ЗМІСТ

[Теоретичні відомості 4](#_Toc115643733)

[Комбінаторика 4](#_Toc115643734)

[Рекурсія і перестановки 4](#_Toc115643735)

[Жадібний алгоритм 5](#_Toc115643736)

[**Завдання за розкладом** **5**](#_Toc115643737)

[**Коли можна бути жадібним?**  **5**](#_Toc115643738)

[**Інші жадібні алгоритми** **7**](#_Toc115643739)

[**Завдання, в яких жадібні алгоритми не дають оптимального рішення** **7**](#_Toc115643740)

[**Варіант 1** **7**](#_Toc115643741)

[**Варіант 2** **8**](#_Toc115643742)

[**Варіант 3** **8**](#_Toc115643743)

[**Варіант 4** **9**](#_Toc115643744)

[**Варіант 5** **10**](#_Toc115643745)

[**Варіант 6** **10**](#_Toc115643746)

[**Варіант 7** **11**](#_Toc115643747)

[**Варіант 8** **11**](#_Toc115643748)

[**Варіант 9** **12**](#_Toc115643749)

[**Варіант 10** **12**](#_Toc115643750)

[**Варіант 11** **13**](#_Toc115643751)

[**Варіант 12** **14**](#_Toc115643752)

[**Варіант 13** **14**](#_Toc115643753)

[**Варіант 14** **15**](#_Toc115643754)

[**Варіант 15** **15**](#_Toc115643755)

[**Варіант 16** **16**](#_Toc115643756)

[**Варіант 17** **16**](#_Toc115643757)

[**Варіант 18** **17**](#_Toc115643758)

[**Варіант 19** **18**](#_Toc115643759)

[**Варіант 20** **18**](#_Toc115643760)

[**Варіант 21** **19**](#_Toc115643761)

[**Варіант 22** **20**](#_Toc115643762)

[**Варіант 23** **22**](#_Toc115643763)

[**Варіант 24** **23**](#_Toc115643764)

[**Варіант 25** **23**](#_Toc115643765)

[**Варіант 26** **24**](#_Toc115643766)

[**Варіант 27** **25**](#_Toc115643767)

[**Варіант 28** **25**](#_Toc115643768)

[**Варіант 29** **26**](#_Toc115643769)

[**Варіант 30** **27**](#_Toc115643770)

[**Варіант 31** **27**](#_Toc115643771)

[**Варіант 32** **28**](#_Toc115643772)

[**Варіант 33** **29**](#_Toc115643773)

[**Варіант 34** **30**](#_Toc115643774)

[**Варіант 35** **31**](#_Toc115643775)

[**Варіант 36** **32**](#_Toc115643776)

[**Варіант 37** **32**](#_Toc115643777)

[**Варіант 38** **33**](#_Toc115643778)

[**Варіант 39** **34**](#_Toc115643779)

[**Варіант 40** **35**](#_Toc115643780)

[**Варіант 41** **36**](#_Toc115643781)

[**Варіант 42** **37**](#_Toc115643782)

[**Варіант 43** **37**](#_Toc115643783)

[**Варіант 44** **39**](#_Toc115643784)

[**Варіант 45** **39**](#_Toc115643785)

[**Варіант 46** **40**](#_Toc115643786)

[**Варіант 47** **41**](#_Toc115643787)

[**Варіант 48** **42**](#_Toc115643788)

[**Варіант 49** **42**](#_Toc115643789)

[**Варіант 50** **43**](#_Toc115643790)

[**Варіант 51** **43**](#_Toc115643791)

[**Варіант 52** **44**](#_Toc115643792)

[**Варіант 53** **45**](#_Toc115643793)

[**Варіант 54** **46**](#_Toc115643794)

[**Варіант 55** **47**](#_Toc115643795)

[**Варіант 56** **47**](#_Toc115643796)

[**Варіант 57** **48**](#_Toc115643797)

[**Варіант 58** **49**](#_Toc115643798)

[**Варіант 59** **50**](#_Toc115643799)

[**Варіант 60** **50**](#_Toc115643800)

[**Варіант 61** **51**](#_Toc115643801)

[**Варіант 62** **51**](#_Toc115643802)

[**Варіант 63** **52**](#_Toc115643803)

[**Варіант 64** **53**](#_Toc115643804)

[**Варіант 65** **53**](#_Toc115643805)

[**Варіант 66** **54**](#_Toc115643806)

[**Варіант 67** **55**](#_Toc115643807)

[**Варіант 68** **55**](#_Toc115643808)

[**Варіант 69** **56**](#_Toc115643809)

[**Варіант 70** **58**](#_Toc115643810)

[**Варіант 71** **58**](#_Toc115643811)

[**Варіант 72** **59**](#_Toc115643812)

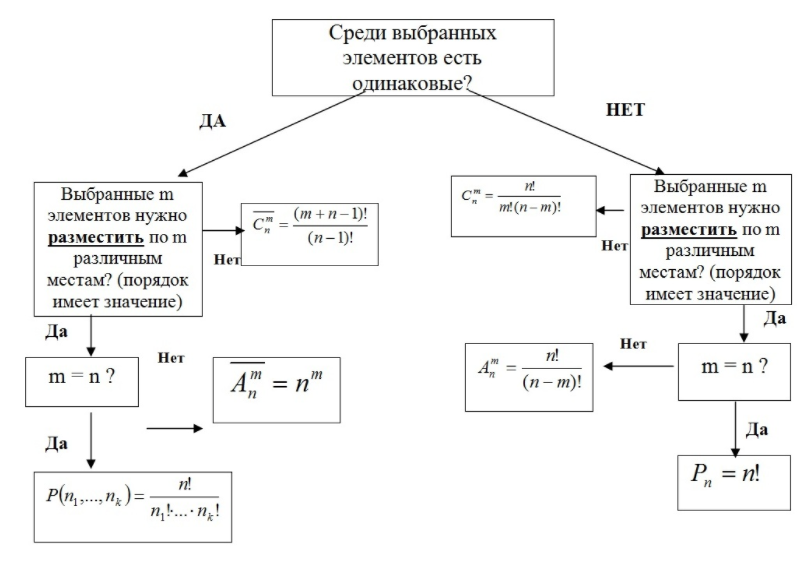
[**Варіант 73** **60**](#_Toc115643813)

[**Варіант 74** **61**](#_Toc115643814)

[**Варіант 75** **62**](#_Toc115643815)

# Теоретичні відомості

## Kombinatorika

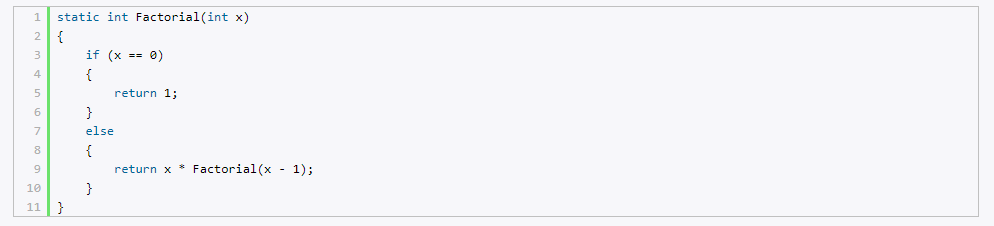


## Рекурсія і перестановки

Рекурсивна функція - це конструкція, в якій функція називає себе.

Взяти, наприклад, факторний розрахунок, в якому використовується формула n! = 1 \* 2 \* ... \* n. Наприклад, факторіал числа 5 дорівнює 120 = 1 \* 2 \* 3 \* 4 \* 5.

Визначимося з методом знаходження факторіала:



## Жадібний алгоритм

Жадібний алгоритм - це алгоритм, який на кожному кроці робить кращий локальний вибір в надії, що остаточне рішення буде оптимальним.

Наприклад, [алгоритм Дейкстри](http://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D1%8B) з знаходження найкоротшого шляху в графі досить жадібний, адже на кожному кроці ми шукаємо вершину з найменшою вагою, на якій ми ще не були, а потім оновлюємо значення інших вершин. В той же час ми можемо довести, що найкоротші шляхи, знайдені в вершинах, є оптимальними.

До речі, [алгоритм Флойда](http://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A4%D0%BB%D0%BE%D0%B9%D0%B4%D0%B0_%E2%80%94_%D0%A3%D0%BE%D1%80%D1%88%D0%B5%D0%BB%D0%BB%D0%B0), який також шукає найкоротші шляхи в графі (правда, між усіма вершинами), не є прикладом жадібного алгоритму. Флойд демонструє ще один метод - метод динамічного програмування.

Використання жадібного алгоритму досить стандартно. Розглянемо його на прикладі наступного завдання:

### **Планування завдання**

Нехай програмісту-фрілансеру Васі Пупкіну дадуть n завдань. Кожне завдання має свій термін виконання, а також свою вартість (тобто якщо він не виконує це завдання, то втрачає стільки грошей). Вася настільки крутий, що може виконати одне завдання за один день. Почати роботу можна з 0. Потрібно максимізувати прибуток.

Класичний приклад використання жадібності: Васі вигідно робити самі «дорогі завдання», а найменш дорогі виконувати не можна - тоді прибуток буде максимальною. Виникає питання: як розподілити завдання? Ми будемо перебирати завдання в порядку убування вартості і заповнювати графік наступним чином: якщо в графіку до його дедлайну знайдеться ще хоча б одне вільне місце для замовлення, то ми поставимо його на останнє з цих місць, інакше не зможемо виконати його вчасно, то поставимо в кінці вакантних місць.

### **Коли можна бути жадібним?**

~~Іноді~~ може бути спокусливо використовувати жадібність всюди, де тільки можна, але на деяких завданнях це неприпустимо. Наприклад, [проблема рюкзака](http://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D1%80%D0%B0%D0%BD%D1%86%D0%B5): злодій пробрався на склад, на якому зберігаються три речі вагою 10 кг, 20 кг і 30 кг і вартістю 60, 100 і 120 дерев'яних вічнозелених нанорублів відповідно. Злодій може нести максимум 50 кг. Потрібно максимізувати прибуток злодія. Якщо діяти тут жадібно і вибрати найцінніше (тобто, 6 нанорублів на кг першого шматка, 5 нанорублів на кг другого і 4 наноррубля на кг третього), то злодій повинен взяти перше, тоді залишиться місце для другого, але оптимальне рішення - друге і третє.

Висновок: існує область застосовності жадібних алгоритмів. Загальних рецептів тут немає, але є досить потужний засіб, за допомогою якого в більшості випадків можна визначити, чи дасть жадібність оптимальне рішення. Це називається матроїд.

***В рамках практичного доведення того, що жадібний алгоритм не є оптимальним, наведена нижче інформація приводиться в ознайомлювальних цілях.***

Як припускає Вікіпедія, [матроїд](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%BE%D0%B8%D0%B4) - це пара (X, I), де X - скінченна множина, яка називається **носієм матроїда**, а I - деяка множина підмножин X, **звана сімейством незалежних множин**. Необхідно виконати наступні умови:

1.   
   Набір I не порожній. Навіть якщо вихідна множина X була порожньою— X = Ø, то i буде складатися з одного елемента—множини, що містить порожній. I = {{Ø}}
2.   
   Будь-яка підмножина будь-якого елемента множини I також буде елементом цієї множини.
3.   
   Якщо множини A і B належать до множини I, а також відомо, що розмір A менше B, то є якийсь елемент x з B, який не належить A, такий, що об'єднання x і A буде належати множині I. Це властивість не зовсім тривіальне, але найчастіше найважливіше з усіх інших.

Матроїд називається зваженим, якщо на множині X існує адитивна вагова функція w. Вага множини буде визначатися як сума ваг її елементів.

Повернемося до програміста-фрілансера із завданнями. Тут можна побачити наступний матроїд: нехай носієм буде набір завдань, а самостійні набори - успішно виконані завдання. Нехай вага кожної програми буде його величиною. Давайте перевіримо, чи є ця пара матроїдом:

1. перше майно, очевидно, виконується. У наш набір входить порожній набір виконаних завдань.Те, що Вася не хоче заробляти, не має значення.
2. також виконується другий набір. Чому це так: Давайте відсортуємо успішно виконані завдання в порядку збільшення терміну. У такому порядку вони все одно будуть успішно завершені. У такому порядку очевидно, що будь-яка підмножина успішно виконаних завдань буде успішно виконана.
3. третя властивість хоч і не очевидна, але виконана. Припустимо, у нас є два набори успішно виконаних завдань А і В, і ми знаємо, що | А| < | Б|. Стандартно давайте розберемо завдання в порядку збільшення дедлайну в обох сетах. Візьмемо завдання від Б, якого немає в А, і спробуємо додати його в набір А. У нас все вийде, адже якщо в А не було місця, то це завдання повинна була бути присутня.

Навіщо я це роблю? Принадність матроїдів полягає в теоремі [Радо-Едмондса](http://ru.wikipedia.org/wiki/%D0%96%D0%B0%D0%B4%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A0%D0%B0%D0%B4%D0%BE_%E2%80%94_%D0%AD%D0%B4%D0%BC%D0%BE%D0%BD%D0%B4%D1%81%D0%B0): якщо довести, що об'єкт є матроїдом, то жадібний алгоритм буде працювати коректно і давати правильний результат.

