# Лабораторна робота №1

**Варіант 1**

Як уже безперечно стало відомо всім, Запоріжжя домігся права на проведення Літніх Олімпійських ігор 2032 року. Планується, що Україні як країні-організатору Олімпійських ігор буде дозволено внести мінімальні зміни до програми Олімпіади. Так, з метою покращення загального командного результату, було вирішено замінити змагання з плавання першістю у абсолютно новій грі "Ґудзики".

Правила гри дуже прості. Перед двома граючими знаходиться купка з K гудзиків. Гравці по черзі беруть гудзики з купки, причому за один хід кожен з них може взяти від 1 до гудзиків. Виграє той зі спортсменів, якому вдасться взяти останній гудзик.

Правила олімпійських змагань будуть лише трохи складнішими за звичайні. Той із гравців, якому за жеребом випадає робити перший хід, отримує можливість власноруч призначити число K, керуючись у своєму виборі лише обмеженнями 3 <= K <= 100 000 000 (саме стільки ґудзиків заготовлено для олімпійського турніру). Той із гравців, який ходитиме другим, вибирає, у свою чергу, число L, яке має відповідати умові 2 <= L < K.

Завдання

На вашу команду покладається дуже відповідальне завдання: необхідно написати програму, яка б допомагала другому гравцю робити свій вибір. Іншими словами, за заданим числом ґудзиків у купці K, необхідно визначити таке число L, яке гарантує перемогу другому гравцю при найкращій грі обох сторін.

Так, наприклад, якщо в купці всього три ґудзики, то перемогу другому гравцю забезпечує вибір L=2. Справді, якщо перший гравець своїм ходом забере один гудзик, то другий зможе виграти, взявши обидва ґудзики, що залишилися, і, навпаки, якщо перший візьме два гудзики, то другий переможе, взявши останній. Початкові дані

Вхід для цього завдання складається з одного рядка, в якому записано однину число K - кількість гудзиків у купці, обрана першим гравцем.

Результат

На вихід слід записати єдине ціле число L - максимальну кількість гудзиків, яке можна взяти за один хід, що забезпечує перемогу другому гравцю. Якщо таких чисел кілька, слід вивести найменше їх. Якщо таких чисел немає, слід вивести число 0.

Приклад вихідних даних 3

Приклад результату 2

# Варіант 2

В одній із острівних держав Карибського басейну всі рішення традиційно приймалися простою більшістю голосів на загальних зборах громадян, яких, на щастя, було не дуже багато. Одна з місцевих партій, прагнучи прийти до влади якомога законнішим шляхом, змогла домогтися деякої реформи виборчої системи. Головним аргументом було те, що населення острова останнім часом значно зросло і проведення загальних зборів перестало бути легким завданням.

Суть реформи полягала в наступному: з моменту введення її в дію всі виборці острова поділялися на K груп (необов'язково рівних за чисельністю). Голосування з будь-якого питання тепер слід проводити окремо в кожній групі, причому вважалося, що група висловлюється "за", якщо "за" голосує більше половини людей у ​​цій групі, а в іншому випадку вважалося, що група висловлюється "проти". Після

Проведення голосування в групах підраховувалося кількість груп, що висловилися "за" і "проти", і вирішення питання вирішувалося позитивно в тому і лише тому випадку, коли груп, що висловилися "за", виявлялося більше половини загальної кількості груп.

Ця система спочатку була з радістю прийнята мешканцями острова. Коли перші захоплення розвіялися, очевидними стали, однак, деякі недоліки нової системи. Виявилося, що прихильники партії, яка запропонувала систему, змогли вплинути на формування груп виборців. Завдяки цьому вони отримали можливість проводити деякі рішення, не володіючи при цьому реальною більшістю голосів!

Нехай, наприклад, на острові було сформовано три групи виборців, чисельністю 5, 5 та 7 осіб відповідно. Тоді партії достатньо мати трьох прихильників у кожній із перших двох груп, і вона зможе провести рішення лише 6-ма голосами "за", замість 9-ти, необхідних при загальному голосуванні.

