

Практическое задание №1

В этом задании требуется написать небольшую программу на Cool. Целью является знакомство с этим языком программирования, а также получение опыта работы с некоторыми инструментами, используемыми в рамках курса. Эта лабораторная *не* может выполняться в группе, работа должна быть индивидуальной. Все последующие лабораторные по желанию могут делаться в команде из двух человек.

Машина с единственным стеком в качестве памяти называется *стековой машиной (stack machine)*. Рассмотрим следующий очень примитивный язык программирования для стековой машины:

Команда	Смысл
<i>int</i>	поместить целое число <i>int</i> на вершину стека
+	поместить «+» на вершину стека
s	поместить «s» на вершину стека
e	произвести вычисление над элементами с вершины стека (смотрите ниже)
d	отобразить содержимое стека
x	завершить работу

Команда «d» просто выводит содержимое стека, по одному элементу в строке, начиная с вершины. Поведение команды «e» зависит от содержимого стека на момент вызова этой команды:

- Если на вершине стека находится «+», то этот символ достается из стека, затем из стека достаются два целых числа, которые складываются, а результат сложения заносится обратно в стек.
- Если на вершине стека находится «s», то этот символ достается из стека, а следующие за ним в стеке два элемента меняются местами.
- Если на вершине стека находится целое число, или стек пуст, то ничего не делается.

Следующие примеры показывают действие команды «e» в различных ситуациях; вершина стека находится слева:

<i>стек до</i>	<i>стек после</i>
+ 1 2 5 s ...	3 5 s ...
s 1 + + 99 ...	+ 1 + 99 ...
1 + 3 ...	1 + 3 ...

Вам нужно реализовать на Cool интерпретатор для этого языка. Входом программы будет последовательность команд, по одной в строке; ваш интерпретатор должен запрашивать команду выводя символ «>». Вам не нужно осуществлять никаких проверок на ошибки: можете считать, что вводятся только допустимые команды, и в стеке всегда лежит нужное для вычислений число аргументов правильных типов. Также можете считать, что все вводимые целые числа являются беззнаковыми. Ваш интерпретатор должен нормально завершать работу; не вызывайте `abort()` после получения команды «x».

Вы можете использовать любой стиль при реализации этой программы, однако, в качестве подготовки к реализации компилятора для Cool мы рекомендуем вам попробовать разработать объектно-ориентированное решение. Одним из возможных подходов может быть определение класса `StackCommand` с некоторым числом общих операций, а затем определение его подклассов, по одному для каждой команды языка. Эти подклассы определяют операции, специфичные для каждой из

этих команд, такие как вычисление команды, ее отображение и так далее. Если хотите, можете использовать классы для осуществления преобразований между числами и строками, определенные в `atoi.cl`, находящемся в директории `/home/compilers_course/examples`. Если вы решите, что какой либо еще код из `/home/compilers_course/examples` будет вам полезен, можете использовать и его.

В рамках подготовки курса было написано решение в примерно 200 строк кода на Cool. Эта информация предоставляется в качестве примерного ориентира количества работы, необходимой для решения этого задания. Разумеется, ваше решение может быть как значительно короче, так и длиннее.

Пример сессии

Ниже приведен пример компиляции и запуска решения.

```
% coolc stack.cl atoi.cl
% spim -file stack.s
SPIM Version 5.6 of January 18, 1995
Copyright 1990-1994 by James R. Larus (larus@cs.wisc.edu).
All Rights Reserved.
See the file README a full copyright notice.
Loaded: /home/ff/cs164/lib/trap.handler
>1
>+
>2
>s
>d
s
2
+
1
>e
>e
>d
3
>x
COOL program successfully executed
```

Получение задания и его сдача

Создайте любую директорию, в которой будет вестись работа, например `PA1`, и перейдите в нее. Далее, находясь в этой директории, выполните следующую команду:

```
% make -f /home/compilers_course/assignments/PA1/Makefile
```

Эта команда создаст в директории несколько нужных вам файлов. Далее следуйте инструкциям из файла `README`.

Для сдачи задания:

- 1) Убедитесь, что ваш код находится в файле `stack.c1` и он компилируется и работает.
- 2) Укажите ваше имя в соответствующем месте файла `README`.
- 3) Ответьте на три вопроса в этом же файле.
- 4) Выполните команду `make submit` из вашей директории.

Обратите внимание, что ваша стековая машина будет тестироваться путем сравнения ее вывода с выводом эталонной реализации. Поэтому ваша программа не должна выдавать ничего кроме пробельных символов (которые будут игнорироваться), приглашения «>» и вывода команды «d». Перед отправкой решения удалите все команды вывода, использующиеся для отладки.