

В этой лабораторной работе от вас потребуется разработать программу, которая реализует интерпретатор подмножества интерпретируемого языка Brainfuck. Более подробно о нём можно почитать, например, тут: <https://habr.com/ru/post/24605/>

Вы реализуете подмножество этого языка программирования, реализующее следующие команды (по вариантам):

Команды, общие для всех вариантов	
«+» - увеличить значение в текущей ячейке	
«-» - уменьшить значение в текущей ячейке	
«<» - перейти в предыдущую ячейку (если возможно)	
«>» - перейти в следующую ячейку (если возможно)	
«:» - проинициализировать значение в ячейке числом 110	
«!» - проинициализировать значение в ячейке числом 40	
Команды по вариантам	
Чётные варианты	Нечётные варианты
«*» - вывести на экран весь массив, начиная с текущей ячейки	«;» - вывести на экран текущую ячейку, перейти вправо
«~» - скопировать в текущую ячейку значение из ячейки слева	«/» - скопировать значение из первой ячейки во все ячейки массива

ЗАДАНИЕ ДЛЯ ВЫБРАВШИХ БАЗОВЫЙ УРОВЕНЬ

Три плюса за лабораторную работу:

1. вывести на экран таблицу ASCII символов;
2. реализовать в коде лабораторной работы программу на подмножестве языка Brainfuck, которая выведет на экран фразу "КАФЕДРА ПКИМС РУЛИТ" (без кавычек) при размере памяти виртуальной машины в 10 байт.
3. считывать код на языке Brainfuck из файла; при запуске программы запрашивать имя файла у пользователя и интерпретировать его содержимое.

При выполнении пункта 2 рекомендую использовать следующие элементы в вашем коде:

```
char memory[10];      – ячейки памяти виртуальной машины
int ip = 0;           – указатель на текущую ячейку
char code[] = "+..."; – код, который обрабатывается вашей виртуальной
машиной.
```

Важно! Должны быть задействованы все команды согласно варианту!

ЗАДАНИЕ ДЛЯ ВЫБРАВШИХ ПОВЫШЕННЫЙ УРОВЕНЬ

Необходимо выполнить все задания на минимум, реализовав код с использованием функций. После этого необходимо выполнить дополнительные задания, приведённые ниже.

Эти задания выполняются последовательно.

- **Плюс балл.** Ваш интерпретатор должен выдавать сообщения об ошибках и предупреждениях. Ошибки – то, что интерпретатор не может выполнить (например, встретился непонятный символ на входе), предупреждения – то, что вы попросили сделать, но интерпретатор предусмотрел, что может возникнуть проблема (например, выход за границы массива).

Сделайте лог красивым, адекватным (например, как в Visual Studio)

- **Плюс балл.** Имя файла должно не запрашиваться у пользователя при запуске, а должно передаваться как аргумент командной строки. Если аргумент не был задан – ругаемся и выходим.