### **Інші жадібні алгоритми**

* [Алгоритм Хаффмана](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0) (адаптивний алгоритм оптимального префіксального кодування алфавіту з мінімальною надмірністю).
* [Алгоритм Крускаля](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9A%D1%80%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D0%B0) (пошук дерева скелета мінімальної ваги в графі).
* [Алгоритм Прими](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9F%D1%80%D0%B8%D0%BC%D0%B0) (знаходження охоплює дерева мінімальної ваги в когерентному графі).

Узагальненням жадібних алгоритмів є [алгоритм Радо-Едмондса](https://ru.wikipedia.org/wiki/%D0%96%D0%B0%D0%B4%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A0%D0%B0%D0%B4%D0%BE_%E2%80%94_%D0%AD%D0%B4%D0%BC%D0%BE%D0%BD%D0%B4%D1%81%D0%B0).

### **Завдання, в яких жадібні алгоритми не дають оптимального рішення**

Для ряду завдань в [класі NP](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_NP) жадібні алгоритми не забезпечують оптимального рішення. До них відносяться:

* [завдання комівояжера](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BC%D0%B8%D0%B2%D0%BE%D1%8F%D0%B6%D0%B5%D1%80%D0%B0);
* [задача мінімального забарвлення графа](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BA%D1%80%D0%B0%D1%81%D0%BA%D0%B0_%D0%B3%D1%80%D0%B0%D1%84%D0%B0);
* [проблема поділу графа на підграфи](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%B1%D0%B8%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B3%D1%80%D0%B0%D1%84%D0%B0);
* [завдання вибору максимального кліка](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D0%BA%D0%BB%D0%B8%D0%BA%D0%B5);
* [Планування завдань](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5)

### **Варіант 1**

Для того щоб заробити величезний капітал, новим росіянам необхідно володіти неординарним складом розуму. Звичайно, при такій складній роботі також повинні бути якісь особливі механізми відпочинку і розваг. Для цих цілей в казино був придуманий спеціальний набір доміно для нових росіян. Звичайні кістки доміно являють собою набір різних комбінацій комбінацій з двох плиток, кожна з яких відображає від 0 до 6 точок. І цей набір являє собою аналогічну комбінацію плиток, але кількість очок на кожній може бути від нуля до заданого значення, яке залежить від інтелектуального рівня гравців. У такому наборі кісток є всілякі поєднання плиток, але жодна з кісток не повторюється (навіть такі поєднання, як 2-5 і 5-2, вважаються однаковими).

Щоб зробити цей набір кісток, виробник зіткнувся з проблемою підрахунку загальної кількості балів на всіх кістках доміно. Це пов'язано з тим, що доміно для нових росіян прикрашають діамантами, які є точками на кахлі і при виготовленні необхідно оцінити вартість.

Допоможіть написати програму, яка вирішить цю проблему.

Вводу

Вхідний файл INPUT.TXT містить одне натуральне число N – максимальну кількість точок на одній плитці доміно. (N ≤ 10000)

Вихід

У вихідному файлі OUTPUT.TXT перерахуйте кількість алмазних каменів, які потрібно виготовити для заданого набору кісток.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 2 | 12 |

### **Варіант 2**

Одного разу, коли вона прийшла додому зі школи, Свєта знайшла записку від мами, в якій попросила приготувати салат. Свєта знала, що салат - це суміш двох і більше інгредієнтів, тому їй не склало труднощів виконати прохання мами.

Але Свєта хоче стати математиком, тому для навчання вирішила порахувати, скільки різних салатів вона може приготувати з наявних продуктів (майонез, огірки, помідори). Після невеликого розрахунку вона отримала відповідь: 4.

Знаючи, що ви любите цікаві головоломки і хочете стати програмістами, Свєта попросила вас написати програму, яка визначає кількість різних салатів на довільну кількість інгредієнтів.

Вводу

Вхідний файл INPUT.TXT містить натуральне число N – кількість доступних інгредієнтів (N < 32).

Вихід

У вихідному файлі OUTPUT.TXT перерахуйте кількість різних салатів.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3 | 4 |
| 2 | 4 | 11 |

### **Варіант 3**

Знайдіть кількість невироджених прямокутників зі сторонами, паралельними осям координат, вершини яких лежать в точках з цілочисельними координатами в межах або на кордоні прямокутника, протилежні кути якого знаходяться в точках (0, 0) і (W, H).

Вводу

Вхідний файл INPUT.TXT містить два натуральних числа W і H, які не перевищують 1000.

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 1 | 1 |
| 2 | 2 1 | 3 |
| 3 | 2 2 | 9 |

### **Варіант 4**

У знаменитому місті Кизилорда, де є N центрів, живе якийсь граф - Азамат. Він хотів знати кількість різних дорожніх споруд між ними, якби було відомо, що два центри можуть бути з'єднані в одному з двох напрямків або взагалі не з'єднані. Наприклад, з N=2 все виявляється 3 варіантами:

два центри не пов'язані між собою

дорога йде від першого до другого центру

дорога йде від другого до першого центру

Вводу

У вхідний файл INPUT.TXT записується єдине натуральне число - кількість центрів в місті, 2 ≤ N ≤ 100.

Вихід

В одному рядку вихідного файлу OUTPUT.TXT потрібно відобразити кількість різних дорожніх будівель.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 2 | 3 |
| 2 | 4 | 729 |

### **Варіант 5**

Візерунок розміру n - це рядок довжини n, кожен з символів якого входить в набір {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, ?}. Шаблони перетворюються в рядки чисел за такими правилами:

• символи від 0 до 9 можуть бути перетворені тільки в себе;• символ a може бути перетворений в будь-який з символів 0,1, 2, 3;• символ b може бути перетворений в будь-який з символів 1,2,3,4;• символ c може бути перетворений в будь-який з символів 2,3,4,5;• символ d може бути перетворений в будь-який з символів 3,4,5,6;• символ e може бути перетворений в будь-який з символів 4,5,6,7;• символ f може бути перетворений в будь-який з символів 5,6,7,8;• символ g може бути перетворений в будь-який з символів символи 6,7,8,9;• символ ? можна конвертувати в будь-який з символів від 0 до 9;

Наводяться два шаблони: p1 і p2. Розглянемо набір рядків S1, які можна отримати з p1 за описаними правилами, і набір рядків S2, які можна отримати з p2. Потрібно знайти кількість рядків, з яких складаються обидва цих набору.

Вводу

Перший рядок файлу INPUT.TXT містить шаблон p1, а другий рядок - шаблон p2. Шаблони мають однаковий позитивний розмір, не більше 9.

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | ??? абетка | 64 |
| 2 | ??? 000 | 1 |
| 3 | abc999 | 0 |

### **Варіант 6**

Вказується натуральне число N. Потрібно написати програму, яка знаходить кількість натуральних чисел, які не перевищують N і не розходяться ні на одне з чисел 2, 3, 5.

Вводу

Вхідний файл INPUT.TXT містить номер N (1 ≤ N ≤ 109).

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 10 | 2 |
| 2 | 20 | 6 |

### **Варіант 7**

На день народження Петі подарували набір листівок з буквами. Тепер Петро з великим інтересом робить з них різні слова. І ось, одного разу, зробивши ще одне слово, Петро зацікавився питанням: «Скільки різних слів можна зробити з тих же карт, що і ця?».

Допоможіть йому відповісти на це питання.

Вводу

У вхідному файлі INPUT.TXT вказується слово, складене Petya - рядок з дрібних англійських букв довжиною не більше 15 символів.

Вихід

У вихідному файлі OUTPUT.TXT виводимо одне ціле число - відповідь на поставлене завдання.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | тільки | 12 |

### **Варіант 8**

Петро грає в цікаву гру. Для цієї гри потрібна монета. Пітер кидає його n разів і підраховує, скільки разів падають «хвости». Якщо хвости падають хоча б м рази, то Петро вважає, що виграв гру.

Одного разу Пітер поцікавився, яка ймовірність того, що він виграє гру. Для цього він хоче знайти кількість послідовностей результатів підкидання монет, що містять рівно n кидків, при яких «хвости» падали не менше м разів.

Допоможіть Петі - знайдіть це число, вважаючи, що з кожним кидком монета може впасти або «орел», або «хвіст».

Вводу

Вхідний файл INPUT.TXT містить два цілих числа: n і m (1 ≤ n ≤ 20,0 ≤ m ≤ n).

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 2 0 | 4 |
| 2 | 3 2 | 4 |

### **Варіант 9**

Всім відомо, що щасливі числа - це ті числа, які містять тільки щасливі числа 7 і 4. Потрібно знайти число щасливих чисел не велике N.

Вводу

Один рядок у вхідному файлі INPUT.TXT містить натуральне число N, що не перевищує 1032.

Вихід

В один рядок вихідного файлу.TXT потрібно вивести одне ціле число - відповідь на поставлене завдання.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 56 | 4 |

### **Варіант 10**

У міському зоопарку містяться тварини n різних видів. Для участі в міжнародній виставці «Три істоти» зоопарк повинен представити трьох тварин різних видів.

Потрібно написати програму, яка розрахує кількість способів відбору трьох тварин для участі у виставці.

Наприклад, якщо в зоопарку два ведмеді, тигр, лев і пінгвін, то є сім способів вибрати трьох тварин:

перший ведмідь, тигр і лев;

перший ведмідь, тигр і пінгвін;

перший ведмідь, лев і пінгвін;

другий ведмідь, тигр і лев;

другий ведмідь, тигр і пінгвін;

другий ведмідь, лев і пінгвін;

тигр, лев і пінгвін.

Вводу

Вхідний текстовий файл INPUT.TXT містить в першому рядку натуральне число n – кількість видів тварин в міському зоопарку (1 ≤ n ≤ 1000). У другому рядку через пробіл записуються n натуральних чисел - кількість тварин відповідного виду. Кількість тварин кожного виду не перевищує 1000.

Вихід

Вихідний текстовий файл OUTPUT.TXT повинен містити один номер – кількість способів відбору трьох тварин для міжнародної виставки.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 42 1 1 1 | 7 |
| 2 | 3100 100 100 | 1000000 |

### **Варіант 11**

Нам потрібно знайти ряд способів розмістити граків N×N K на шахівниці, щоб вони не били один одного. Всі граки вважаються однаковими.

Вводу

У вхідному файлі INPUT.TXT записуються натуральні числа N і K (N, K ≤ 8).

Вихід

У вихідному файлі OUTPUT.TXT виведіть одне ціле число, відповідь на задачу.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 8 8 | 40320 |

### **Варіант 12**

Перестановка N елементів - це впорядкований набір n різних чисел від 1 до N. Кількість всіх перестановок порядку N дорівнює PN = N!

Необхідно знайти перестановку за її номером в лексикографічному порядку (за алфавітом). Наприклад, для N=3 лексикографічний порядок перестановок виглядає наступним чином:

(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1).

Таким чином, перестановка (2,3,1) має в даній послідовності число 4.

Вводу

У першому рядку вхідного файлу INPUT.TXT записується число N (1 ≤ N ≤ 12) - кількість елементів в перестановці, у другому - число K (1 ≤ K ≤ N!) - число перестановки.

Вихід

У вихідному файлі .TXT вивести N чисел - потрібну перестановку - через пробіл.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 11 | 1 |
| 2 | 32 | 1 3 2 |

### **Варіант 13**

Визначимо відстань між лініями однакової довжини SA і SB (позначають d(SA, SB)) як суму для всіх 1 ≤ i ≤ |SA| найкоротші відстані між буквами SA(i) і SB(i) в циклічно замкнутому англійському алфавіті (тобто після букви «a» стоїть буква «b», ..., після букви «z» - «a»). Наприклад, d(aba, aca) = 1, а d(aba, zbz) = 2.

Напомним, что циклическим сдвигом строки S называется строка (обозначим, как S→k) Sk+1Sk+2Sk+3 ... С| С| S1S2 ... Sk для некоторого k, где | С| – длина строки S.

Ступінь циклічної відстані між струнами SA і SB (|SA| = | СБ|) сума називається:

Текст

Опис генерується автоматично

Потрібно обчислити ступінь циклічної відстані вказаних STRINGS SA і SB.

Вводу

Вхідний файл INPUT.TXT містить два рядки однакової довжини, не більше 105 символів. Рядки складаються тільки з маленьких букв англійського алфавіту.

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | Викл. | 1 |
| 2 | Дкс | 8 |

### **Варіант 14**

Нехай x — натуральне число. Назвемо його дільником, якщо 1 ≤ ≤ x, а залишок від ділення x на y дорівнює нулю.

Вказується число x. Знайдіть кількість його дільників.

Вводу

Вхідний файл INPUT.TXT містить вказане число x (1 ≤ x ≤ 1018). Всі прості дільники числа x не перевищують 1000.

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 12 | 6 |
| 2 | 239 | 2 |

### **Варіант 15**

Нехай x — натуральне число. Назвемо його дільником, якщо 1 ≤ ≤ x, а залишок від ділення x на y дорівнює нулю.

Вказується число x. Знайдіть число його дільників, що діляться на кожне з простих чисел, на які x ділиться.

Вводу

Вхідний файл INPUT.TXT містить ціле число x (1 ≤ x ≤ 1018). Всі прості дільники числа x не перевищують тисячі.

