Задание 1

В жизни каждого приличного студента-программиста в ОТИ НИЯУ МИФИ рано или поздно наступает важное событие в жизни. Студенту поручают выполнять задания по заказу «Предприятия», о деятельности которого ничего не известно. Известно лишь, что предприятие занимается настолько важными исследовательскими делами, что более значимой деятельности не существует.

С сегодняшнего дня вы являетесь исполнителем заданий «Предприятия».

Прежде, чем поручать кандидату исполнять серьёзные задания, предприятие предпочитает испытать кандидата на некоторых тренировочных примерах, на которых кандидат сможет проявить себя с лучшей стороны.

В настоящее время отдел разбора сложных математических выражений департамента математических выражений работает над разбором сложных математических выражений, которые предоставляет отдел предоставления сложных математических выражений департамента математических выражений. В процессе своего становления отдел предоставления сложных математических выражений предоставлял сложные математические выражения с простой структурой (предоставляемые выражения могли включать скобки одного вида), но несмотря на простую структуру, эти выражениях могли содержать ошибки – не для каждой открывающей скобки была закрывающая скобка, или скобки шли в неверном порядке, например: a+(a-)b\*c)/(c-(d+a)/)f-s()). Для того, чтобы определять, является ли предоставленное выражение корректным, отделом решения задач программным способом департамента разработки программного обеспечения было разработано программное обеспечение, определяющее, содержит ли математическое выражение допустимую скобочную структуру. Приложение работало по простому алгоритму: в начала работы специальная переменная имела значение 0; когда в выражении встречалась открывающая скобка, переменная увеличивала своё значение на 1; когда в выражении встречалась закрывающая скобка, переменная уменьшала своё значение на 1. Если в процессе работы программы значение переменной становилось отрицательным, или в конце работы программы значение переменной не было равно 0, делался вывод, что скобки в выражении расставлены неверно. Всё было хорошо до тех пор, пока отдел предоставления сложных математических выражений не вышел на новый виток развития и не стал предоставлять выражения, содержащие до 4-х видов скобок, например:  
[a+[b/(c-d+{a-b-c}\*[a\*(b\*{a-d-<a/n>})])]]. Проблема с путаницей в расстановке скобок оставалась, но ранее используемая программа была рассчитана на «отлавливание» только одного вида скобок.

Руководство «Предприятия» получило информацию, что вы начали изучать структуры данных, и приняло решение испытать вас, дав вам задание разработать программу, определяющую, верно ли расставлены скобки в математическом выражении, содержащем несколько видов скобок.

Руководство «Предприятия» поставило задачу написать программу, позволяющую определять, используя стек, верно ли расставлены скобки в математическом выражении, содержащем несколько видом скобок.

Отдел постановки задач департамента разработки программного обеспечения сформулировал для вас как для исполнителя следующие требования в техническом задании.

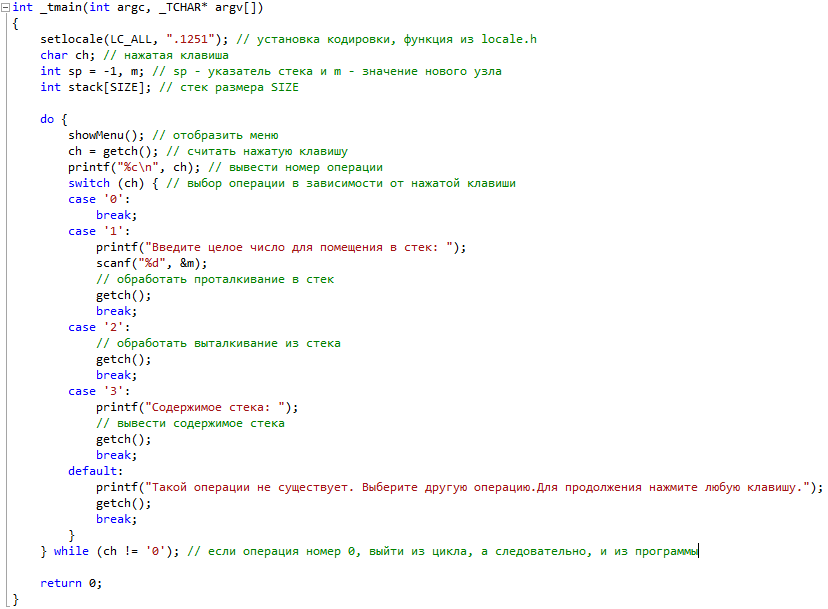
**Часть 1.**

1) Программа должна иметь имя 2022\_SAS01\_01\_Фамилия, где фамилия – фамилия автора программы на латинице.

