Выполнил Саркисянц Юрий, студент группы СКБ221

Отчет

# Как работает программа

В начале, при запуске программы, требуется ввести: «**--forward**» или «**--reverse**». После запуска необходимо ввести пример в инфиксной или постфиксной форме соотственно, нажать Ctrl+D и программа запуститься

Если мы хотим считать с потока cin, то при запуске необходимо ввести флаг «**--forward**» или «**--reverse**». Если хотим считать с файла, то так же вводим те же флаги, что и в первом случае, но добавляем «--file» и название файла с форматом. Далее вводим выражение в инфиксной форме без пробел, в постфиксной — используя их.

# Как устроен алгоритм

## Функция reverse()

Необходимо, чтобы были введены хотя бы 2 числа и какая-либо операция.

1. Объявляем переменные **с1** и **с2** типа **integer** и **operation** типа **char**.
2. Считываем целое число **c1**.
3. Пока можно считать число **c2**.

a)Считываем целое число **c2** и оператор действия **operation**.

b)Выполняем соответствующее действие и результат записываем в **c1**

1. Выводим значение **c1**

## Функция forward()

Необходимо, чтобы были введены хотя бы 2 числа и какая-либо операция.

1) Объявляем два стека **SymStack** и **ValStack** — символов и значений в виде статических массивов типа char и integer соответственно. Затем создаем переменные, представляющие собой индексы этих массивов — **iVal** и **iSym**. Также необходимо создать переменную token, где будет храниться каждый считанный нами символ.

2) Пока токен существует

a) если он целый, то преобразовываем его в тип int и пусть он будет последним в стеке ValStack.

b)если это операция, то

b.1) пока в стеке что-то есть и символ, который является последним, имеет приоритет не меньший считанному только что символу, то берем это значение из стека SymStack, присваиваем в operaion, также берем из **ValStack** 2 последних числа, присваиваем их в **c1** и **c2**, и выполняем опервцию над этими элементами, результат записываем в стек ValStack.

b.2) кладем эту операцию в **SymStack**

1. 3) Если **token** – открывающаяся скобка, то кладем ее в стек S**ymStack**.

4) Если **token** – закрывающаяся скобка.

1. Пока символ на стеке **SymStack** не равен открывающейся скобке, вытаскиваем **operation** из стека **SymStac** и числа **c1** и **c2** из стека **ValStack**, выполняем математическую операцию, результат кладем в стек **ValStack**.
2. Вытаскиваем открывающуюся скобку из стека **SymStack**.
3. Пока **SymStack** не пуст, вытаскиваем **operation** из стека **SymStack** и числа **c1** и **c2** из стека **ValStack**, выполняем математическую операцию, результат кладем в стек **ValStack**.
4. Выводим результат, лежащий в **ValStack** (он является единственным числом в данном стеке)

## Функция main()

Ход функции выглядит следующим образом:

1. Если никакие флаги не были введены пользователем, то выводится ошибка «**You don’t enter flag**» и программа прекращает выполнение.
2. Идет поочередный перебор введенных флагов.
3. Если введен только один флаг, то выполняется **reverse()** или **forward()** соответственно.
4. Если первым флагом ввести --forward или --reverse, затем ввести —file <название файла>, то выполняется чтение с файла и **reverse()** или **forward(),** в зависимости от того, какой флаг был введен первым.
5. В ином случае выводится ошибка «**You entered wrong flag**» и программа прекращает выполнение.
6. Программа прекращает выполнение.

# Вывод

В результате выполнения лабораторной работы №4 я научился работать с потоком ввода и вывода, научился писать заголовочные файлы, make-файлы, также научился работать c системой Git и сумел написать консольное приложение для считывания из потока/файла.