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 12 | 2 |
| 2 | 239 | 1 |

### **Варіант 16**

Розглянемо фігуру, схожу на зображену на малюнку (великий рівносторонній трикутник, складений з малих рівносторонніх трикутників). На малюнку зображена фігура, що складається з 4 рівнів трикутників.

Потрібно написати програму, яка буде визначати, скільки в ній трикутників (необхідно враховувати не тільки «малі» трикутники, але в цілому всі трикутники - зокрема, трикутник, виділений жирним шрифтом, а також вся фігура, є цікавлять нас трикутниками).

Вводу

Вхідний файл INPUT.TXT містить один номер N , кількість рівнів на малюнку (1 ≤ N ≤ 105).

Вихід

Вихідний файл на виході.TXT повинен містити одне число - кількість трикутників в такій формі.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 | 1 |
| 2 | 2 | 5 |
| 3 | 4 | 27 |

### **Варіант 17**

У вас є N вишикуваних коробок, червоні та B сині кульки. Всі червоні кульки (схожі на сині кульки) ідентичні. Можна покласти кульки в ящики. Допускається розміщення в ящиках одночасно кульок як одного, так і двох видів. Також допускається залишити частину ящиків порожніми. Не обов'язково складати всі кульки в коробки.

Ви хочете написати програму, яка визначає кількість різних способів, за допомогою яких можна заповнити ящики кульками.

Вводу

Входной файл INPUT.TXT содержит целые числа N, A, B. (1 ≤ N ≤ 20, 0 ≤ A, B ≤ 20)

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 1 1 | 4 |
| 2 | 2 1 1 | 9 |

### **Варіант 18**

Петро і Вася захопилися азартними іграми на гроші. Гра складається з безлічі раундів, в кожному з яких кидається монета і якщо «Хвости» падають, то Пітер виграє і отримує очко, в іншому випадку очко присуджується Васі. Бали, набрані в різних раундах, підсумовуються і перемагає перший, хто набрав N очок.

Несподівано в середині гри пролунав дзвінок, і хлопцям довелося закінчити гру. Допоможіть Петі і Васі справедливо розділити монети на лінії між собою відповідно до математичним очікуванням виграшу.

Вводу

Вхідний файл INPUT.TXT містить 4 цілих числа N, K1, K2 і S в одному рядку. N – кількість очок, які необхідно набрати для перемоги. К1 і К2 - поточні точки Петі (К1) і Вася (К2). S – кількість монет, про які йде мова і які потрібно розділити між гравцями.

Ограничения: 1 ≤ N ≤ 50.0 ≤ K1, K2 < N, 1 < S < 10100.

Вихід

У вихідному файлі.TXT, в пробілі відображають два цілих числа: кількість монет, які повинні отримати Петя і Вася. Не забувайте, що сума цих чисел повинна точно дорівнювати S. Гарантовано, що однозначна відповідь в цілих числах існує.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3 1 2 16 | 4 12 |
| 2 | 4 1 1 2 | 1 1 |
| 3 | 5 1 4 32 | 2 30 |

### **Варіант 19**

Пітер виписав всі комбінації з N перших англійських букв на букви K. У кожному поєднанні він виписував букви в лексикографічному порядку. Він виписував поєднання в лексикографічному порядку, по одному на рядок. Тепер він хоче знати, яке слово написано в М-му рядку.

Вводу

Вхідний файл INPUT.TXT містить цілі числа N, K, M (1 ≤ N ≤ 26, 1 ≤ K ≤ N). Гарантовано, що М не перевищує кількість всіх призначених комбінацій.

Вихід

У вихідному файлі OUTPUT.TXT виведіть комбінацію M-ї виведеної системи.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 4 2 3 | до |

Пояснення

Всі комбінації знаходяться в тому порядку, в якому вони пишуться: ab, ac, ad, bc, bd, cd. Тут комбінація 3 м - це реклама.

### **Варіант 20**

Гру у волейбол виграє та команда, яка першою набере 25 очок з перевагою не менше двох очок. У разі рівного рахунку 24-24 гра триває до тих пір, поки не буде досягнута перевага в 2 очки (26-24; 27-25).

Дві зіграні гри, які закінчилися з однаковим рахунком, будуть вважатися різними, якщо лінії, в яких виписаний порядок підрахунку очок командами, не будуть рівними.

Комітет змагань з волейболу зацікавився кількістю різних ігор, що завершилися з рахунком 25:23. Їх було 16123801841550.

Визначте, скільки існує різних партій, які закінчуються в заданому балі.

Вводу

Вхідний файл INPUT.TXT вказує підсумковий рахунок в партії (тобто такий, в якому перемогу в грі дає одна з команд). Також відомо, що жодна з команд не набрала більше 40 очок.

Вихід

У вихідному файлі OUTPUT.TXT, перерахуйте кількість різних пакетів, які закінчуються цим обліковим записом.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 25:12 | 1251677700 |
| 2 | 20:25 | 1761039350070 |
| 3 | 25:23 | 16123801841550 |

### **Варіант 21**

Перестановка N елементів - це впорядкована множина з N різних чисел від 1 до N.

Розміщення порядку K - це підмножина елементів деякої перестановки порядку N. Наприклад, (1, 3) - це розміщення порядку 2 для перестановки (1, 2, 3) порядку 3.

Потрібно визначити його положення в лексикографічному порядку всіх можливих розміщень, утворених з усіх можливих перестановок порядку N із заданого місця.

Наприклад, лексикографічна послідовність всіх можливих розміщень для K=2 і N=3 виглядає наступним чином:

(1,2), (1,3), (2,1), (2,3), (3,1), (3,2)

Таким чином, зміщення (2,3) в даній послідовності нумерується 4.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить цифри N і K (1 ≤ K ≤ N ≤ 12). Другий рядок містить K чисел з діапазону від 1 до N - розміщення.

Вихід

У вихідному файлі OUTPUT.TXT виведіть сингулярний номер цього місця.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3 23 2 | 6 |
| 2 | 6 41 3 2 5 | 14 |

### **Варіант 22**

Ваня вирішив зайнятися музикою. А ось який музичний інструмент вибрати для вивчення, він поки не визначився. Його друг Женя, будучи трохи старшим і досвідченішим, вже добре освоїв гітару. Він хоче показати Вані можливості гри на гітарі.

Гітара має шість струн, кожну з них можна затиснути пальцем на певному ладі або залишити відкритою (не натискати ні на який лад), це змінює ноту, яка звучить при коливанні струни.

Щоб продемонструвати можливості інструменту, Женя веде таку розмову з Ванею. Ваня називає певний акорд, а Женя розповідає йому, скільки способів можна взяти цей акорд на гітарі, тобто вибрати на кожній струні лад і затиснути його (або залишити відкритим), щоб цей акорд звучав. Євгенія любить гучний звук, тому повинні звучати всі струни.

Звичайно, в музиці є сім нот, але крім них є ще й напівтони. Тому для зручності будемо вважати, що купюр всього 12. Їх позначення такі: A, Bb, B, C, C#, D, D#, E, F, F#, G, G#.

Крім того, C# можна позначити як Db, Bb як A#, D# як Eb, F# як Gb і G# як Ab. Позначення приміток дається в тому порядку, в якому вони слідують, тобто за А слід Bb, за ним слідує B , . . . , G# знову слід A. Відстань між будь-якими двома сусідніми нотами становить один півтон.

Гітара має шість струн, вони пронумеровані від 1 до 6. Кожна струна видає певну ноту при її звучанні і ніяк не притискається. Набір нот, відповідних відкритим струнам, називається гітарним тюнінгом. Якщо відкритий рядок, що випускає номер ноти i, затиснутий на j-tom fret (лади нумеруються від одиниці), то цей рядок буде видавати ноту i + j, тобто з ноти i необхідно перейти до j нот в циклі. Наприклад, якщо відкритий рядок видає ноту D, то утримуючи її на третьому ладі, отримуємо ноту F, а на восьмій - ноту Bb.

На гітарі Вані є N ладів, тому 1 ≤ j ≤ N. Існує велика кількість акордів, але вони побудовані за загальною схемою. Спочатку вибирається так званий акордний тонік - нота, з якої він буде будуватися. Потім щодо нього виписуються інші ноти акорду.

Запис акорду складається з двох частин. Перша частина - це завжди позначення тоніка акорда. У другій частині описується саме звучання акорду. Обмежимося кількома особливими випадками.

Мажорний акорд

До тоніку додаються ноти, розділені 4 і 7 півтонами. Друга частина запису цього акорду порожня.Приклад: Eb, складається з нот Eb, G, Bb.

Мінорний акорд

До тоніку додаються ноти, розділені 3 і 7 півтонами.Друга частина запису цього акорду складається з невеликої англійської літери m.Приклад: Am, складається з нот A, C, E.

Великі перегородки

Він утворюється з мажорного акорду шляхом додавання ноти на відстані 10 півтонів від тоніка.Друга частина складається з числа 7.Приклад: Е7, складається з нот Е, В, Г#, Д.

Мала перегородка

Він утворюється з мінорного акорду шляхом додавання ноти на відстані 10 півтонів від тоніки.Друга частина складається з букви m і числа 7.Приклад: Gm7, складається з нот G, D, Bb, F.

Напишіть програму, яка допоможе Джин відповісти на питання Вані. Зауважимо, що акорд вважається взятим на гітару, якщо не опущена нота акорду і не взята нота, що не належить акорду. На відміну від справжньої гри на гітарі, відстань між ладами, на яких затиснута хоча б одна струна, може бути будь-яким.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить N , кількість ладів гітари (0 ≤ N ≤ 9). У другому рядку записується гітарний тюнінг - шість нот, відповідних шести струнам гітари. Нотатки позначаються як у тексті завдання та відокремлюються одна від одної пробілами. Важливий випадок з листами. У третьому рядку записується акорд, який необхідно взяти на гітару.

Вихід

У вихідному файлі.TXT, перерахуйте кількість способів взяти заданий акорд на заданій гітарі.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3Е Б Г Д А ЕК | 4 |
| 2 | 0А С Е А С ЕАм | 1 |

Пояснення прикладів

Якщо позначимо варіант взяття акорду з шістьма цифрами - ладами, на яких затискаються струни з першого по шостий, розглядаючи відкриту струну, затиснуту на нульовому ладі, то в першому прикладі чотири способи такі:

(0; 1; 0; 2; 3; 0) (3; 1; 0; 2; 3; 0) (0; 1; 0; 2; 3; 3) (3; 1; 0; 2; 3; 3)

У другому прикладі можна грати тільки на відкритих струнах, але гітара налаштована точно на необхідний акорд.

### **Варіант 23**

В результаті чергового геніального поєднання Остап Бендер і його супутники - К діти лейтенанта Шмідта виявилися Х рублями п'ятирублевими банкнотами. І ось, як завжди, справа дійшла до поділу ...

Шура Балаганов запропонував «по справедливості», тобто всім в рівній мірі. Паніковський вирішив віддати собі половину, а решту «по заслугах». Кожен з дітей лейтенанта К пропонував щось цікаве. Однак у Великого комбінатора була своя думка з цього приводу...

Ваше завдання - знайти ряд способів розділити наявні гроші між усіма учасниками цих славних подій: К дітьми лейтенанта Шмідта і Остапа Бендера.

Вводу

Вхідний файл INPUT.TXT містить цілі числа X (0 ≤ X ≤ 500) і K (0 ≤ K ≤ 100). Природно, число Х ділиться на 5. А при розподілі не допускається розривання п'ятирублевих банкнот.

Вихід

У вихідному файлі OUTPUT.TXT виведіть одне ціле число, кількість способів ділення.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 15 2 | 10 |

### **Варіант 24**

Назвемо число гладким, якщо його цифри, починаючи з найвищої цифри, утворюють не зменшувану послідовність. Давайте впорядкуємо всі такі числа в порядку зростання і присвоїмо кожному число.

Потрібно вивести N-е гладке число на число N.

Вводу

Вхідний файл INPUT.TXT містить цифру N (1 ≤ N ≤ 2147483647).

Вихід

Вихідний файл OUTPUT.TXT повинен містити потрібне N-е гладке число.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 | 1 |
| 2 | 11 | 12 |
| 3 | 239 | 1135 |

### **Варіант 25**

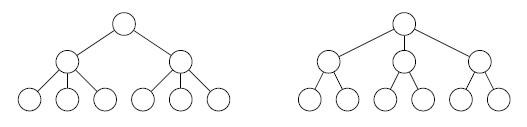
2-3 дерево - це елегантна структура даних, винайдена Джоном Хопкрофтом. Він призначений для використання з тією ж метою, що і бінарне дерево пошуку. 2-3 дерево - це дерево з коренем, яке має такі властивості:

корінь і кожна внутрішня вершина має або 2, або 3 дітей;

глибина всіх листя однакова.

Єдиний виняток - коли дерево містить рівно одну вершину. При цьому корінь дерева є одночасно і листом і тому не має діток. Основна суть цих властивостей полягає в тому, що дерево з L листям має висоту O(log L).

Якщо говорити в загальному, то може бути кілька 2-3 дерев з L листям. Наприклад, на наступній ілюстрації показано два можливих дерева з 6 листами.



