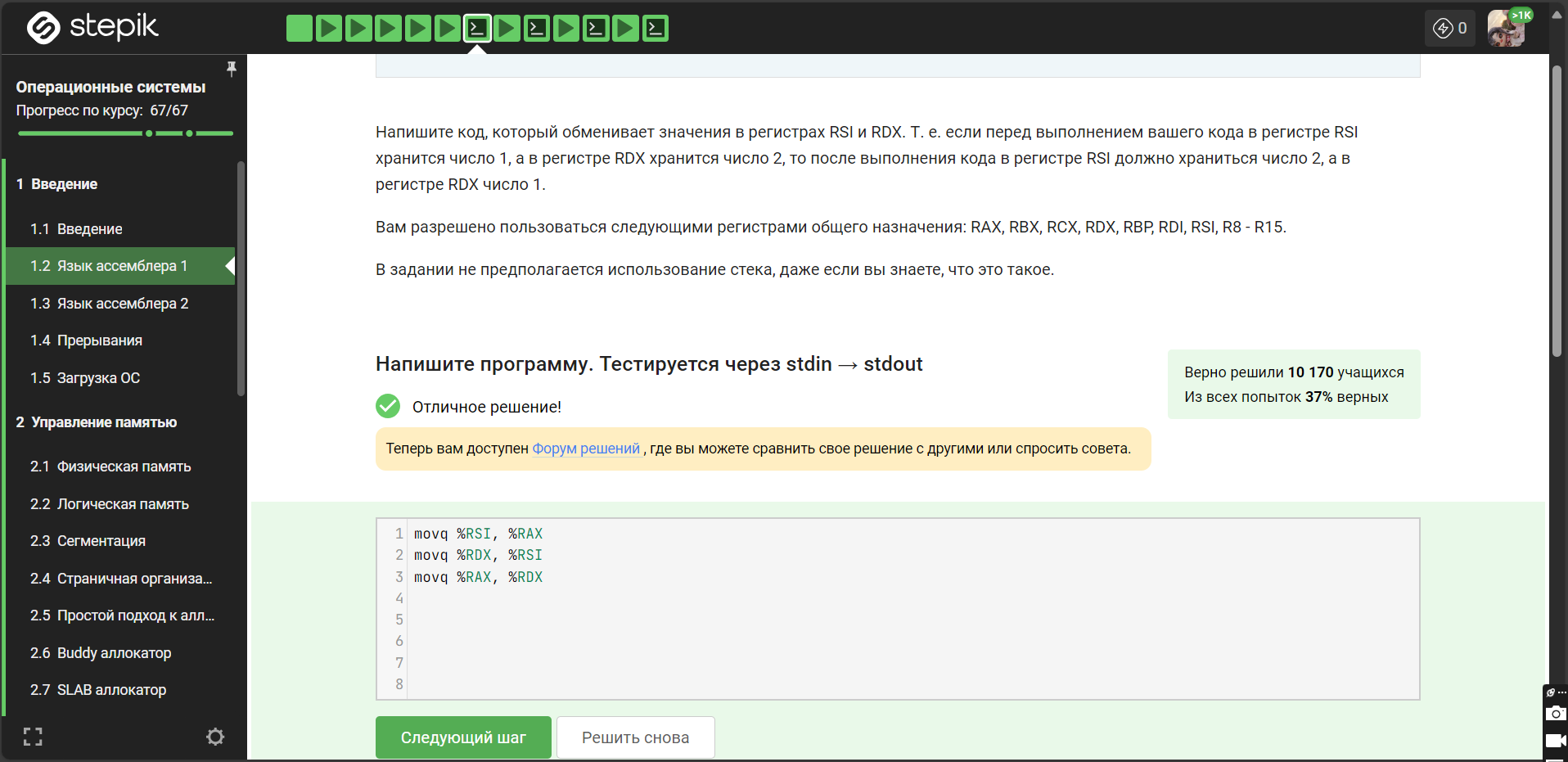
1. Попробуйте по аналогии с примером в лекции написать функцию, возвращающую минимум из двух переданных ей аргументов. Аргументы (беззнаковые целые числа) передаются в регистрах RDI и RSI. Результат работы функции должен быть сохранен в регистре RAX.
2. Теперь более сложное задание. Вам требуется написать функцию pow, которая принимает на вход два беззнаковых числа  xи  p (в регистрах RDI и RSI соответственно) и возвращает значение x в степени pв регистре RAX. Гарантируется, что  x и pне могут быть равны 0 одновременно (по отдельности они все еще могут быть равны 0). Также гарантируется, что ответ помещается в 64 бита.

