

Московский авиационный институт (национальный исследовательский университет)  
Вступительный междисциплинарный экзамен в магистратуру  
по направлению подготовки  
01.04.02 «Прикладная математика и информатика»  
Магистерская программа  
«Математическое и программное обеспечение вычислительных систем»  
Билет 35

1	Задача на программирование. Язык C++. Напишите программу, которая просит пользователя ввести целое число - палиндром. Сгенерируйте исключение, если число не является палиндромом. Обработайте исключение, в обработчике уведомите пользователя о том, что число не палиндром.				
2	Задача на программирование. Языки C, C++, или Java. Составить программу определения числа различных вершин дерева.				
3	<p>Задача на программирование. Языки C, C++ или Java. Составить программу решения задачи. Сережа – начинающий бизнесмен, изучив последние тенденции в мире, он решил, что фрилансу решительно не хватает ещё одной системы по поиску похожих программ. Так как Сережа – математик и не очень хорошо программирует, он нанял вас, чтобы вы реализовали данную систему. И хотя Сережа не программист, он отлично знает, как должна работать система поиска. Программа фрилансера будет по очереди сравниваться со всеми программами из интернета и для каждой программы из интернета будет выдаваться её сходство с данной программой. Сходством будет являться минимальное количество операций вставки знака на произвольную позицию, удаления любого знака, либо замены любого знака на другой, применённых к общедоступному тексту, необходимых для получения программы фрилансера. Для начала в качестве минимального прототипа от вас требуется реализовать функцию нахождения сходства двух программ.</p> <p><i>Входные данные</i> В двух строках заданы тексты двух программ P и Q, состоящие из строчных латинских букв.</p> <p><i>Выходные данные</i> Выведите минимальное количество операций вставки, удаления и замены, которые необходимо применить к программе P, чтобы получить программу Q.</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Пример входных данных</td><td style="width: 50%;">Пример выходных данных</td></tr> <tr> <td>from for</td><td>3</td></tr> </table> <p>Ожидаемая временная сложность алгоритма: <math>O( P  Q )</math>, где <math> S </math> — длина строки S.</p>	Пример входных данных	Пример выходных данных	from for	3
Пример входных данных	Пример выходных данных				
from for	3				
4	<p>Задача на программирование. Языки C, C++ или Java. Составить программу решения задачи. Владик не только отличный программист, но и восхитительный физик. Недавно он создал вечный двигатель, работа которого основывается на открытом недавно физико-техническом эффекте. Так как ему никто не верит, он решил представить свой двигатель в виде набора состояний и набора возможных переходов. Такое описание должно существенно упростить проверку его гипотезы. А пока Владик будет заниматься описанием двигателя, он попросил вас реализовать программу, которая будет проверять, является ли его двигатель вечным или нет.</p> <p>Двигатель называется вечным, если существует замкнутый путь по его состояниям, при прохождении которого двигатель генерирует больше энергии чем тратит.</p> <p><i>Входные данные</i> В первой строке даны два числа N и M — количество состояний в двигателе и количество возможных переходов между состояниями. В следующих M строках даны описания возможных переходов между состояниями в виде троек <math>U_i, V_i</math> (<math>1 \leq U_i, V_i \leq N</math>) и <math>E_i</math> — номера состояний, между которыми происходит переход, и количество энергии, которое производится при этом переходе, отрицательное <math>E_i</math> означает, что двигатель поглощает <math>-E_i</math> единиц энергии на переход между этими состояниями. Все переходы являются однонаправленными.</p> <p><i>Выходные данные</i> В случае, если двигатель Владика действительно является вечным, выведите “1” в противном случае выведите “0” (без кавычек).</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Пример входных данных</td><td style="width: 50%;">Пример выходных данных</td></tr> <tr> <td>4 4 1 2 -1 2 3 -1 3 4 -10 4 1 100</td><td>1</td></tr> </table> <p>Ожидаемая временная сложность алгоритма: <math>O(NM)</math>.</p>	Пример входных данных	Пример выходных данных	4 4 1 2 -1 2 3 -1 3 4 -10 4 1 100	1
Пример входных данных	Пример выходных данных				
4 4 1 2 -1 2 3 -1 3 4 -10 4 1 100	1				

Максимальная оценка 100 баллов ставится за четыре полностью решённых задачи.