Для заданого числа L знайдіть кількість різних 2-3 дерев з L листям. Оскільки відповідь може бути досить великим, виведіть його за модулем R.

Вводу

Вхідний файл INPUT.TXT містить два цілих числа: L і R (1 ≤ L ≤ 5 000, 1 ≤ R ≤ 109).

Вихід

У вихідному файлі.TXT виводять одне число - кількість різних 2-3 дерев, які мають рівно L листя, взяті за модулем R.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 6 1000000000 | 2 |
| 2 | 7 1000000000 | 3 |

### **Варіант 26**

Василь Афанасьєв в якості курсової роботи отримав завдання - побудувати комп'ютер, який би працював не з цифрами, а з рядками. Вася вперше закріпив якусь абетку. Давайте позначимо К кількість букв в цьому алфавіті. Далі Вася записав набір різних струн, довжиною не більше N кожної, які назвав базовими. Комп'ютер може працювати тільки з рядками, які отримані шляхом об'єднання (т. Е. Атрибуції) деяких рядків з цього набору один одному (один і той же рядок може використовуватися кілька разів при приписуванні). Однак виявилося, що початковий базовий набір був надмірним! Це означає, що в ньому була рядок, при видаленні з набору набір не змінює безліч рядків, з якими може працювати комп'ютер.

Васю цікавило питання про те, скільки рядків може бути в невимірної множині, і скільки таких максимальних множин існує.

Так як його комп'ютер ще не готовий, він попросив вас порахувати це число.

Вводу

Вхідний файл INPUT.TXT містить два цілих числа по 1 ≤ N ≤ 1 000 і 1 ≤ K ≤ 100 – відповідно, максимальна довжина рядка і кількість символів в алфавіті.

Вихід

У першому рядку вихідного файлу.TXT відображають кількість рядків в максимальному наборі. Другий - кількість таких наборів.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 1 | 11 |
| 2 | 3 2 | 121 |

### **Варіант 27**

Для заданих натуральних чисел N і K потрібно обчислити кількість чисел від 1 до N, які мають в двійковій нотації рівно K нулі.

Наприклад, якщо N=8 і K=1, то в двійковій системі числення ми можемо записати всі числа від 1 до 8:

1, 10, 11, 100, 101, 110, 111 і 1000.

Звідки видно, що тільки цифри 10, 101 і 110 мають в записі рівно один нуль, тобто правильна відповідь - 3.

Вводу

Один рядок у файлі INPUT.TXT містить два натуральних числа з пробілом N і K, що не перевищує 109.

Вихід

В один рядок вихідного файлу.TXT потрібно вивести одне ціле число - кількість чисел від 1 до N з K нулями в двійковому поданні.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 8 1 | 3 |
| 2 | 13 2 | 4 |
| 3 | 1000 5 | 210 |

### **Варіант 28**

Іван Іванович запросив на свій день народження багатьох гостей. Він писав на картках імена всіх гостей і розкладав ці картки на столі, вважаючи, що кожен гість буде сидіти там, де знайде листівку зі своїм прізвищем (прізвища у всіх гостей різні). Однак гості не звернули уваги на карти і сіли за стіл без особливого порядку. При цьому Іван Іванович з подивом виявив, що в призначеному для нього місці не сів жоден гість.

Потрібно написати програму, яка знайде, скільки способів можна розсадити гостей, щоб ніхто з них не сидів там, де була карта з його ім'ям.

Вводу

У вхідному файлі INPUT.TXT ціле число N встановлюється на кількість гостей (1 ≤ N ≤ 100).

Вихід

Вихідний файл OUTPUT.TXT повинен містити одне ціле число, яке є кількістю способів розсадження гостей.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 | 0 |
| 2 | 2 | 1 |
| 3 | 5 | 44 |
| 4 | 20 | 895014631192902121 |

### **Варіант 29**

Дано послідовність натуральних чисел 7, 11, 13, 14, 19, 21, 22, 25, ....

Потрібно написати програму, яка знайде N-й доданок цієї послідовності з заданого N.

Вводу

Вхідний файл INPUT.TXT містить цифру N (1 ≤ N ≤ 2147483647).

Вихід

У вихідному файлі OUTPUT.TXT виведіть N-й член послідовності.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 | 7 |
| 2 | 6 | 21 |

### **Варіант 30**

У Стародавньому Китаї було придумано багато цікавих ігор. Всім відома дуже відома гра «Маджонг». У цьому завданні ми розглянемо ще одну цікаву китайську гру «Цзяньшидзі». Давайте коротко сформулюємо правила гри:

З двох куп каменів два гравці по черзі можуть взяти:

Довільне ненульове кількість каменів з однієї купи.

При цьому однаковим довільним ненульовим числом каменів з обох паль.

Перемагає той, хто бере останній камінь самостійно.

Неважко визначити, в яких випадках виграє перший гравець, а в яких - другий. Завдання полягає в тому, щоб знайти так звані програшні пари (ai, bi), а значить, в купах каменів, що містять ai і bi stones, той, хто в даний момент робить свій хід, програє. Вам належить вирішити цю проблему, але в дещо іншій формі.

Давайте позначимо втрачає положення, для прогулянкового положення, коли в купах каменів а і б (а, б). Оскільки порядок паль не має значення, ми завжди будемо вважати, що ≤ б. Давайте впорядкуємо всі втрачають позиції в лексикографічному порядку, іншими словами:

(a1, b1) < (a2, b2), если (a1< a2) или ((a1= a2) и (b1< b2)).

Пронумеруємо пари, що програли, починаючи з 0. Ваше завдання: знайти k-ту пару програшних куп.

Вводу

У першому рядку вхідного файлу INTPUT.TXT знаходиться цифра N, (1 ≤ N ≤ 1000) - кількість тестів у файлі. Наступні N рядків містять числа ki, (0 ≤ ki ≤ 109) порядковий номер пари програшних куп, які ви хочете знайти.

Вихід

У вихідному файлі.TXT вивести N пар чисел (aki, bki) по одному на кожен рядок.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3012 | 0 01 23 5 |

### **Варіант 31**

У сучасній біології вченим часто доводиться стикатися з послідовностями ДНК. Ці послідовності часто дуже довгі, а їх ручна обробка вимагає багато часу і сил. Тому є ідея автоматизувати цей процес.

Для цього можна використовувати комп'ютерні методи обробки даних, наприклад, алгоритми на лініях дуже корисні. У цій задачі послідовність ДНК буде представлена у вигляді непорожнього рядка, всі символи якої входять в набір {A, G, C, T}.

Нехай буде дві послідовності ДНК: s = s1s2 ... sn і t = t1t2 ... Тм. Скажімо, що t може бути результатом еволюції з s, якщо s є підрядом t, тобто існує така послідовність індексів 1 ≤ i1 < i2 < ... < в ≤ m, що s1=ti1, s2=ti2, ... sn=олово. Необхідно з'ясувати, чи можна отримати послідовність t в результаті еволюції з s.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить послідовність s, другий - послідовність t. Розмір вхідного файлу не перевищує 256 кілобайт.

Вихід

У вихідному файлі OUTPUT.TXT виведіть слово YES, якби послідовність t могла розвинутися з s, а слово NO могло б розвинутися інакше.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | GTAAGCTA | Так |
| 2 | AAAGGAAAAAT | Ні |

### **Варіант 32**

Одного нейлера вигнали з роботи, і тепер йому потрібно якось заробляти на життя. Після деяких роздумів він вирішив, що може мати дуже хороші гроші, продаючи власне волосся. Відомо, що приймальні пункти купують волосся довільної довжини за вартістю C.U. за кожен сантиметр. Оскільки ринок волосся дуже динамічний, ціна на один сантиметр волосся змінюється щодня так само, як і курс валют. Неформальний - дуже хороший бізнес-аналітик. Він зміг підрахувати, якою буде ціна одного сантиметра волосся в кожному з наступних N днів (для зручності пронумеруємо дні в хронологічному порядку від 0 до N-1). Тепер він хоче визначити, в який з цих днів йому слід продати волосся, щоб заробити максимальну суму грошей після всіх N днів. Відзначимо, що волосся у непарнокопитних ростуть тільки вночі і виростають по 1 сантиметру за ніч. Також слід зазначити, що до 0-го дня неодружений з горем стриг волосся голим і до 0-го дня довжина його волосся становила 1 сантиметр.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить ціле число N (0 < N ≤ 100). У другому рядку заповнюється пробіл N натуральними числами, що не перевищують 100, що відповідають значенню C[i] 1 сантиметра волосся на кожен i-день.

Вихід

В один рядок вихідного файлу .TXT потрібно вивести максимальну суму грошей, яку неформальний може заробити за N днів.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 573 31 96 24 46 | 380 |
| 2 | 101 2 3 4 5 6 7 8 9 10 | 100 |
| 3 | 1010 9 8 7 6 5 4 3 2 1 | 55 |

### **Варіант 33**

Пітер вирішує головоломку, яка влаштована наступним чином. Наводиться квадратна таблиця розміру N×N, в кожній комірці якої написана якась англійська буква. Крім того, наводиться список ключових слів. Піту потрібно взяти ще одне ключове слово і знайти його в таблиці. Тобто знайдіть всі букви цього слова в таблиці, і вони повинні бути розташовані так, щоб осередок, в якій знаходиться кожна наступна буква слова, прилягала до клітинки, в якій написана попередня буква (осередки називаються сусідніми, якщо у них є загальна сторона - тобто вони примикають вертикально або горизонтально). Наприклад, на малюнку показано, як слово олімпіада може розташовуватися в таблиці.

Коли Петро знаходить слово, він викреслює його зі столу. Не можна використовувати букви, які вже були закреслені в інших ключових словах. Після того як всі ключові слова знайдені і закреслені, в таблиці залишається ще кілька букв, з яких Петро повинен скласти зашифроване в пазлі слово.

Допоможіть Петі у вирішенні цієї головоломки, написавши програму, яка згідно з цією таблицею і списком ключових слів буде виписувати, з яких букв Петро повинен додати слово, тобто які букви залишаться в таблиці після закреслення ключових слів.

Вводу

Вхідний файл INPUT.TXT містить два числа N (1 ≤ N ≤ 10) і M (0 ≤ M ≤ 100). Наступні N рядків на N великих англійських буквах описують головоломку. Наступні рядки М містять слова. Слова складаються тільки з великих англійських букв, кожне слово має довжину від 1 до 100 символів. Гарантовано, що всі ключові слова можна буде знайти і закреслити за правилами, описаним вище в таблиці.

Вихід

У вихідному файлі OUTPUT.TXT, в алфавітному порядку відображаються інші букви таблиці.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 5 3POLTERWYMSOAIPTBDANRLEMESOLYMPIADPROBLEMTEST | AENRSW |
| 2 | 3 2ISQABCIQWIIS | ABCQQW |

### **Варіант 34**

Як ви вже дізналися, Пітер любить програмувати. Нещодавно він вирішив реалізувати популярну карткову гру «Дурень». Але у Петі ще мало досвіду, йому терміново потрібна ваша допомога.

Як відомо, в «Дурня» грають з колодою з 36 карт. У програмі Petina кожна картка представлена у вигляді рядка з двох символів, де перший символ представляє ранг ('6', '7', '8', '9', 'T', 'J', 'Q', 'K', 'A') карти, а другий символ представляє масть ('S', 'C', 'D', 'H'). Ранги перераховані в порядку зростання пріоритету.

Піту потрібно вирішити наступну задачу: чи зможе гравець, маючи набір карт N, обіграти карти М, які роблять для нього хід? Для того щоб дати відсіч, гравцеві потрібно прикрити кожну з карт, за допомогою яких під ним робиться поворот картою зі своєї колоди. Карта може бути прикрита або старшою картою тієї ж масті, або картою козирного костюма. Якщо прихована карта сама по собі є козирем, то її можна прикрити тільки старшим козирем. Тільки одна картка може покривати одну картку.

Вводу

У першому рядку вхідного файлу INPUT.TXT знаходяться два натуральних числа N і M (N ≤ 35, M ≤ 4, M ≤ N), а також символ R, що означає козирний костюм. У другому рядку перераховані N карти в руках гравця. У третьому рядку перераховані карти М, які потрібно відбити. Всі карти відокремлені одна від одної одним простором.

Вихід

У вихідному файлі OUTPUT.TXT введіть "ТАК", якщо ви можете дати відсіч, або "НІ", якщо ви не можете.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 6 2 ХЗН КК АД 7С АХ 9С6Д 6С | Так |
| 2 | 4 1 D9S KC AH 7D8D | Ні |

### **Варіант 35**

Супутників в космосі в наші дні сотні, і всі вони обмінюються даними. При цьому система розпізнавання сигналу працює за схемою «Друг-Чужий». Один із супутників відправляє запит на інший супутник у форматі двох цілих чисел, а другий супутник відповідає на перший також двома цілими числами. Перші два числа першого супутника - це сума цифр і кількість цифр двох чисел, на які повинен відповісти другий супутник. При цьому відповіддю повинні бути цифри, що представляють найбільші і дрібні з можливих значень, які можуть бути сформовані за методикою, описаної вище.

Вам належить написати програму, яка формує відповідь для другого супутника за відомими номерами, отриманими з першого супутника.