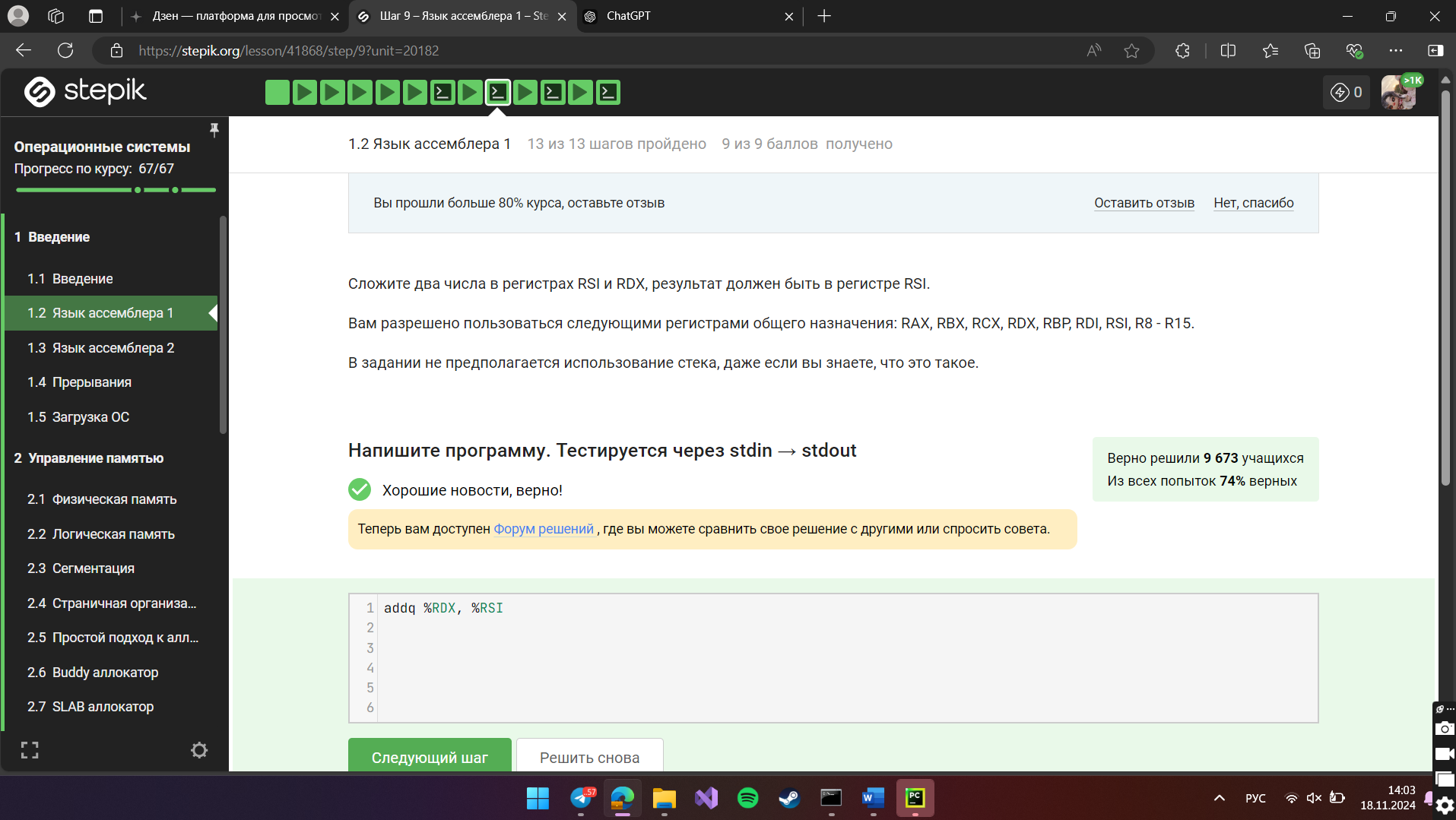
**ВАЖНО:** не забудьте, что кроме инструкций условного перехода (jcc) есть и инструкция безусловного перехода jmp - она может вам пригодиться.

