http://nvk26.lutsk.ua/cgi-bin/new-client?contest_id=22 (zboru2017-1, zboru2017-2, zboru2017-3, zboru2017-4, zboru2017-5,
пароль 1)

Еволюція

Під час досліджень, присвячених появі життя на планеті Олімпія, вченими було зроблено декілька сенсаційних відкриттів:

$11. Усі живі організми планети походять від однієї бактерії BitozoriaProgramulis.

$12. Еволюція проходила крок за кроком (за припущенням вчених – під час змін клімату на планеті) .

$13. На кожному кроці еволюції з кожного виду утворювалися рівно два підвиди, а попередній вид зникав.

$14. Якщо вважати появу бактерії BitozoriaProgramulis першим кроком еволюції, то нині існуючі живі організми знаходяться на
N
-
му кроці
.

Щоб не вигадувати назви під час досліджень, вчені пронумерували всі види організмів, що будь-коли існували на планеті. Для цього вони намалювали дерево еволюції із коренем BitozoriaProgramulis, яка отримала номер 1. Далі нумерували види кожного кроку еволюції зліва направо. Таким чином безпосередні підвиди

Bitozoria
Programulis
отримали номери 2 та 3. Наступними були занумеровані види третього кроку еволюції – підвиди виду 2 отримали номери 4 та 5, а виду 3 – номери 6 та 7, і т.д.

Завдання

Напишіть програму EVO, яка за номерами двох видів обчислить номер виду їх найближчого спільного предка у дереві еволюції.

Вхідні дані

Перший рядок вхідного файлу EVO.DAT містить ціле число N ($1 \le N \le 100$) – кількість етапів еволюції, що відбулися на планеті Олімпія до теперішнього часу. Другий та третій рядки файлу містять по одному натуральному числу, що представляють номери видів, для яких потрібно знайти номер їх найближчого спільного предка.

Вихідні дані

Єдиний рядок вихідного файлу EVO.SOL має містити натуральне число – номер найближчого предка для двох видів.

Приклад вхідних та вихідних даних

EVO.	DAT

EVO.	SOL

4

15

12

3

18

233016

233008

14563

Працівники (100 балів)

На заводі кожна з N деталей може бути обробленою на одному з двох верстатів: Аабо B . Кожна деталь має порядковий номер від 1 до
N
.
До обробки деталі поступають послідовно, у відповідності зі своїми номерами. Кількість деталей завжди парна.

Існують правила, за якими визначається чи можна обробляти деталь на певному верстаті.

$11) Якщо на поточний момент на верстаті Вбула оброблена така ж кількість деталей, як і на верстаті A, то наступна деталь повинна бути оброблена на верстаті
A
.

$12) У підсумку на кожному з верстатів повинно бути оброблено однакову кількість деталей.

Скільки існує людей, стільки і думок. Кожен із працівників цього заводу запропонував свою послідовність обробки деталей, причому всі пропозиції виявилися різними, але такими, що задовольняють правилам 1 і 2.

Завдання

Напишіть програму STAFF, що за інформацією про кількість деталей N визначає максимальну можливу кількість працівників заводу.

Вхідні дані

Єдиний рядок вхідного файлу STAFF.DAT містить парне число N (2≤N≤28) – кількість деталей яку необхідно обробити.

Вихідні дані

Єдиний рядок вихідного файлу STAFF.SOL має містити ціле число – максимальну можливу кількість працівників заводу.

Приклад вхідних та вихідних даних

STAFF	.DAT

STAFF	.SOL

4

2

Перший працівник вважає що на верстаті Анеобхідно обробити деталі 1 та 2, а на верстаті В, відповідно, 3 та 4. Другий має думку, що на верстаті А потрібно обробити деталі 1 та 3, а на станке В – детали 2 та 4. Інших варіантів послідовності обробки немає.

Розв’язок задачі «Працівники»

Автор: Шаміль Ягіяєв

Розв’язок 1 (Пошук з поверненням)

Найпростіший розв’язок задачі — це перебір усіх можливих варіантів обробки деталей на верстатах. У такому випадку кожного разу, коли буде оброблятися певна деталь, буде розглянуто обидва варіанти верстатів (при цьому буде перевірено, чи задовольняє це наведеним правилом). При такому підході алгоритм буде мати складність порядку $O(2^N)$.

Розв’язок 2 (Динамічне програмування)

Твердження. Задача еквівалентна задачі обчислення кількості вірних варіантів розташування n=N/2 пар круглих дужок у арифметичному виразі.

Дійсно, обробка i-ої деталі на верстаті A може означати що i-та дужка у виразу буде відкриваючою „(”, а обробка на верстаті
B
,
що закриваючою „)”. Неважко переконатися у тому що таке формулювання є адекватним початковому, проаналізувавши правила обробки деталей.

Будемо продовжувати розв’язувати задачу, використовуючи таке більш зручне формулювання.

Відомо, що коли кількість пар дужок n=1 кількість різних розташувань дужок дорівнює 1.
Нехай K(s)— це розв’язок задачі для будь-якого s
<
n
(
K
(0)=1).
Тоді число
K
(

n
)
можна отримати наступним чином.

Таким чином, кожне значення від 1 до nнеобхідно обрахувати 1 раз (проміжні результати потрібно зберігати) , на що необхідно лінійну кількість операцій. Тобто, загальна складність такого алгоритму, повертаючись до початкового формулювання складатиме
O
(
N
).

(2*n)!/(n!*(n+1)!)

(N+2)*(n+3)*

2*3*4*5

S=s+(n+2)/2+ (n+3)/3+….

Розв’язок 3 (Комбінаторні обчислення)

Задача про відшукання кількості розташування дужок у арифметичному виразі є досить поширеною. Послідовність чисел, які утворюються при розв’язанні цієї задачі при n=1,2, … називають послідовністю Каталана. Загальний член цієї послідовності можна обчислити за формулою:

Розв’язок 4 (Масив констант)

Можна помітити, що кількість різних вхідних даних дорівнює 13, тобто усі ці випадки можна перелічити у програмі, заздалегідь обрахував відповіді на них. Звісно, за час туру неможливо вручну перелічити усі випадки, і доведеться робити це з допомогою

комп’ютера.

Дільники

За заданим натуральним числом N необхідно обчислити кількість натуральних чисел, які є дільниками N! (факторіалу числа N).

Наприклад, при N=4, N!=4·3·2·1=24. Це число має такі дільники: 1, 2, 3, 4, 6, 8, 12, 24. Таким чином шукана кількість дорівнює 8.

Завдання

Напишіть програму DIVISOR, що за натуральним N, знаходить кількість дільників його факторіалу.

Вхідні дані

Єдиний рядок вхідного файлу DIVISOR.DAT містить одне ціле число N (1≤N≤45).

Вихідні дані

Єдиний рядок вихідного файлу DIVISOR.SOL має містити одне ціле число – знайдену кількість дільників числа N!

Приклад вхідних та вихідних даних

DIVISOR	.DAT

DIVISOR	.SOL

4

8