Вводу

Вхідний файл INPUT.TXT містить 2 натуральних числа S і K, що представляють собою суму і кількість цифр відповідно (K ≤ 100). Гарантовано, що є можливість скласти хоча б одне K-значне число, сума цифр якого дорівнює S.

Вихід

У вихідному файлі.TXT виводимо дві цифри - відповідь другого супутника. Слід пам'ятати, що всі числа не мають нулів на початку.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 3 | 100 100 |
| 2 | 2 3 | 200 101 |
| 3 | 3 4 | 3000 1002 |

### **Варіант 36**

Васа вже втомилася від завдань на лініях! А ти? І що робити? Що ж, приступимо. Наводиться рядок з дрібних букв англійського алфавіту. Дозволяється переміщати будь-які його символи не більше, ніж на K положеннях в будь-яку сторону, щоб в кінцевому підсумку всі вони стояли в різних положеннях (за винятком випадку, коли K = 0). Наприклад, якщо рядок aababac і K = 2, то ви можете отримати рядки abaaabc або aaaabbc, але не зможете отримати aaacbab або aaaacbb.

Вася хоче переконатися, що рядок, отримана за допомогою такої операції, є мінімально можливою лексикографічно (тобто розташовується раніше за всіх за правилами упорядкування слів в словнику). Що йому робити?

Вводу

Перший рядок файлу INPUT.TXT встановлюється в K (K ≥ 0). У другому рядку вказується сама оригінальна непорожня рядок, довжиною не більше 100 000 дрібних англійських букв. Гарантовано, що K не перевищує довжину струни.

Вихід

У вихідному файлі .TXT вивести лексикографічно мінімальний з можливих результатів.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 2аабабак | aaaabbc |

### **Варіант 37**

Петро, вивчаючи, як змінюється курс рубля по відношенню до долара і євро, вивів закон, за яким відбуваються ці зміни (або думає, що він відкликав :)). Згідно з цим законом, Петя розрахував, яким буде курс рубля по відношенню до долара і євро в найближчі N днів.

Петя має 100 рублів. У кожен з днів він може обмінювати валюти один на одного за поточним курсом без обмеження суми (при цьому курс долара по відношенню до євро відповідає значенню, яке можна отримати, обмінявши долар на рублі, а потім ці рублі на євро). Оскільки Петро буде оперувати не готівковою валютою, а банківським рахунком, він може здійснювати обмінні операції з будь-яким (в тому числі і не цілим) кількістю одиниць будь-якої валюти.

Напишіть програму, яка підрахує, яку найбільшу кількість рублів Петро зможе отримати до кінця N-го дня.

Закони зміни валютного курсу влаштовані таким чином, що протягом зазначеного періоду рублевий еквівалент суми, яка може бути у Петі, не перевищить 108 рублів.

Вводу

Перший рядок файлу INPUT.TXT містить один N номер (1 ≤ N ≤ 5000). У кожному з наступних N рядків пишеться 2 цифри, розраховані за законами Петіна на відповідний день - скільки рублів буде коштувати 1 долар, а скільки рублів обійдеться в 1 євро. Всі ці значення не менше 0, 01 і не більше 10000. Значення задаються точно і виражаються дійсними числами не більше двох цифр після коми.

Вихід

У вихідному файлі OUTPUT.TXT відобразіть потрібне значення двома символами після десяткової коми.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 41 1010 5.535.53 1.256 5 | 4000.00 |

### **Варіант 38**

Дано N цілих чисел. Кожен з них може бути змінений один раз не більше ніж на ціле значення L як в сторону збільшення, так і в бік зменшення або залишення без змін. Якщо після такої операції деякі з чисел рівні, то вони зараховуються як одне ціле. З цими цифрами зазначена операція виконувалася таким чином, щоб залишалося мінімально можливу кількість чисел. Потрібно написати програму для визначення цієї суми.

Вводу

Вхідний файл INPUT.TXT містить в першому рядку натуральні числа L і N (N ≤ 100, L ≤ 3200), в другому рядку N числа (в діапазоні від -32768 до 32767), записані через пробіл.

Вихід

У вихідному файлі OUTPUT.TXT виведіть однину як відповідь на проблему.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 10 311 21 27 | 1 |
| 2 | 5 36 10 27 | 2 |

### **Варіант 39**

Ну, не гноми, а якесь покарання! — подумала Білосніжка, в черговий раз намагаючись покласти гномів спати. Один буде покладений, інший вже прокинувся! І так всю ніч. У Білосніжки є н гноми, і всі вони дуже різні. Вона знає, що для того, щоб укласти i-го гнома спати, потрібні хвилини штучного інтелекту, а після цього він буде спати рівно дві хвилини. Допоможіть Білосніжці дізнатися, чи зможе вона отримати хоча б хвилину відпочинку, коли всі гноми сплять, і якщо так, то в якому порядку укласти гномів спати.

Наприклад, нехай буде тільки два гноми, a1=1, b1=10, a2=10, b2=20. Якщо Білосніжка вперше почне класти першого карлика, то їй буде потрібно цілих 10 хвилин, щоб прокласти другий, і за цей час прокинеться перший. Якщо вона почне з другого карлика, то встигне закласти першого і отримати цілих 10 хвилин відпочинку.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить число n (1 ≤ n ≤ 105), другий рядок містить цифри a1, a2, . an, третій - числа b1, b2, . . . бн (1 ≤ ай, бі ≤ 109).

Вихід

У вихідному файлі.TXT виводимо n чисел, в якому ви хочете помістити гномів в ліжко. Якщо Білосніжка не може відпочити, відобразіть число −1.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 21 1010 20 | 2 1 |
| 2 | 210 1010 10 | -1 |

### **Варіант 40**

Розглянемо наступну цікаву гру для двох гравців. Для цієї гри потрібна таблиця з 2 рядків і N стовпців, в осередку яких записані натуральні числа, наступного вигляду:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| А1 | А2 | А3 | ... | Грн. |
| В1 | В2 | В3 | ... | Бн |

Гравці по черзі роблять ходи. Починає гру 1-й гравець.

В один оборот 1-й гравець виконує наступні дві дії:

вибирає довільну колонку (наприклад, j-y), яка ніколи не була обрана одним з гравців на попередніх ходах;

додає до своїх балів число Адж.

В один оборот 2-й гравець виконує наступні дві дії:

вибирає довільну колонку (наприклад, j-y), яка ніколи не була обрана одним з гравців на попередніх ходах;

додає до своїх балів число Bj.

Гра закінчується, коли один з гравців не в змозі зробити хід (з тієї причини, що всі колонки вже обрані). Спочатку у кожного з гравців по 0 очок.

Після закінчення гри відбувається взаєморозрахунки між гравцями. Наприклад, 1-й гравець набрав очки S1, а 2-й гравець набрав очки S2. У тому випадку, коли S1 > S2, 2-й гравець дає 1-му гравцеві S1-S2 UDE (звичайні валютні одиниці). В іншому випадку 1-й гравець віддає 2-го гравця S2-S1 в UDE. З цих позицій мета 1-го гравця - максимізувати розмір S1-S2, а мета 2-го гравця - максимізувати S2-S1.

Назвемо вартість гри значенням S1-S2 при оптимальній грі обох гравців. Напишіть програму, яка визначає вартість гри.

Вводу

У першому рядку вхідного файлу INPUT.TXT записується натуральне число N - кількість стовпців в таблиці (1 ≤ N ≤ 300000). Наступні N рядків описують числа у стовпцях таблиці. I-i цих рядків містить два натуральних числа Ai і Bi, розділених одним пробілом (1 ≤ Ai, Bi ≤ 3000).

Вихід

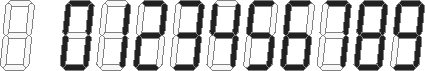
У файлі OUTPUT.TXT вивести одне ціле число - вартість гри.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 11 1 | 1 |
| 2 | 21 11 1 | 0 |
| 3 | 31 23 45 6 | 2 |

### **Варіант 41**

Нещодавно Вася придбав настільний калькулятор з рідкокристалічним індикатором. Цей показник відображає N цифр за допомогою N однакових елементів.



Відзначимо, що кожен елемент містить сім смуг, кожна з яких може бути як білою, так і чорною. Зокрема, при відображенні цифри «1» дві смуги мають чорний колір.

Вася - дуже допитливий хлопчик, тому йому хочеться знати, які максимальні і мінімальні N-цифри можуть відображатися на індикаторі його нового калькулятора, щоб чорними були саме смужки К.

Напишіть програму, яка знайде відповідь на питання Васіна. Майте на увазі, що числа не можуть містити нулі на початку.

Вводу

Вхідний файл INPUT.TXT містить два натуральних числа N і K (1 ≤ N ≤ 100, 1 ≤ K ≤ 700).

Вихід

У першому рядку вихідного файлу OUTPUT.TXT, відображають мінімальне число, у другому рядку вихідного файлу набирають максимальне число. Якщо жодне число не може бути представлено таким чином, вихідний файл повинен містити один рядок NO SOLUTION.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 5 15 | 1011797111 |
| 2 | 10 1 | НЕМАЄ РІШЕННЯ |

### **Варіант 42**

Для зберігання двох агресивних рідин А і В використовується ємність з багатошарової перегородкою, яка виготовляється з існуючих N листів. Для кожного аркуша i (i = 1, ..., N) відомий час його розчинення рідиною А – ай і рідиною В – бі. Розчинення перегородки кожної з рідин відбувається послідовно лист за листом, з постійною швидкістю по товщині листа.

Потрібно написати програму для проектування такого розділу, час розчинення якої було б максимальним.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить цифру N (1 ≤ N ≤ 256). Кожна з наступних N рядків містить два додатних дійсних числа ai і bi, розділених пробілом (числа не перевищують 106 і складаються не більше ніж з 11 значущих цифр).

Вихід

Перший рядок вихідного файлу.TXT записувати час розчинення розділу з точністю не менше 10-3. У наступному рядку файлу запишіть номери аркушів в тому порядку, в якому вони розташовані від рідкої А до рідкої В, розділивши цифри пробілами.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 41 21 20.5 1.57 3.5 | 6.0004 2 1 3 |

### **Варіант 43**

У багатьох торгових фірмах є посаду менеджера із закупівель. Як відомо, вони займаються тим, що згідно з торговим планом, що представляє собою перелік найменувань товарів, для кожного з яких вказується необхідна кількість, закуповують зазначений в ньому товар на оптових складах. Торговий план складається керівництвом компанії. Іноді у менеджерів із закупівель виникає бажання принести користь не тільки своїй компанії, але і собі.

Якраз зараз, відразу після підписання чергового торгового плану на замовлення n найменувань товарів, відкрилася нова оптова база. Як це часто буває відразу після відкриття, його ціни на багато товарів нижче закладених в плані. Я думаю, ви можете скористатися цим.

Гроші виділялися на закупівлю товару з припущенням, що всі товари будуть закуповуватися на старій оптовій базі. Менеджер хоче, скориставшись можливістю купити товар на новій базі, витратити якомога менше коштів на покупку необхідної кількості товару (невитрачені гроші він, звичайно ж, зможе взяти собі).

Щоб не викликати сильних підозр, покупки не слід проводити на новій основі, загальна планова вартість якої становила більше d грошових одиниць. Залишилося тільки порахувати, який товар і в якій кількості слід придбати на новій базі, щоб грошей було якомога більше не витрачено.

Вводу

Перший рядок містить чотири числа: n (1 ≤ n ≤ 1000), d, а також k1 і k2 (1 ≤ k1, k2 ≤ 1000) - кількість найменувань товарів, наявних на раніше відкритій і новій базі відповідно. Після цього йдуть n рядків, кожна з яких містить назву товару в плані і його кількість (додатне дійсне число). За ними слід два блоки ліній k1 і k2 відповідно відокремлені від попереднього і розділені рядком перекладу - назвами товарів за базами і цінами за одиницю товару відповідно. Всі ціни - це позитивні цифри, навіть на новій базі.

Назви продуктів складаються не більше ніж з 100 англійських букв і підкреслень, і не чутливі до регістру. Дійсні числа даються не більше ніж двома цифрами після десяткової коми і не більше 106. Гарантовано, що всю продукцію з плану можна буде придбати на старій базі. Враховуйте, що будь-який товар на будь-якій базі можна купити в будь-якій дробовій кількості, а також, що будь-яке дробове число грошових одиниць можна використовувати при оплаті.

Вихід

У вихідному файлі OUTPUT.TXT відображати n дійсних чисел, по одному на рядок, які вказують кількість відповідного товару, придбаного в новій базі. На i-му рядку відобразіть кількість товару, що йде i-го в плані. Помилки, менші за 0,01, будуть проігноровані.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 4 11.00 5 4Солт 3суГар 0.5Powdered\_MilK 7fLoUR 8flOuR 1CINNAmOn 8saLt 0.5SuGAr 20poWdErED\_MILk 3FLOur 2SALT 0.4liQUID\_HydRoGen 10000sUgAR 10 | 2.00000.50000.00000.0000 |

### **Варіант 44**

Антон працює кур'єром. У нього багато замовлень. Антону потрібно рівно один день, щоб виконати одне замовлення. Для кожного замовлення визначається вартість і термін його виконання (кількість днів, що залишилися до запланованого дня виконання замовлення). Прокинувшись, Антон вивчив його графік і зрозумів, що, можливо, він не зможе виконати всі накази, і його можуть звільнити. Тому він вирішив виконати лише деякі з них таким чином, щоб отримувати максимальний дохід.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить ціле число N (1 ≤ N ≤ 1000) – кількість замовлень. Потім рядки N описують дані кожного порядку Ti і Ci (натуральні числа, що не перевищують 105). Там, де Ti - останній день, в який ще можна виконати замовлення, Ci - нагорода за виконання замовлення.