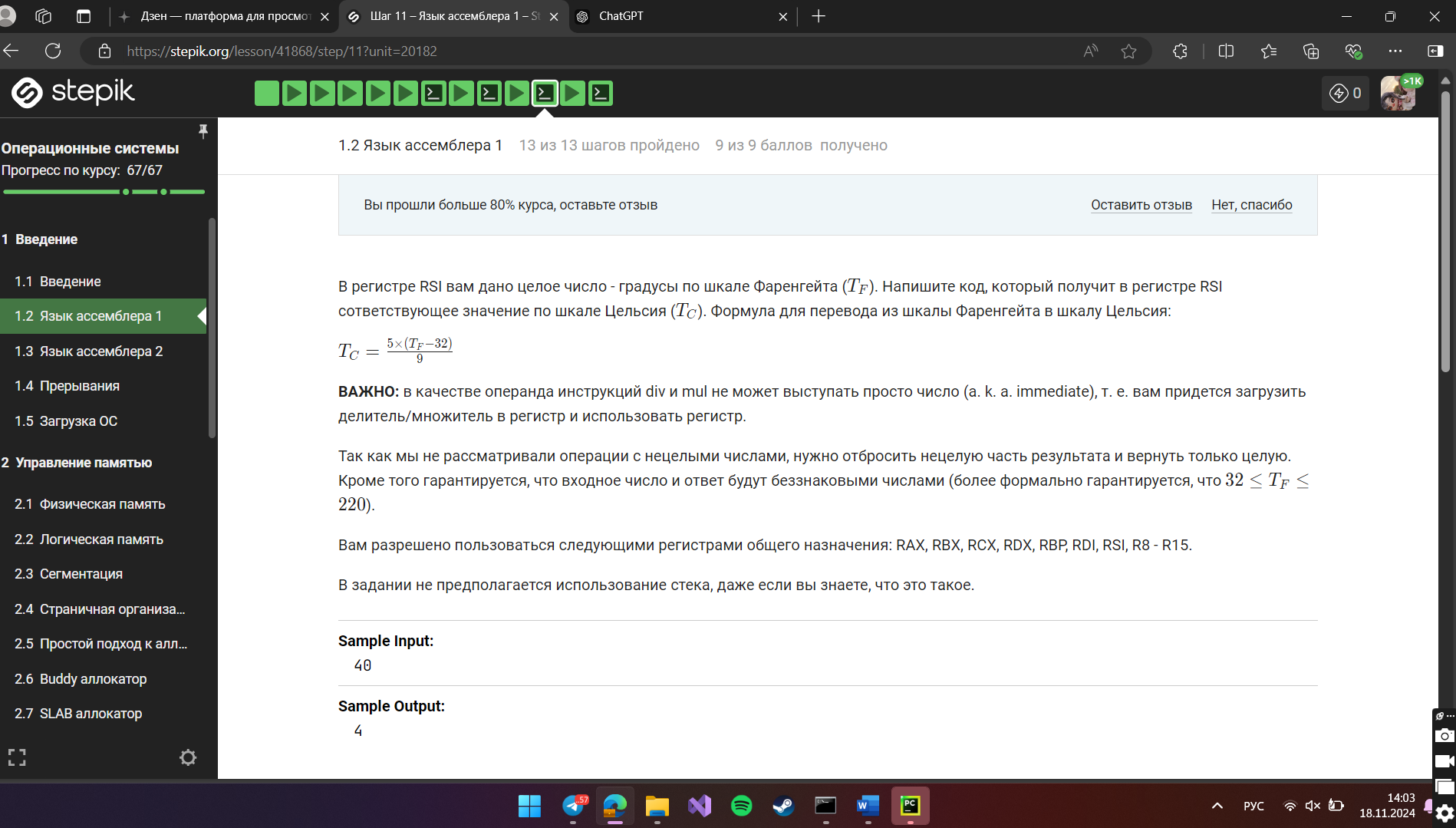
.

