Homework 7 RUS

**Степень**

Напишите рекурсивную функцию int power(int a, int n) которая считает an.

|  |  |
| --- | --- |
| 220 0 | 1 |
| 1023 1 | 1023 |
| 2 11 | 2048 |

**Факториал**

Напишите рекурсивную функцию int factorial(int n), которая считает факториал.

|  |  |
| --- | --- |
| 5 | 120 |
| 0 | 1 |
| 2 | 2 |

**Делимость**

Вход программы имеет следующий формат. Сначала вводится N > 0- количество чисел в последовательности А. Далее идут N целых чисел. После этого следует число M > 0 - количество чисел в последовательности В. Далее следуют M целых **положительных**числа. Требуется найти и напечатать те числа из последовательности А, которые делятся на *все*числа из последовательности B. Сначала вывести количество таких чисел, а на следующей строчке сами эти числа, разделенные пробелом (можно и в конце пробел ставить).

|  |  |
| --- | --- |
| Ввод | Вывод |
| 5  100 200 150 0 -25  4  5 50 25 20 | 3  100 200 0 |
| 6  1 2 3 4 5 6  2  2 3 | 1  6 |
| 3  7 8 9  7  1 2 3 4 5 6 7 | 0 |
| 2  99 999  7  1 3 1 9 1 3 3 | 2  99 999 |
| 5  1 2 3 4 5  1  2 | 2  2 4 |

**Требование**. Посчитайте НОК чисел второй последовательности. Напишите функции int gcd(int a, int b) и int lcm(int a, int b). Можете также написать фунцкию int lcm(const std::vector<int> & numbers), которая считает НОК всей последовательности. Да, в C++ несколько функций могут иметь одно и то же название, если у них разные количество или типы аргументов.

**Программа “Числоотгадывательница - 10”**

Программа Числоотгадывательница-10 замечательна во-первых тем, что не все могут произнести ее имя. Ну а во-вторых она удивительна тем, что умеет безошибочно отгадывать числа загаданные пользователем не более чем за 10 попыток (предполагается, что загадано целое положительное число меньшее, чем 1000). Программа пытается угадать число. Если ей это удалось, пользователь должен ввести 0. Если она сказала слишком маленькое число, пользователь должен ввести 1. Если она сказала слишком большое число, то пользователь должен ввести -1. После этого программа снова делает попытку и так до тех пор пока не отгадает число (т. е. пользователь введет 0) или же пока не истратила все 10 попыток. При этом если после десятой попытки пользователь опять говорит, что число не отгадано, программа уверена, что значит в какой-то момент пользователь ввел неправильную информацию, то есть сжульничал, и сообщает ему об этом. Ваша задача написать программу-числоотгадывательницу-10. Вот пример работы программы:

Программа выводит: **Загадайте целое число из интервала [1, 999].                                                                                                                                                                                                                                                                      Введите 0 когда будете готовы,**

Пользователь загадывает число. (предположим я загадал 307). Пользователь вводит **0**.

Программа выводит. **Моя попытка номер 1: 500**

Так как 307 < 500, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 2: 250**

Так как 307 > 250, пользователь вводит  **1**

Программа выводит. **Моя попытка номер 3: 375**

Так как 307 < 375, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 4: 313**

Так как 307 < 313, пользователь вводит  -**1**

Программа выводит. **Моя попытка номер 5: 282**

Так как 307 > 282, пользователь вводит  **1**

Программа выводит. **Моя попытка номер 6: 298**

Так как 307 > 282, пользователь вводит  **1**

Программа выводит. **Моя попытка номер 7: 306**

Так как 307 > 306, пользователь вводит  **1**

Программа выводит. **Моя попытка номер 8: 310**

Так как 307 < 310, пользователь вводит  -**1**

Программа выводит. **Моя попытка номер 9: 308**

Так как 307 < 308, пользователь вводит  -**1**

Программа выводит. **Моя попытка номер 10: 307**

Так как 307 = 307, пользователь вводит  0

Программа выводит. **Я уложилась в 10 попыток!**

Теперь рассмотрим пример, когда пользователь пытается обмануть программу.