Вихід

У вихідному файлі OUTPUT.TXT відображають максимальну винагороду, яку можна отримати, виконуючи замовлення.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 21 102 12 | 22 |
| 2 | 31 101 203 24 | 44 |

### **Варіант 45**

Тітка Люба тільки що випрала всю білизну і тепер перед нею стоїть непросте завдання - як висушити її, щоб жодна річ не встигла зіпсуватися. Відразу після прання річ, яку я прав, має вологість wi. Якщо його сушити на мотузці, то через хвилину його вологість знижується на 1, а якщо на батареї - то на r (якщо вологість була менше r, то вона стає дорівнює 0). Більш того, у тітки Люби багато мотузок (досить сушити всі речі одночасно), а батарея всього одна, причому настільки маленька, що на ній не можна сушити дві речі одночасно. Річ i-I зіпсується, якщо вона не висохне вчасно ді. Допоможіть тітці Любі скласти план, коли повісити яку річ на батарею.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить цілі числа n (1 ≤ n ≤ 105) - кількість мокрих речей, і r (1 ≤ r ≤ 109). Наступні n рядків містять описи випраних речей – пари чисел wi і di (1 ≤ wi, di ≤ 109).

Вихід

У вихідному файлі OUTPUT.TXT виведіть план сушіння у вигляді пар цілих чисел ti і ki, де ti - час в хвилинах від початку сушіння, а ki - номер речі, яку потрібно повісити на батарею в цей момент. Вихідні пари в порядку збільшення ti. Пар не повинно бути більше 105. Не виводьте числа, більші за 109. Якщо немає можливості висушити всі речі, приберіть слово «Неможливо».

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3 32000 10002000 20002500 1500 | 0 3500 11000 3 |
| 2 | 3 32000 10002000 10002000 1000 | Неможливий |

### **Варіант 46**

У зв'язку з міжпланетним турніром з шашок було прийнято рішення про будівництво орбітального готелю. Це повинен був бути великий куб з блоків N×N×N – маленьких кубиків по 1×1×1, причому кожен блок повинен був бути пофарбований зовні з усіх боків в один колір. При цьому деякі блоки могли бути пофарбовані в один колір.

Через рік були зроблені фотографії готелю з кожної з 6 сторін: спереду, зліва, ззаду, справа, зверху, знизу. За рік експлуатації могло статися так, що через крихке кріплення частина блоків, з яких був побудований готель, відкололася і полетіла у відкритий космос. Комісія з реставрації готелю хоче встановити максимально можливу кількість залишилися блоків від зроблених фотографій.

Отже, потрібно скористатися типами готелів (cube N×N×N, з яких, можливо, з 6 сторін викидаються деякі кубики 1×1×1), щоб визначити, з якого максимального числа блоків 1×1×1 він може складатися. Може виявитися, що готель складається з двох і більше не пов'язаних між собою частин.

Вводу

У першому рядку вхідного файлу INPUT.TXT знаходиться цифра N - розмір готелю (1≤N≤10). На наступних N рядках написані види готелю з 6 сторін (в наступному порядку: спереду, зліва, ззаду, справа, зверху, знизу). Кожен такий вид являє собою таблицю N×N, в якій різними великими англійськими літерами позначаються різні кольори, а символ «». (крапка) - те, що в цьому місці можна заглянути прямо через готель. Два послідовних вигляду відокремлюються один від одного рівно по одному простору в кожному з N рядків.

Нижня межа верхнього виду відповідає верхній межі переднього виду, а верхня межа нижнього виду - нижній межі переднього виду. Для переднього, заднього і бічного видів верхня і нижня частина виду відповідають верху і низу готелю.

Вхідні дані вірні, тобто вхідний файл описує стан, який можна отримати.

Вихід

Висновок у вихідний файл.TXT одне число - це бажана максимальна кількість блоків, що залишилися в готелі.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3.Р. YR .Y. RYY .Y. . R.GRB YGR BYG RBY GYB GRB.R. YRR .Y. RRY .R. . Y. | 11 |
| 2 | 2ZZ ZZ ZZ ZZ ZZ ZZ ZZ ZZ ZZ | 8 |

### **Варіант 47**

Толік придумав нову технологію програмування. Він хоче умовити своїх друзів скористатися ним. Однак все не так просто. I-й друг погодиться використовувати технологію Толіка, якщо його авторитет не менше, ніж ai (авторитет виражається цілим числом). Як тільки він почне його використовувати, до авторитету Толіка додасться число bi (є люди, у яких bi < 0). Допоможіть Толіку провести якомога більше його друзів по істинному шляху.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить два числа: n (1 ≤ n ≤ 1000) – кількість друзів Толіка, і початковий авторитет Толіка. Наступні n рядків містять пари чисел ai та bi. Всі числа є цілими числами за модулем не більше 106.

Вихід

У вихідний файл.TXT, виводимо число m - максимальну кількість друзів, яке може залучити Толік, а потім m цифри - числа друзів в тому порядку, в якому їх потрібно збудити.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 5 11 36 -56 -42 22 -1 | 41 4 3 5 |

### **Варіант 48**

По колу є N коробок. У кожній коробці є один правий і один лівий сусід. У i-й коробці знаходиться кульки Ай. Відомо, що загальна кількість кульок у всіх ящиках не перевищує Н. За один оборот допускається перекладання одного м'яча з коробки в наступний. Яка найменша кількість ходів, які доведеться зробити, щоб в кожній коробці було не більше одного м'яча?

Вводу

Перший рядок файлу INPUT.TXT містить ціле число N (1 ≤ N ≤ 1000). Другий рядок визначає послідовність N цілих чисел A1, A2, ... , AN (0 ≤ Ai ≤ N). Сума всіх значень ШІ не перевищує N.

Вихід

У вихідному файлі OUTPUT.TXT відображати мінімальну кількість переміщень, які ви шукаєте.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 71 0 0 0 2 3 1 | 7 |

### **Варіант 49**

Вказується натуральне число N. Потрібно написати програму, яка розраховує кількість різних тризначних чисел, отриманих з N шляхом викреслення цифр зі своєї десяткової нотації.

Вводу

Вхідний текстовий файл INPUT.TXT містить одне натуральне число N (1 ≤ N ≤ 10100).

Вихід

Вихідний текстовий файл OUTPUT.TXT повинен містити одне ціле число - знайдене число тризначних чисел.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 12 | 0 |
| 2 | 111111111110011111111 | 4 |

### **Варіант 50**

Знайдіть ряд способів представити його як суму чотирьох натуральних чисел: x = a + b + c + d, де a ≤ b ≤ c ≤ d.

Вводу

Вхідний файл INPUT.TXT містить ціле число x (1 ≤ x ≤ 1500).

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3 | 0 |
| 2 | 5 | 1 |

### **Варіант 51**

Вказується дві клітинки шахівниці. Необхідно визначити, чи можна одним рухом шахового коня дістатися з однієї клітини в іншу, а якщо немає, то необхідно з'ясувати, чи можна двома ходами дістатися з однієї клітини в іншу.

Вводу

Вхідний файл INPUT.TXT містить координати двох осередків в загальному форматі: кожна координата записується у вигляді англійської малої літери і цифри, координати відокремлюються один від одного комою і пробілом.

Вихід

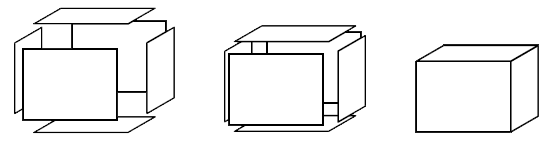
Вихідний файл OUTPUT.TXT повинен містити цифру "1", якщо є можливість перейти з однієї комірки в іншу за 1 хід, або цифру "2", якщо ви можете отримати за 2 ходи, або "НІ", якщо одна комірка недоступна з іншої за 1 або 2 переміщення.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | а1, ч8 | Ні |
| 2 | а1, б3 | 1 |
| 3 | а1, д4 | 2 |

### **Варіант 52**

Іван працює на заводі, який випускає важку техніку. Його робота дуже проста – він збирає коробки і упаковує в них обладнання для клієнтів. Кожна така коробка являє собою паралелепіпед. Для складання короба Іван використовує шість прямокутних дерев'яних плиток. Кожна пластина представляє одну зі сторін коробки.



Петро підбирає плитку для Івана. Петро недостатньо розумний і тому часто робить помилки - він приносить Івану таку плитку, з якої неможливо зібрати коробку. Але Іван не довіряє Петру. Тому він завжди витрачає багато часу, пояснюючи Петру, де він припустився помилки.

На щастя, Пітер любить все, що пов'язано з комп'ютерами, і вважає, що комп'ютери ніколи не роблять помилок. Іван вирішив, що може використовувати це в своїй роботі. Іван попросив вас написати програму, яка підкаже за вказаними розмірами з шести плиток: чи можна з них побудувати коробку.

Вводу

Вхідний файл INPUT.TXT містить шість рядків, кожен з яких містить два натуральних числа w і h (1 ≤ w, h ≤ 10 000) – ширину і висоту пластини в міліметрах.

Вихід

У вихідному файлі OUTPUT.TXT введіть "POSSIBLE", якщо є можливість зібрати коробку з цих пластин, і "IMPOSSIBLE" інакше.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1345 25842584 6832584 1345683 1345683 13452584 683 | МОЖЛИВИЙ |
| 2 | 1234 45671234 45674567 43214322 45674321 12344321 1234 | НЕМОЖЛИВИЙ |

### **Варіант 53**

Діаграма, діаграма сонячного проміння

Опис автоматично генерується із середньою впевненістю

Гра в дартс дуже популярна в Великобританії і Голландії. У грі беруть участь кілька гравців. Вони по черзі кидають три дротики в ціль.

На початку гри у кожного гравця є певна кількість очок, зазвичай 501. За кожне попадання дротика в ціль кількість гравця зменшується на певне число, в залежності від того, в яку частину цілі він влучив. Першим, хто досяг нульових балів, вважається переможець.

Зовнішній вигляд цілі показаний на малюнку праворуч. Він розділений на 20 секторів, розташованих навколо невеликого центрального кола. Це коло, в свою чергу, ділиться на внутрішню і зовнішню частину (іноді внутрішню частину називають «яблуком»). Попадання на зовнішню сторону центрального кола оцінюється в 25 очок, а в «бичаче око» - в два рази більше, тобто в 50 балів. Значення сектора дорівнює тому числу, яке на ньому записано. Крім того, на цілі виділяються два кільця - зовнішнє і внутрішнє. Вони оцінюються в два і три рази вище, відповідно, ніж решта відповідного сектора.

Існують додаткові правила для останньої серії кидків, при яких гравець повинен досягти нульових очок. У цій серії гравцеві доведеться кинути в ціль від одного до трьох дротиків. Гравець повинен досягти рівно нульових очок, отримання негативної суми вважається помилкою. Останній дротик повинен бути «подвійним», тобто потрапити в зовнішнє кільце будь-якого сектора або в «бичаче око» - (вважається подвоєнням зовнішньої частини центрального кола).

Наприклад, один з правильних способів закінчити гру з 50 очками - кинути дротики на «18» і «Д16».

Способи «Д20», «10», або «20», «Т10» не підходять: останній кидок не подвоюється. Ще один можливий спосіб перемогти в цьому випадку - просто вдарити Бика. За кількістю очок, що залишилися, знайдіть всі способи правильно закінчити гру.

Вводу

Вхідний файл INPUT.TXT містить число n - кількість точок, що залишилися (1 ≤ n ≤ 200).

Вихід

У першому рядку вихідного файлу OUTPUT.TXT, output k - кількість способів правильно закінчити партію. Кожна з наступних k рядків повинна містити опис одного правильного способу. При цьому цифра від 1 до 20 відповідає потраплянню у відповідний сектор. Буква «Д» перед цифрою вказує на попадання в зовнішнє (подвоєння) кільце, а «Т» - на внутрішнє (потрійне). Зовнішня частина центрального кола позначається як «25», а «бичаче око» позначається словом «Бик».

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 5 | 71 Д1 Д11 2 Д11 Д2Д1 1 Д1Т1 Д12 1 Д13 Д1 |

### **Варіант 54**

Описується рекурсивна функція з трьома параметрами F(a, b, c):

если a ≤ 0 або b ≤ 0 или c ≤ 0, то F(a, b, c) = 1

если a > 20 или b > 20 или c > 20, то F(a, b, c) = F(20, 20, 20)

если a < b и b < c, то F(a, b, c) = F(a, b, c-1) + F(a, b-1, c-1) - F(a, b-1, c)

иначе F(a, b, c) = F(a-1, b, c) + F(a-1, b-1, c) + F(a-1, b, c-1) - F(a-1, b-1, c-1)

Однак якщо зазначена функція реалізована безпосередньо, то навіть для невеликих значень a, b і c (наприклад, a = 15, b = 15, c = 15) програма буде працювати кілька годин! Необхідно реалізувати ефективний алгоритм розрахунку функції F, який встигне знайти будь-яке її значення менш ніж за одну секунду!