Вам треба написати програму, яка визначає за заданим розбиттям виборців на групи мінімальну кількість прихильників партії, достатню, за деякого розподілу їх за групами, для прийняття будь-якого рішення.

Введення.

Вхідний файл для цього завдання складається із двох рядків. У першому рядку файлу записано натуральне число K<101 – кількість груп виборців. У другому рядку через пропуск записані K натуральних чисел, які задають кількість виборців у групах. Для спрощення визначення поняття "більшість голосів" будемо вважати, що і кількість груп, і кількість виборців у кожній групі є непарними числами. Ви також можете вважати, що населення острова не перевищує 10001 чоловік.

Висновок.

У вихідний файл слід записати однину натуральне число - мінімальна кількість прихильників партії, здатне вирішити результат голосування.

Приклад файлу input.txt. 3

5 7 5

Приклад файлу output.txt. 6

# Варіант 3.

Прямокутники, площі яких рівні, називаються рівновеликими. Написати програму, що містить усі можливі цілочисельні сторони рівновеликих прямокутників заданої площі. Однакові прямокутники, що виходять заміною сторін, друкувати не треба.

Вхідні дані:

Площа чотирикутника (1<=n<=50000). Вихідні дані:

Сторони чотирикутника (A\*B) Приклад вхідних даних:

12

Приклад вихідних даних:

1\*12 2\*6 3\*4

# Варіант 4

Є N банок з цілими обсягами V1, ..., Vn літрів, порожня посудина і кран з водою. Чи можна за допомогою цих банок налити в посудину рівно V літрів води?

Вихідна інформація міститься у файлі input.txt у першому рядку - кількість банок N<100, потім N рядків з об'ємами банок і в останньому рядку об'єм судини.

Програма має вивести у файл output.txt одне із двох слів YES! чи NO.

# Варіант 5

Є монети з різними фіксованими номіналами, вираженими у копійках (наприклад, 3 та 5 копійок) у достатній кількості. Написати програму COINS.\*, яка:

а) визначає, чи можна уявити задану суму S (виражену в копійках), користуючись монетами заданих номіналів,

б) якщо це можливо, то становить цю суму за допомогою мінімальної кількості монет.

Вхідний файл COINS.DAT містить у першому рядку суму S (0 < S < 1000000000), у другому рядку - N - кількість різних номіналів (0 < S < 20), а в наступних N рядках - А1 ... АN - номінали (у порядку зростання), які можна використовувати (0 < A1 < A2 < ... < AN < 1000000000).

Вихідні дані: Вихідний файл COINS.SOL повинен містити в першому рядку знак "+", якщо задану суму S можна подати, і знак "-", якщо не можна. Якщо подана сума існує, то наступні N рядків повинні містити кількість монет кожного номіналу, які необхідні для подання суми S за допомогою мінімальної кількості монет.

Приклади введення та виведення

Представити 514 копійок за допомогою монет номіналом у 3 та 5 копійок COINS.DAT:

514

2

3

5

COINS.SOL:

+ 3

101

Представити 27 копійок за допомогою монет номіналом у 5 та 13 копійок COINS.DAT:

27

2

5

13

COINS.SOL:

-

# Варіант 6

У дитинстві у мене була розвага - кидатися помідорами з балкона, так щоб забризкати помідор перехожих, що прогнили нутрощами. Я тоді помітив, що на кожен кубічний сантиметр помідори (до речі, вона має форму ідеальної кулі) припадає квадратний метр поверхні (це пояснюється тим, що помідор падає з 7-го поверху і розмазується великою площею дрібними бризками). Тобто. помідора об'ємом 3 куб. див. забризкає коло з центром у точці падіння площею 3 кв. м.

Я намітив кілька точок, у які точно потраплю. В яку з цих точок треба кинути найменшого помідору, щоб забризкати всіх перехожих?

Вхідні дані:

У першому рядку вхідного файлу input.txt міститься число n – кількість намічених точок, у наступних n рядках – координати цих точок з точністю до 2 знаків після коми. У наступному рядку міститься m - кількість осіб, у наступних m

рядках – координати людей. 1 <= m, n <= 100. Координати перебувають у проміжку від - 100 до 100.

