Считываем строку, введенную пользователем(предполагаем, что это корректное выражение).

Сначала проверяем на корректность скобочной последовательности с помощью функции checkbrackets(string), которая принимает в качестве параметра нашу строку.

Если скобочная последовательность неверная, то выводится сообщение об этом и программа завершает работу.

Иначе вызываем функцию getRPN(string), которая также принимает на вход строку и возвращает нам наше выражение, преобразованное в обратную польскую запись.

Описание функции getRPN(string):

Для хранения операторов используется стек. Для хранения представления обратной польской записи будем использовать строку rpn, которая изначально пустая. В цикле мы пробегаемся по элементам выражения:

* Если символ равен пробелу, то пропускаем и переходим на следующую итерацию
* Если символ равен цифре, то вызываем функцию getstring(string, int&), которая принимает наше выражение и текущую позицию в строке(мы передаем ее по ссылке для того, чтобы она также изменялась и по окончанию работы функции getstring(string, int&) мы продолжили с того места, где она закончила). По окончанию работы функции getstring(string, int&) мы получим текущее число в строковом типе. Добавляем его в нашу строку rpn и добавляем пробел для более читабельного вида наше записи, чтобы все не сливалось.
* Если символ равен открывающейся скобке, то добавляем ее в наш стек операторов.
* Если символ равен закрывающейся скобке, то в цикле, пока стек с операторами не пуст или пока верхний элемент стека не равен открывающейся скобке, мы добавляем в нашу строку оператор и извлекаем его из стека, между операторами ставим пробел(опять же для лучшей читаемости нашей записи). По окончанию цикла извлекаем открывающуюся скобку.
* Иначе проверяем на то, является ли этот символ оператором(+, -, /, \*), если да, то пробегаем в цикле и смотрим: пока стек не пуст и пока в стеке верхний элемент это тоже оператор и пока приоритет у верхнего элемента больше или равен приоритету текущего оператора, который мы встретили, то мы добавляем в нашу результирующую строку rpn верхний оператор стека и извлекаем его, затем добавляем пробел в строку rpn. По окончанию цикла добавляем наш текущий оператор в стек.
* Если наша строка с выражением закончилась, а стек еще не пуст, то мы добавляем операторы в строку rpn с извлечение их из стека.

По окончанию работы функции мы получим наше выражение, представленное в виде обратной польской записи.