**1. brackets.jar**

Если из корректно записанного арифметического выражения, содержащего числа, знаки операций и открывающие и закрывающие круглые скобки, выбросить числа и знаки операций, а затем записать оставшиеся в выражении скобки без пробелов между ними, то в результате получится правильное скобочное выражение [скобочное выражение "(()(()))" - правильное, а "()(" и "())(" - нет].

N - неотрицательное целое число, вводится с клавиатуры. Найти число правильных скобочных выражений, содержащих N открывающих и N закрывающих скобок. Вывести скобочные выражения.

**2. mincostpath.jar**

You are given a list of cities. Each direct connection between two cities has its transportation cost (an integer bigger than 0). The goal is to find the paths of minimum cost between pairs of cities. Assume that the cost of each path (which is the sum of costs of all direct connections belonging to this path) is at most 200000. The name of a city is a string containing characters a,...,z.

**Input**

*s* [the size of a set of minimum cost paths to find for pairs of cities <= 10]

*n* [the number of cities <= 10000]

*NAME* [city name]

*p* [the number of neighbors of city *NAME*]

*nr* *cost* [*nr* - index of a city connected to *NAME* (the index of the first city is 1)]

[*cost* - the transportation cost]

*r* [the number of pairs <= **100**]

*NAME1 NAME2* [*NAME1* - source, *NAME2* – destination for the pair]

**Output**

*cost* [the minimum transportation cost from city *NAME1* to city *NAME2* (one per line)]:*path*

[empty line separating the sets of paths]

**Example of the data (data\_for\_mincostpath.txt)**

Input:

1

4

gdansk

2

2 1

3 3

bydgoszcz

3

1 1

3 1

4 4

torun

3

1 3

2 1

4 1

warszawa

2

2 4

3 1

2

gdansk warszawa

bydgoszcz warszawa

**3. smalleditor.jar**

Небольшой текстовый редактор:

- ввод текста (набора строк) в консоли;

- вывод текста в консоль или назначенный файл;

- логгирование в консоль или лог-файл по умолчанию logfile.log;

- выход из редактора – двойной ENTER;

- опции вывода и логгирования – в аргументах командной строки.

**4. sumfact.jar**

Find the sum of the digits in the number 100! (i.e. 100 factorial). {Correct answer: 648}