Вводу

Входной файл INPUT.TXT содержит три целых числа a, b, c - параметры функции F (-104 ≤ a,b,c ≤ 104).

Вихід

У вихідному файлі OUTPUT.TXT вивести, вивести значення функції F(a, b, c).

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 1 1 | 2 |
| 2 | 2 2 2 | 4 |
| 3 | 10 4 6 | 523 |
| 4 | 50 50 50 | 1048576 |

### **Варіант 55**

Вася вчиться в третьому класі і зараз здає тему «Прості дроби з натуральним чисельником і знаменником». Виходить, що дріб називається регулярним, якщо його чисельник менше знаменника, і невимовним, якщо чисельник і знаменник взаємно прості. Вася дуже любить математику і тому вдома вирішує безліч завдань. На даний момент Вася шукає найбільший правильний непередаваний дріб, в якому сума чисельника і знаменника дорівнює N.

Потрібно написати програму, яка допоможе Васі вирішити цю проблему.

Вводу

Вхідний файл INPUT.TXT містить одне ціле число N (3 ≤ N ≤ 2∙109).

Вихід

Вихід вихідного файлу.TXT повинен містити два числа, чисельник і знаменник знайденого дробу, розділені пробілом.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3 | 1 2 |
| 2 | 10 | 3 7 |

### **Варіант 56**

У багатьох книгах з цікавої математики дана така задача. Розмістіть по периметру трикутної кімнати 3 стільці так, щоб на кожній стіні їх було по 2. Її рішення - поставити стілець в кожному кутку кімнати.

Це завдання легко узагальнюється. Нехай кімната буде трикутником ABC. Данці: загальна кількість n стільців, кількість стільців nAB, які повинні стояти біля стіни ab, кількість стільців nBC, які повинні стояти біля стіни BC, кількість стільців nAC, які повинні стояти біля стіни змінного струму. Необхідно знайти відповідне розташування стільців або встановити, що його не існує. При цьому стільці можна розміщувати тільки в кутах кімнати і уздовж стін, стільці не можна розміщувати в центрі кімнати. У будь-якому з кутів можна поставити довільну кількість стільців.

Вводу

Входной файл INPUT.TXT содержит целые числа n, nAB, nBC, nAC (0 ≤ n, nAB, nBC, nAC ≤ 1000).

Вихід

У першому рядку вихідного файлу .TXT, введіть NO, якщо стільці не можуть бути розташовані зазначеним чином. В іншому випадку виведіть YES в перший рядок вихідного файлу, а в другому рядку виведіть 6 ненегативних цілих чисел: kA, kAB, kB, kBC, kC, kAC - відповідно, кількість стільців, які потрібно розмістити в кутку А, уздовж стіни АВ, в кутку В, уздовж стіни БК, в кутку С і уздовж стіни ЗМІННОГО СТРУМУ.

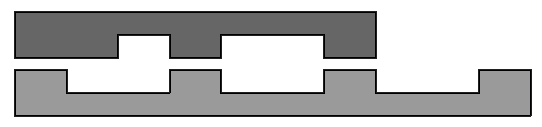
Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3 2 2 2 | ТАК1 0 1 0 1 0 |
| 2 | 3 3 2 2 | Ні |

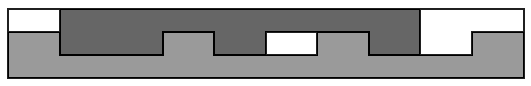
### **Варіант 57**

Науково-дослідна лабораторія відомої автомобільної компанії розробила спеціальний механізм, що дозволяє підвищити ефективність гальм за рахунок рівномірного завантаження деталей, що використовуються в гальмах.

Однією з основних складових механізму є 2 прокладки, які в процесі взаємодії накладаються один на одного. Кожна подушечка довжиною n ділиться на n секцій, кожна з яких має висоту h або 2h. Таким чином, подушечки мають зубчасту форму без заокруглення.



У процесі взаємодії прокладок важливо, щоб вони накладалися один на одного і при цьому загальна довжина отриманого з'єднання була найменшою.



За заданою конфігурації прокладок потрібно визначити найменшу довжину їх можливого з'єднання, при якій загальна висота конструкції не перевищує значення 3ч. При цьому забороняється обертати прокладки і видаляти зуби.

Вводу

Вхідний файл INPUT.TXT містить 2 рядки, що описують конфігурацію з 2 колодок. Кожна конфігурація визначається послідовністю цифр 1 і 2, відповідних висоті кожного зуба прокладки. Кожна з ліній не порожня і має довжину, що не перевищує 100.

Вихід

У вихідному файлі OUTPUT.TXT ви хочете вивести найменшу довжину конструкції із вказаних колодок.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 21121121122212112 | 10 |
| 2 | 1212121221212121 | 8 |
| 3 | 221122112221212 | 15 |

### **Варіант 58**

У вас є N каменів з масами W1, W2, ... WN. Потрібно розкласти камені на 2 купи, щоб різниця в масах цих паль була мінімальною.

Вводу

У першому рядку вхідного файлу INPUT.TXT записується число N – кількість каменів (1 ≤ N ≤ 18). У другій лінії маси каменів W1, W2, ... WN (1 ≤ Wi ≤ 105).

Вихід

В один рядок вихідного файлу.TXT потрібно вивести одне ненегативне ціле число - мінімально можливу різницю між масами в два ворсу.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 55 8 13 27 14 | 3 |

### **Варіант 59**

Назвемо перестановку N чисел 1... N K-перестановка, якщо будь-які два суміжні елементи в ній відрізняються не більше ніж K.

Необхідно знайти кількість K-перестановок з N чисел від 1 до N.

Вводу

Вхідний файл INPUT.TXT містить натуральні числа N і K (K ≤ N ≤ 9).

Вихід

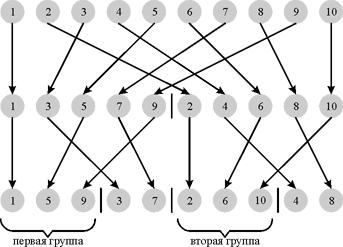
У файлі OUTPUT.TXT виводимо ціле число - відповідь на проблему.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 3 1 | 2 |
| 2 | 4 2 | 12 |

### **Варіант 60**

Легіон є основною організаційною одиницею в армії Стародавнього Риму. У різний час легіони мали різну чисельність і різні формування. Найпростішою конструкцією була лінія. Для того щоб відібрати трьох солдатів з N солдатів легіону, що вишикувалися в лінію для розвідки, виконувалися такі операції: якщо в лінії було більше трьох солдатів, то ряд ділився на два, один з яких складається з солдатів, що стоять на парних позиціях, а другий - з стоять на непарних позиціях. При всіх отриманих лініях ця процедура повторювалася до тих пір, поки в кожній з них не залишилося не більше трьох солдатів. Якщо солдатів залишилося троє, то цю групу можна відправити на розвідку.



Необхідно визначити, скільки груп з трьох осіб може утворитися з вихідної лінії.

Вводу

Вхідний файл INPUT.TXT містить єдине ціле число N - кількість солдатів в ряду (0 ≤ N ≤ 1018).

Вихід

У вихідному файлі OUTPUT.TXT вивести одне ціле число, кількість груп по три людини, утворені з вихідного рядка.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 10 | 2 |
| 2 | 4 | 0 |

### **Варіант 61**

Дано рядок, що складається з N пар різних символів. Ви хочете відобразити всі перестановки символів у цьому рядку.

Вводу

Вхідний файл INPUT.TXT містить рядок, що складається з N символів (1 ≤ N ≤ 8), символів - букв англійського алфавіту і цифр.

Вихід

У вихідному файлі OUTPUT.TXT вивести по одній перестановці на кожен рядок. Перестановки можна зробити висновок в будь-якому порядку. Не повинно бути повторень і рядків, які не є перестановками вихідного.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | Викл. | Абба |
| 2 | IOX | XOIOIXOXIOOXIIOX |

### **Варіант 62**

Марсіани Міша і Маша вирішили разом підібрати подарунок на день народження Каті. Коли вони нарешті знайшли бажане і упакували річ в красиву коробку, їм довелося вирішити, як підписати подарунок. Друзі думали, що найкращим рішенням буде складання загальної підпису так, щоб вона містила їх імена в якості підструнків.

Майте на увазі, що на Марсі прийнято розписуватися повними іменами, а для марсіан вони можуть бути досить довгими.

Вводу

Вхідний файл INPUT.TXT містить два рядки, в яких записуються повні імена друзів. Імена, як не дивно, складаються з букв англійського алфавіту, з яких тільки перша - велика. Довжина імені коливається від 1 до 1000 символів.

Вихід

У вихідному файлі.TXT виводимо найкоротший рядок, в якій одночасно зустрічаються імена Міша і Маша. Букви, з яких починаються імена в цьому рядку, слід зробити великими. Якщо рішень кілька, відобразіть те, яке менше в алфавітному порядку (припустимо, що будь-яка велика буква менше будь-якої малої літери).

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | МішаМаша | МашаМіша |
| 2 | ЛипняЛяля | Юляля |

### **Варіант 63**

Король Флетленду вирішив вирубати частину дерев, що ростуть перед його палацом. Дерева перед царським палацом висаджені в ряд, там ростуть n дерев, відстані між сусідніми деревами однакові.

Після вирубки перед палацом повинні залишитися м дерев, а відстані між сусідніми деревами повинні бути однаковими. Допоможіть королю розібратися, скільки способів вирубки дерев.

Потрібно написати програму, яка за заданими числами n і m визначить, скільки є способів вирубати частину n дерев, щоб після вирубки залишилося m дерев і сусідні дерева виявилися на рівній відстані один від одного.

Вводу

Вхідний файл INPUT.TXT містить два цілих числа n і m (0 ≤ m, n ≤ 1000).

Вихід

В один рядок вихідного файлу.TXT потрібно вивести одне ціле число - кількість шляхів, які ви шукаєте.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 5 3 | 4 |

Пояснення прикладу

Якщо умовно позначити початкове місце розташування дерев перед палацом як «ТТТТТ», то можливі результати після вирубки такі: «ТТТ..», «.». ТТТ.», «.. ТТТ», «Т.Т.Т».

### **Варіант 64**

Дано рядок, що складається з N символів. Ви хочете відобразити всі перестановки символів у цьому рядку.

Вводу

Вхідний файл INPUT.TXT містить рядок, що складається з N символів (1 ≤ N ≤ 8), символів - букв англійського алфавіту і цифр.

Вихід

У вихідному файлі OUTPUT.TXT вивести по одній перестановці на кожен рядок. Перестановки можна зробити висновок в будь-якому порядку. Не повинно бути повторень і рядків, які не є перестановками вихідного.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | Викл. | Абба |
| 2 | 122 | 122212221 |

### **Варіант 65**

Дається три числа: a, b, c. Необхідно з'ясувати, чи можна переставити цифри в числах a і b так, щоб сума була c.

Вводу

Входной файл INPUT.TXT содержит три целых числа: a, b, c (0 ≤ a, b, c < 109). Числа разделены пробелом.

Вихід

У вихідному файлі OUTPUT.TXT повинні бути відображені ТАК, якщо можлива потрібна перестановка цифр, в іншому випадку необхідно вивести NO. Якщо відповідь позитивна, в другому рядку має відображатися число x, отримане шляхом перестановки цифр числа a, а число y, отримане шляхом перестановки цифр числа b, сума якого дорівнює c. Числа x і y на виході не повинні містити початкових нулів. Числа в рядку розділені пробілом. Якщо рішень кілька, то слід вивести пару, в якій число х мінімально.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 12 31 25 | ТАК12 13 |
| 2 | 12 31 26 | Ні |
| 3 | 101 2 13 | ТАК11 2 |

### **Варіант 66**

Ми всі грали в кості в дитинстві. У нас були кубики з цифрами, і кубики з буквами. Існували і різнокольорові кубики. Маленький Андрій також любить грати в кості. У нього є кілька наборів кубиків, причому всі кубики з одного набору пофарбовані однаково, а кубики з різних наборів пофарбовані по-різному.

У Андрія на столі два кубики. Допоможіть йому визначити, чи належать вони до одного набору чи ні. Стіл Андрія скляний, тому він бачить кольори всіх граней куба. Куби належать до однієї множини, якщо одну з них можна перевести в іншу комбінацією обертань навколо осей, що проходять через середини протилежних граней.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить опис першого кубика в форматі: колір лицьової сторони, колір задньої грані, колір верхньої грані, колір нижньої грані, колір лівої грані, колір правої грані. У другому рядку йде опис другого куба в тому ж форматі. Кольори граней кодуються цифрами. Всі числа у вхідному файлі цілі, додатні і не перевищують 100.