Вихідні дані:

У першому рядку містяться координати точки, в яку треба кидати помідору, а у другому - найменший радіус r помідори в сантиметрах.

Приклад вхідних даних:

1

0 0

2

1 0

0.5 0.5

Приклад вихідних даних:

0.00 0.00

0.91

# Варіант 7

У моїй групі навчається один хлопець - Аріф, який часто займається на уроках математики взагалі то математикою, але не зовсім тією, якою треба: то він виписує в стовпчик квадрати всіх чисел, то всі ступеня двійки, то таблицю множення до 100\*100, коротше кажучи, наш Аріф - ще той. Нещодавно отримав 2 за контрольну, бо на всіх попередніх уроках займався далеко не тим, що треба. Одного разу він вигадав собі нову розвагу - став виписувати в рядок (без прогалин) четверті ступені всіх простих чисел поспіль. У нього вийшов приблизно такий рядок:

16816252401...

Потім йому стало цікаво чи можна визначити, яка цифра стоїть на n-му місці в цьому рядку? Формулу підібрати він не зміг, та напевно її й нема. Завдання полягає в тому, що незважаючи на відсутність формули, визначити, яка цифра стоїть на n-му місці.

Вхідні дані:

У вхідному файлі input.txt міститься однина n (0 < n < 25000). Вихідні дані:

У вихідному файлі output.txt міститься одна цифра, яка стоїть у цьому рядку на n-му місці.

Приклад 1:

input.txt:

1

output.txt:

1

Приклад 2:

input.txt 5

output.txt 6

# Варіант 8

Жителі однієї держави дуже люблять різні математичні головоломки. Навіть той, хто хоче отримати в'їзну візу, повинен вирішити завдання: знайти ключове слово. Умова завдання така:

На аркуші написано кілька довгих чисел. Якщо скласти всі цифри у кожному числі, вийдуть нові цифри. Далі, слід скласти всі цифри у кожному із знову отриманих чисел. Процес слід продовжувати доти, доки в результаті не залишаться числа, менші за 10. Після цього все просто: числа від 0 до 9 - це номери літер в алфавіті (у цьому

державі алфавіт складається з десяти літер). Заміна чисел буквами та дає ключове слово.

Напишіть програму, яка шукатиме ключове слово. Формат вхідного файлу:

Перший рядок - алфавіт держави: десять літер, розташованих за зростанням порядкових номерів без пробілів.

Другий рядок - кількість чисел (N<=255)

Наступні N рядків - власне вихідні числа (по одному рядку, у кожному трохи більше 255 цифр).

Формат вихідного файлу:

Ключовеслово Приклад:

input.txt АГЕІКЛМОРТ 4

8267

19929

54262

000000000000

output.txt ЛІГА

# Варіант 9

Написати програму, яка визначає мінімально можливу кількість гравців у команді КВК, якщо відомо, що дівчат у команді більше X%, але менше Y%.

Формат вхідних даних

У першому рядку вхідного файлу записані числа X та Y, розділені пропуском. X та Y - цілі, 1 <= X,Y <= 100.

Формат вихідних даних

Потрібно вивести одне число – мінімальна кількість членів команди за обмежень, заданих у вхідному файлі.

Приклад введення 40 50

Правильний висновок для прикладу 7

# Варіант 10.

Маршрут руху автомобіля заданий у вигляді координат вершин ламаної.

Необхідно визначити кількість лівих поворотів (суміжні ділянки ламаної не лежать на одній прямій). Автомобіль починає рух з будь-якої точки (! Тут уточнення від мене - автомобіль починає рух з першої вершини ламаної, а ось її координати будь-які!).

Формат вхідних даних:

Перший рядок вхідного файлу input.txt складається з одного числа, кількості ланок ламаної; у наступних рядках – пари натуральних чисел, координати вершин ламаної.