2) Программа должна использовать программный код, разработанный непосредственно исполнителем.

3) Структура программы должна быть следующей: главная функция main выводит меню, определяет, какую операцию хочет выполнить пользователь, и в зависимости от выбранной операции вызывает нужную функцию.

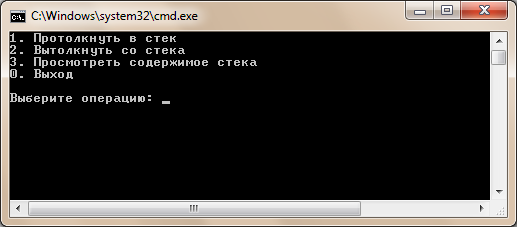
4) Общая схема программы должна быть такой:



5) Описывая любую функцию, следует исходить из того, что функция при вызове принимает фактические параметры, в процессе работы выполняет некоторые вычисления и преобразования с использованием этих параметров и в конце работы возвращает результат при помощи оператора **return**. **Внутри функции не должен происходить ввод и вывод информации** (если только эта функция специально не предназначена для ввода и вывода). Функция принимает входные параметры из вне, и возвращает результат работы.

6.1) Размер стека задается константой макросом SIZE. Макрос определяется так: #define SIZE 5. После такого определения на одной из стадий компиляции все вхождения SIZE в программу будут заменены на 5.

6.2) Функция showMenu определена так: void showMenu(). Единственное, что делает эта функция – выводит на экран меню программы:



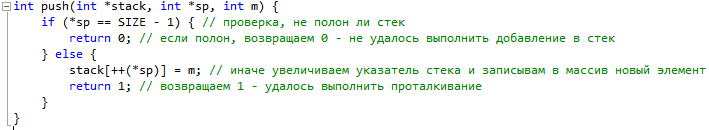
6.3) Для проталкивания в стек должна использоваться функция push. Она может быть описана, к примеру, так (но по вашему усмотрению может быть описана иначе с учетом приведенных ранее требований): **int** push(**int** \*stack, **int** &sp, **int** m). Данная функция помещает m в стек и возвращает 1 в случае, если удалось поместить новый элемент в стек, и возвращает 0, если не удалось (стек полон).

6.4) Для выталкивания должна использоваться функция, описанная как, например, так (но по вашему усмотрению может быть описана иначе с учетом приведенных ранее требований): **int** pop(**int** \*stack, **int** &sp), которая производит выталкивание элемента из стека и возвращает его значение.

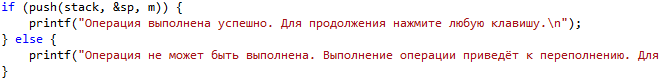
6.5) Функция для вывода содержимого стека описывается как **void** view(**int** \*stack, **int** sp).

7) Следует обратить внимание на методы передачи параметров в функции на примере функции push. Особенность данной функции в том, что она, кроме того, что помещает новый элемент в стек, ещё и изменяет значение указателя sp.

Функция push может выглядеть так:



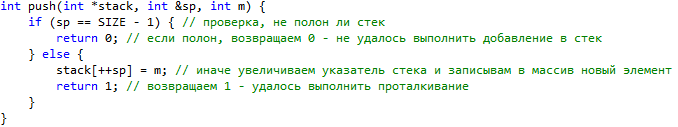
А вызов её так:



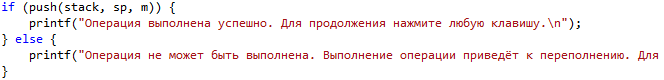
Переменная stack принимается как указатель, т.к. она есть массив. Указатель стека sp принимается как указатель, т.к. при изменении его внутри функции, он должен измениться и вне функции.

Работа с указателями не является безопасной. Но в языке С такой способ передачи параметров, чтобы значение изменялось и внутри функции, и вне функции, был единственным возможным.