Вихід

У вихідному файлі OUTPUT.TXT введіть "YES", якщо перший і другий куби належать одному набору, і "NO" інакше.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 2 3 4 5 61 2 3 4 5 6 | Так |
| 2 | 1 2 3 4 5 61 1 1 1 1 1 | Ні |
| 3 | 1 2 3 4 5 65 6 3 4 2 1 | Так |

### **Варіант 67**

Перестановка P[1..n] розміру n - це набір чисел від 1 до n, розташованих в певному порядку. При цьому він повинен містити рівно один раз кожну з цих цифр. Прикладами перестановок є 1,3,4,5,2 (для n=5) і 3,2,1 (для n=3), а, наприклад, 1,2,3,4,5,1 не є перестановкою, так як число 1 зустрічається двічі.

Число i називається фіксованою точкою для перестановки P, якщо P[i] = i. Наприклад, в перестановці 1,3,4,2,5 є рівно дві фіксовані точки: 1 і 5, а перестановка 4,3,2,1 не має фіксованих точок.

Дається два числа: n і k. Знайти кількість перестановок розміру n з точно k фіксованими точками.

Вводу

Вхідний файл INPUT.TXT містить два цілих числа n (1 ≤ n ≤ 9) і k (0 ≤ k ≤ n).

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 5 2 | 20 |
| 2 | 9 6 | 168 |
| 3 | 2 1 | 0 |
| 4 | 9 0 | 133496 |

### **Варіант 68**

Наближається Новий рік, і в магазинах починають з'являтися різні ялинкові прикраси. На полицях можна побачити різні кулі, шишки, зірочки, але все ж найкрасивішою прикрасою є гірлянда з різнокольорових лампочок. Одна з компаній, що займаються виготовленням ялинкових прикрас, вирішила в цьому році виготовити гірлянди на замовлення.

Гірлянди, що випускаються цією фірмою, складаються з лампочок різних кольорів, з'єднаних проводами. Всього в гірлянді є n лампочок, кожна з яких пофарбована в один з k кольорів, і m проводів (кожен провід з'єднує рівно дві лампочки). Далі ми будемо вважати, що цибулини нумеруються натуральними числами від 1 до n.

На жаль, далеко не кожна конструкція гірлянди відповідає естетичним уявленням покупців. По-перше, лампочки, з'єднані одним проводом, повинні бути різних кольорів, а по-друге, конфігурація самої гірлянди (тобто які лампочки і як з'єднуються проводами) не може бути ніякої.

Один з відділів компанії вже провів дослідження і знайшов найбільш «вдалу» конфігурацію. Ваше завдання - знайти ряд способів фарбування цибулин, щоб отримана гірлянда задовольняла естетичні погляди покупців.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить три цілих числа: n, k, m (1 ≤ n, k ≤ 8, 0 ≤ m, ≤ 10). Наступні m рядків описують дроти. Опис кожного проводу складається з двох чисел u і v (1 ≤ u, v ≤ n, u ≠ v) – номерів лампочок, з'єднаних цим проводом.

Вихід

У вихідному файлі OUTPUT.TXT виведіть відповідь на завдання.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 2 2 11 2 | 2 |
| 2 | 4 4 0 | 256 |
| 3 | 4 4 61 21 31 42 32 43 4 | 24 |

### **Варіант 69**

Деякі набори з n слів довжини n мають цікаву властивість - їх можна розташувати в осередках квадрата n×n так, що всі слова множини можна читати як по вертикалі, так і по горизонталі.

Прикладом такого набору слів є {"ДАТА", "ЗНАХІДКА", "ІДЕЯ", "ДАЛІ"}. Їх можна оформити наступним чином:

Зображення, що містить текст, табло, екран, електроніку

Опис генерується автоматично

Відзначимо, що кожне слово можна читати як по горизонталі, так і по вертикалі. Такі квадрати називаються словниковими квадратами, найбільший відомий словниковий квадрат англійською мовою має розмір 10×10.

Розглянемо ще один приклад словникового квадрата:

Знімок екрана комп'ютера

Опис автоматично генерується з низькою впевненістю

Вам даються такі 2n слів, що з них ви зможете побудувати два різних словникових квадрата розміром n×n. Ваше завдання - розділити ці слова на дві групи, n слів в кожній, і побудувати словниковий квадрат зі слів кожної групи.

Гарантовано, що всі слова, дані вам, є англійськими словами (деякі з них можуть бути досить рідкісними словами, іменами або спеціальними термінами).

Вводу

Перший рядок файлу INPUT.TXT містить ціле число n (2 ≤ n ≤ 10). Кожна з наступних 2n рядків містить слово, що складається з великих букв англійського алфавіту. Кожне слово містить рівно n букв.

Вихід

У вихідному файлі OUTPUT.TXT відображаються два квадрати словника, побудовані на основі даних слова. Відокремте квадрати порожнім рядком.

Приклад

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 4ARTSBESTCRABDATEFINDIDEANEXTRARE | CRABRAREARTSBESTFINDIDEANEXTDATE |

### **Варіант 70**

Студент Діма – студентка другого курсу університету. На другому курсі його університету викладається курс математичної логіки. У цьому курсі особлива увага зосереджена на автоматичних доказах - програмах, що дозволяють використовувати їх для доведення різних складних теорем.

Курсова робота Діми з математичної логіки полягає в наступному: необхідно довести за допомогою автоматичного доказу ХОЛ, що шахівниця розміром 2Н х 2Н, з якої вирізається одна клітина, може бути покрита в один шар кутами з трьох осередків.

Діма не вірить, що це правда, і намагається зробити контрприклад. Ваше завдання - довести, що Діма неправий, і вирішити проблему введення Діміна.

Вводу

Вхідний файл INPUT.TXT містить три натуральних числа N, X, Y (N ≤ 6; X,Y ≤ 2N). Ці цифри задають дошку 2N x 2N, з якої вирізається осередок з координатами (X, Y). X - горизонтальна координата, Y - вертикальна, (1, 1) - верхній лівий кут дошки.

Вихід

У вихідному файлі OUTPUT.TXT вивести 2N рядки на цифри 2N - номери кутів, що покривають відповідні комірки. Кожен куточок характеризується своїм унікальним приміщенням. Кути нумеруються, починаючи з одного, без зазорів. Вирізану клітку слід позначити нулем.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 1 1 | 0 11 1 |
| 2 | 2 2 2 | 2 2 3 32 0 1 34 1 1 54 4 5 5 |

### **Варіант 71**

Кілька людей вирішили поїхати відпочити на природу, подихати свіжим повітрям і т.д. Як це часто буває, деякі з них дружать один з одним, а деякі ні. Щоб нікому не псувати настрій, вони вирішили розділитися на кілька груп. При цьому в кожній групі має бути не більше 5 осіб і вони повинні дружити один з одним.

Знайдіть групування, де найбільший розмір групи найбільший (серед усіх розбивок).

Вводу

Перший рядок вхідного файлу INPUT.TXT містить ціле число n (1 ≤ n ≤ 15) - кількість осіб. Наступні n рядків містять n чисел. Якщо i-й і j-й люди дружать, то j-е число i + 1-й рядок дорівнює 1, в іншому випадку - 0.

Вихід

У першому рядку вихідного файлу.TXT, відображають кількість груп. У другому рядку вихідного файлу виводять n чисел (i-е число - це номер групи, в якій знаходиться i-й людина). Оскільки в будь-якому випадку кількість груп не буде перевищувати n, пронумеруйте групи з цілими числами від 1 до n. Якщо розв'язків кілька, то виведіть будь-які.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 31 1 01 1 00 0 1 | 21 1 2 |
| 2 | 81 1 1 1 1 1 1 11 1 1 1 1 1 1 11 1 1 1 1 1 1 11 1 1 1 1 1 1 11 1 1 1 1 1 1 11 1 1 1 1 1 1 11 1 1 1 1 1 1 11 1 1 1 1 1 1 1 | 31 1 1 1 1 2 2 3 |

### **Варіант 72**

Одним з класичних NP-повних завдань є так зване «Завдання рюкзака». Вона формулюється наступним чином. Наводяться N об'єктів, кожен з яких характеризується вагою wi і корисністю pi. Підбирати набір цих предметів потрібно так, щоб загальна вага цього набору не перевищував W, а загальна корисність була максимальною.

Ваше завдання - написати програму, яка вирішить проблему про рюкзак.

Вводу

Перший рядок вхідного файлу INPUT.TXT містить натуральні числа n (1 ≤ n ≤ 20) і W (1 ≤ W ≤ 109). Кожна з наступних n рядків містить опис одного елемента. Кожен опис складається з двох чисел: wi - вага об'єкта і pi - його корисність (1 ≤ wi, pi ≤ 109).

Вихід

У першому рядку вихідного файлу.TXT, відображають кількість обраних елементів і їх загальну корисність. У другому рядку проведіть їх числа в порядку зростання через пробіл (пункти нумеруються від одиниці в тому порядку, в якому вони перераховані у вхідному файлі).

Якщо ви шукаєте кілька наборів, виберіть той, у якому найменша кількість елементів. Якщо після цього відповідь все ж неоднозначний, виберіть набір, в якому перший пункт має мінімально можливе число, з усіх таких виберіть той, в якому другий пункт має мінімально можливе число, і так далі.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 2 1010 1009 80 | 1 1001 |
| 2 | 5 10080 100050 55050 55050 55050 550 | 2 11002 3 |
| 3 | 6 10080 100050 55050 55050 55050 550100 1100 | 1 11006 |

### **Варіант 73**

У чарівній країні використовуються монети номіналом А1, А2,..., АМ. чарівник прийшов в магазин і виявив, що у нього рівно дві монети кожного номіналу. Йому потрібно заплатити суму Н. Написати програму, яка визначає, чи зможе він платити без змін.

Вводу

У вхідному файлі INPUT.TXT спочатку записується число N (1 ≤ N ≤ 109), потім число M (1 ≤ M ≤ 15), а потім M парами різних чисел A1, A2,..., AM (1 ≤ Ai ≤ 109).

Вихід

У вихідному файлі.TXT, відображають кількість монет, які доведеться віддати чарівній людині, якщо він зможе оплатити зазначену суму без здачі. Якщо рішень кілька, виведіть варіант, при якому чарівник віддасть якомога меншу кількість монет. Якщо без здачі не обійтися, то виведіть один номер 0. Якщо у чарівника не вистачає грошей на виплату зазначеної суми, зніміть один номер -1 (мінус один).

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 5 21 2 | 3 |
| 2 | 7 21 2 | -1 |
| 3 | 5 23 4 | 0 |

### **Варіант 74**

Знайдіть кількість N-значних чисел, в яких сума цифр дорівнює їх добутку. Вивести найменше таке число для даного N.

Вводу

Один рядок у вхідному файлі INPUT.TXT містить одне натуральне число N, яке не перевищує 20.

Вихід

В одному рядку вихідного файлу .TXT повинні відображатися через пробіл 2 числа: кількість шуканих чисел і найменше серед них.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | 1 | 10 0 |
| 2 | 2 | 1 22 |

### **Варіант 75**

У кожній мові програмування, навіть найпростішому, є оператор розгалуження, що дозволяє перевірити істинність логічного виразу і в залежності від його результату виконати дію. Умова оператора гілки — це логічний вираз, який може призвести як до істинного, так і до хибного. Змінні, які можуть брати участь у логічному виразі, називаються булевих змінними. Булеві змінні можуть бути об'єднані в складні умови за допомогою логічних операцій (функцій):

AND(x1,x2,.. xs). Операція «I» повертає істинне значення, якщо всі її операнди істинні. (2 ≤ кількість операндів ≤ с);

OR(x1,x2,.. xs). Операція OR повертає значення істинності, якщо принаймні один з її операндів істинний. (2 ≤ кількість операндів ≤ с);

НЕ(х1). Операція «НЕ», змінює значення операнда х1 на протилежне (операнд завжди один).

У логічному стані можуть використовуватися кілька логічних функцій, вкладених один в одного, тобто результат однієї функції може бути використаний іншою в якості операнда. Наприклад, AND(A, B, OR(C, D)). Цей вираз буде вірним, коли вірні A, B і (C або D).

Потрібно написати програму, яка буде визначати результат виразу з доступного логічного виразу і значення логічних змінних. Кількість операндів для функцій AND і OR завжди дорівнює двом (s=2).

Вводу

Перший рядок вхідного файлу INPUT.TXT містить логічний вираз (довжиною не більше 255 символів). Другий рядок містить два числа, розділених одним або декількома пробілами: N - кількість блоків (не більше 10), K - кількість змінних (не більше 26). За N блоками кожен має наступну структуру: складається з K рядків, кожен містить вираз типу <зміна = значення>. Змінні задаються великими англійськими літерами, значення - з константами TRUE або FALSE (великі літери).

Вихід

У файлі виводу OUTPUT.TXT виведіть N рядків зі значенням результату логічного виразу для змінних відповідного блоку.

Приклади

|  |  |  |
| --- | --- | --- |
| Ні | ВХІДНІ ДАНІ.TXT | ВИХІД.TXT |
| 1 | І(A,NOT(B))1 2A=FALSEB=TRUE | ДЕТАЛЬНІШЕ |
| 2 | OR(NOT(AND(A,B)),A)3 2A=FALSEB=TRUEA=TRUEB=TRUEA=FALSEB=FALSE | TRUETRUE |