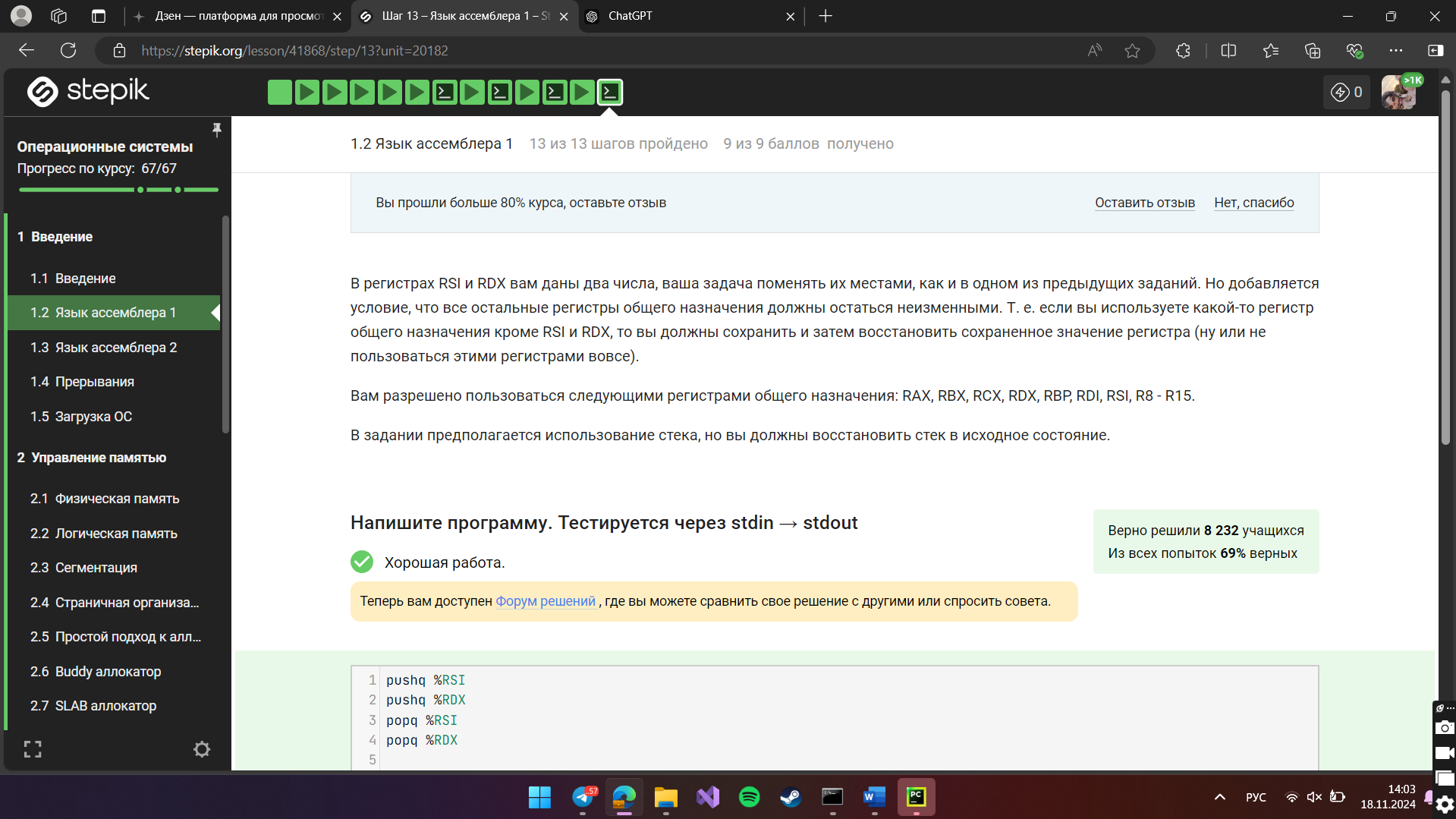
## **Выполнение программы**