В языке С++ появился новый способ передачи параметров «по ссылке». Описание функции, принимающей параметр по ссылке выглядит так:



А вызов её так:



Ссылку в С++ можно понимать или как альтернативное имя объекта, или как безопасный вариант указателей. Ссылки имеют три особенности, отличающие их от указателей:

- при объявлении ссылка обязательно инициализируется ссылкой на уже существующий объект данного типа. Ссылка (как и указатель) не может использоваться без инициализации;

- ссылка пожизненно указывает на один и тот же адрес;

- при обращении к ссылке операция \* производится автоматически.

Объявление ссылок очень похоже на объявление указателей, только вместо звёздочки «\*» нужно писать амперсанд «&».

Что произошло в описании выше? Аргумент sp функции push стал не указателем, а ссылкой. Поэтому теперь:

- при вызове функции push компилятор сам передаст адрес переменной sp, нет необходимости специально его просить;

- внутри функции push следует обращаться с sp, как с обычной переменной: только компилятор знает, что внутри это — указатель.

8.1) Результат выполнения операции проталкивания нужно проверить, и в зависимости от результата вывести сообщение, удалось выполнить проталкивание, или нет;

8.2) Перед выталкиванием из стека нужно убедиться, не пуст ли стек. Для этого можно создать новую функцию **int** isEmpty(**int** sp), которая принимает указатель стека sp и проверяет, не равен ли он -1. В случае, если равен – возвращает 1, иначе – 0.

8.3) При выводе содержимого стека также нужно проверить, не пуст ли стек. И только в случае, если не пуст, выполнять вывод содержимого.

8.4) Функции push и pop принимают sp как ссылку, т.к. они изменяют значение sp. Функции isEmpty и view принимает sp как простую переменную, т.к. они не должны изменять её значение.

**Часть 2.**

1) Программа должна иметь имя 2022\_SAS01\_02\_Фамилия, где фамилия – фамилия автора программы на латинице.

2) Программа должна использовать функции для работы со стеком, разработанные в программе 2020\_SAS01\_01\_Фамилия.

3) Работа программы осуществляется в интерактивном режиме посредством меню (программа должна иметь дружественный интерфейс):

1. Ввод выражения

2. Проверка выражения

3. Выход

Результат работы программы выводится в консоль.

4.1) При выбора пункта «1» программа запрашивает строку – математическое выражение, содержащее скобки не более 4-х видов ( (), [], {}, <> ), буквы латинского алфавита в верхнем и нижнем регистре, цифры и знаки арифметических операций ( +, -, \*, /).

4.2) При выборе пункта 2 программа анализирует строку и выдаёт результат: «Верно» либо «Неверно».

4.3) При выборе пункта 3 программа завершает работу.

5) Для определения, верно ли записано выражение, программа использует стек: (1) когда в выражении встречается открывающая скобка, она кладётся на стек; (2) когда в выражении встречается закрывающая скобка, проверяется, какая скобка находится на вершине стека: (2a) если стек не пуст и на вершине находится открывающая скобка того же вида, то она снимается со стека, (2b) если стек пуст или на вершине находится скобка другого вида, делается вывод, что выражение имеет неверную структуру, дальнейший анализ прекращается; (3) когда символы строки заканчиваются, а стек не пуст, делается вывод, что выражение имеет неверную структуру.

Оценка выполнения каждого задания (2022\_SAS01 и 2022\_SAS02) производится следующим образом:

Оценка 5 (отлично): все пункты задания выполнены полностью, программа является работоспособной, исполнитель обоснованно отвечает на контрольные вопросы.

Оценка 4 (хорошо): все пункты задания выполнены полностью, программа является работоспособной, исполнитель хорошо отвечает на контрольные вопросы, но не всегда может обосновать их.

Оценка 3 (удовлетворительно): пункты задания выполнены частично, программа является частично работоспособной, исполнитель затрудняется в ответах на контрольные вопросы.

Оценка 2 (неудовлетворительно): пункты задания выполнены частично, программа не является работоспособной, исполнитель затрудняется в ответах на контрольные вопросы.

Соответствие оценок баллам балльно-рейтинговой системы при оценивании выполнения практической работы:

Оценка 5 (отлично) – 2 балла;

Оценка 4 (хорошо) – 1,5 балла;

Оценка 3 (удовлетворительно) – 1 балл;

Оценка 2 (неудовлетворительно) – 0 баллов.