Программа выводит: **Загадайте целое число из интервала [1, 999].                                                                                                                                                                                                                                                                      Введите 0 когда будете готовы,**

Пользователь загадывает число. (предположим я загадал 600). Пользователь вводит **0**.

Программа выводит. **Моя попытка номер 1: 500**

Так как 600 > 500, пользователь вводит  **1**

Программа выводит. **Моя попытка номер 2: 750**

Несмотря на то, что  600 < 750, пользователь вводит  **1**

Программа выводит. **Моя попытка номер 3: 875**

Так как 600 < 875, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 4: 813**

Так как 600 < 813, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 5: 782**

Так как 600 < 782, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 6: 766**

Так как 600 < 766, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 7: 758**

Так как 600 < 758, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 8: 754**

Так как 600 < 754, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 9: 752**

Так как 600 < 752, пользователь вводит  **-1**

Программа выводит. **Моя попытка номер 10: 751**

Так как 600 < 751, пользователь вводит  **-1**

Программа выводит. **Я не отгадала за 10 попыток. Значит вы жульничали!!!**

Понятно, что выводы программы надо печатать транслитом или на английском и не обязательно именно так, как в примерах.

**Лесенки\***

У вас N одинаковых кубиков. Если всех их положить друг на друга, получится башенка высоты N.  Можно также вместо одной длинной башенки составить несколько маленьких. Например, если имеется 6 кубиков, то можно составить “лесенку” из трех башенок - с высотой 1, 2, и 3. Формально, лесенкой назовем последовательность башенек, каждая из которых строго выше предыдущей. Таким образом из 6 кубиков можно составить 4 различных лесенoк.

6

1 5

2 4

1 2 3

Из 8-и кубиков можно составить  6 различных лесенок.

8

1 7

2 6

3 5

1 2 5

1 3 4

По заданному N определить количество возможных лесенок, которые можно из них построить.

|  |  |
| --- | --- |
| 6 | 4 |
| 8 | 6 |
| 10 | 10 |
| 15 | 27 |
| 100 | 444793 |

**Подсказка.** Напишите вспомогательную рекурсивную функцию NumberOfLadders(int n, int firtsTowerHeight), которая принимает не только количество кубиков, но и высоту первой башенки. Конечным ответом будет NumberOfLadders(n, 1) + NumberOfLadders(n, 2) + … NumebrOfLadders(n, n);

**Одномерная логистика**

В одномерной стране Осьляндии все объекты находятся на одномерной оси, а их местоположение определяется единственной целочисленной координатой x. При этом несколько объектов могут иметь одну и ту же координату. Ваша задача, зная местоположение всех объектов ответить на большое количество запросов типа “Сколько объектов находятся в интервале [A, B]?”.

На вход программа получает число N - количество объектов в Осьляндии, после чего следуют N целых чисел - координаты объектов. Далее следует число M - количество запросов, после чего сами запросы. Каждый запрос состоит из двух чисел A и B (A <= B). Для каждого запроса нужно вывести единственное натуральное число - ответ на запрос, т.е. количество объектов в Осьляндии, которые находятся в данном интервале.

**Требование:** каждый запрос должен обрабатываться за время O(logN).

|  |  |
| --- | --- |
| 10  1 1 7 1 2 1 7 6 -1 0  10  0 0  0 10  -10 10  1 2  1 3  10 20  -20 -10  8 8  7 8  -1 0 | 1  9  10  5  5  0  0  0  2  2 |

[Google Диск](https://docs.google.com/)–[Сообщение о нарушении](https://docs.google.com/abuse?id=16zwHU8KZbwez-dpAO4a9jh4itj-QrbNOcdT_KtnbvHE)–Интервал автоматического обновления в минутах: 5