**1.2. Раздел Ассемблер 1.**

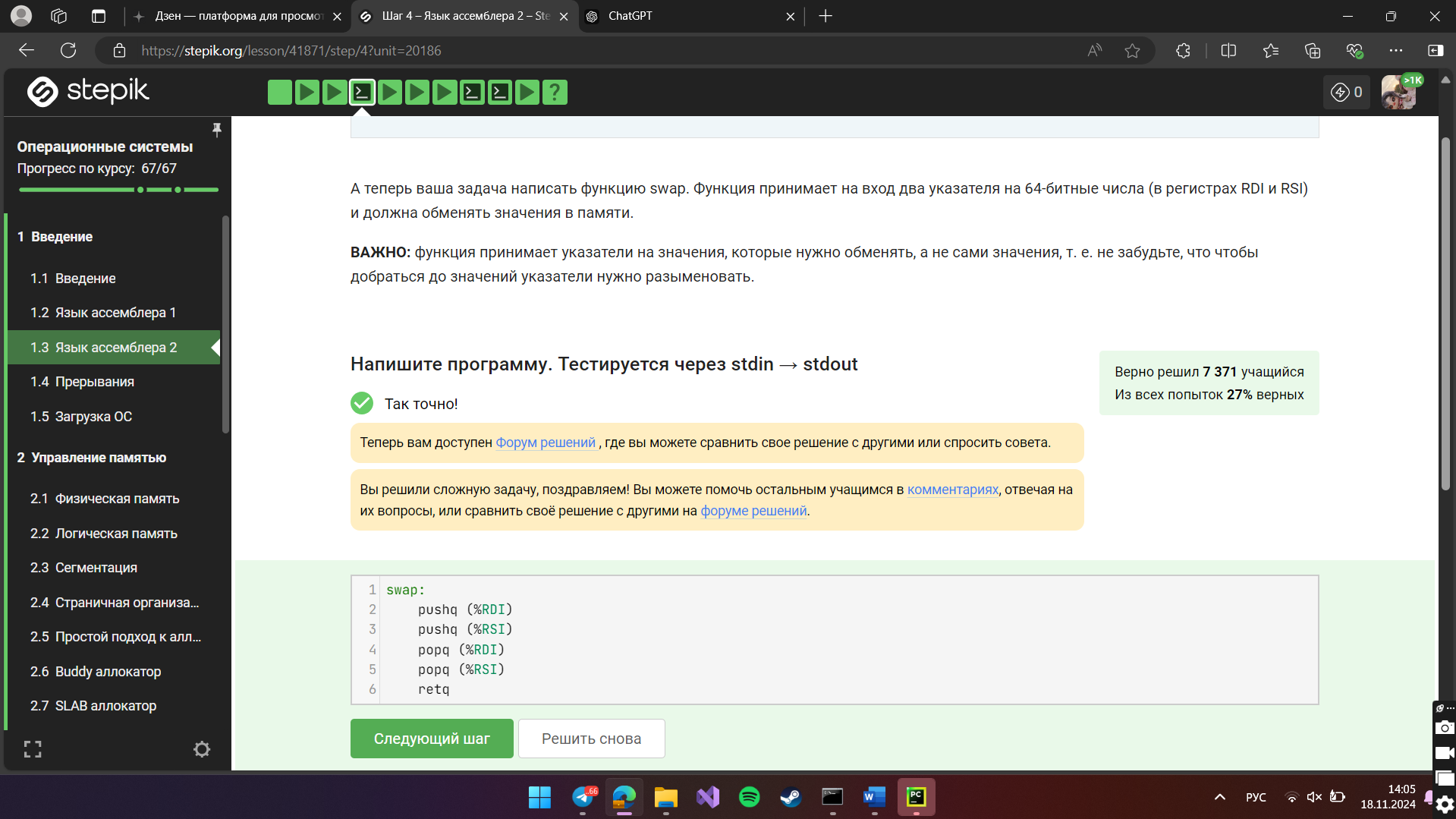
1) 

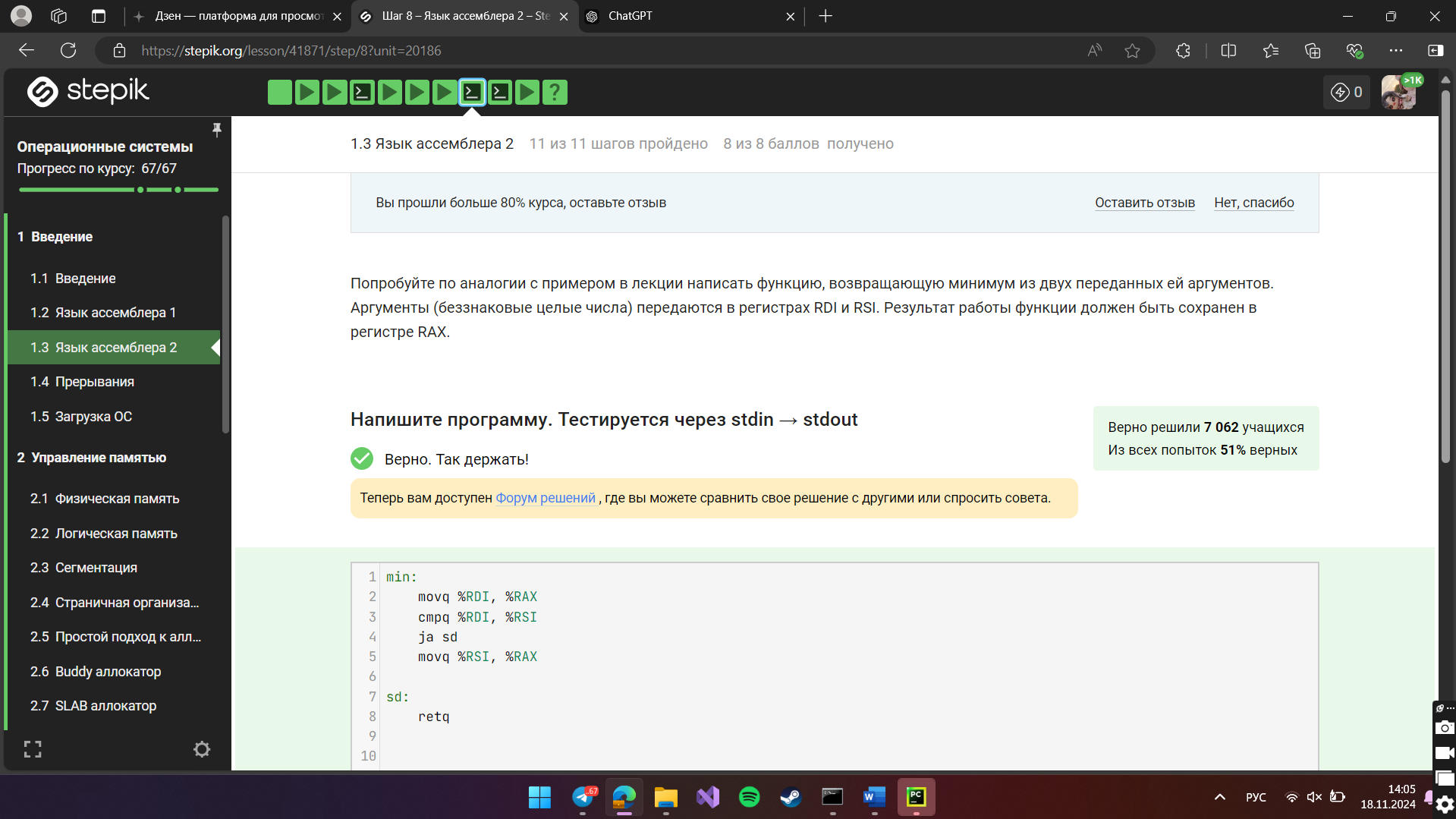
2) 

3) 

4) 

**1.3 Язык ассемблера 2**

1) 

2) 

3) 