Формат вихідних даних:

Вихідний файл output.txt містить одне число – кількість лівих поворотів input.txt:

4

1 1

2 2

3 2

3 3

2 3

output.txt:

2

# Варіант 11.

На кільцевому маршруті №54 протяжністю S, що проходить повз пансіонат "Енергетик", працює N автобусів. Автобуси пронумеровані числами від 1 до N у порядку їхнього прямування за маршрутом. Автобус з номером 1 рухається за автобусом з номером N. Розклад складено таким чином, що автобуси рухаються з однаковою швидкістю V0 та рівними інтервалами між ними. Рух автобусів контролює диспетчер.

О 12 годині дня деякі K автобусів одночасно знімаються з маршруту і вирушають на обід. Для відновлення рівності інтервалів між автобусами, що продовжують рух за маршрутом, знадобиться деякий час Т і, можливо, зміна швидкості деяких автобусів за командою диспетчера. Протягом цього часу автобуси повинні рухатися з постійними швидкостями з інтервалу [Vmin, Vmax], призначеними диспетчером. Зміна швидкості руху автобуса відбувається миттєво. Після закінчення часу Т автобуси відновлюють рух маршрутом зі швидкістю V0.

Потрібно написати програму для автоматичного диспетчера, яка обчислює мінімальний час Tmin, за який інтервали руху між автобусами, що залишилися, стануть рівними, і швидкості руху кожного з них протягом цього часу.

Вхідні дані

Вхідний файл bus.in містить два рядки.

У першому рядку знаходяться натуральні числа N, К, S, Vmin, Vmax і V0, де K<N<=10000, S<=10000, Vmin<Vmax<=10000, Vmin<=V0<=Vmax.

У другому рядку розташовані у порядку зростання K чисел – номери автобусів, знятих з маршруту. Усі дані у рядках розділені пробілами.

Вихідні дані

У першому рядку вихідного файлу bus.out має бути значення Tmin.

У кожному з наступних N - K рядків повинні бути по два розділені пропуском числа - номер автобуса на маршруті і швидкість його руху протягом часу Tmin. Номери автобусів упорядкувати за зростанням. Значення Tmin та швидкостей виводити з точністю до 4-х значущих цифр після десяткової точки.

Приклад 1

Вхідний файл bus.in 4 1 60 21 70 60

3

Вихідний файл bus.out 0.2041

1 45.5

2 21

4 70

Приклад 2

Вхідний файл bus.in 4 2 40 30 80 50

2 4

Вихідний файл bus.out 0

1 50

3 50

# Варіант 12.

У деякому царстві, у деякій державі було N міст, і всі вони, судячи з головної карти імператора, мали цілі координати. У роки лісу були дрімучі, дороги ж будувати вміли лише паралельно осям координат, отже відстань між двома містами визначалося як |x1 - x2| + | y1 - y2 |.

Імператор вирішив побудувати n+1 місто і зробити його столицею своєї держави, при цьому координати столиці також повинні бути цілими. Місце для столиці слід вибрати так, щоб середня арифметична відстань між столицею та іншими містами була якнайменша. Проте, зрозуміло, столицю не можна будувати дома існуючого міста.

Нелегке завдання вибрати місце для столиці доручено Вам. Формат вхідних даних

Перший рядок вхідного файлу містить число N - кількість міст (1<=N<=100). Наступні N рядків містять координати міст - пари цілих чисел, що не перевищують 1000 по абсолютній величині.

Формат вихідних даних

Виведіть у вихідний файл два цілі числа - координати точки, де слід побудувати столицю. Якщо рішень кілька, виведіть будь-яке.

Приклади capital.in 8

0 0

1 0

2 0

0 1

2 1

0 2

1 2

2 2

capital.out 1 1

capital.in 4

0 0

1 1

0 1

1 0

capital.out 0 2

# Варіант 13.

Не секрет, що багато жителів Лощини Бурштинової Роси часто звертаються до ченців Монастиря з різними питаннями. Щовесни жителі навколишніх сіл запитують Настоятеля, скільки мішків зерна вони зберуть восени з одного посадженого навесні мішка. Настоятелю набридло щоразу винаходити відповідь на це питання, і він вирішив придумати алгоритм, який би генерував відповідь (не подумайте, що він був просто шарлатаном, але ж не можна турбувати Будду з таких дрібниць). Він вирішив, що відповідатиме за таким правилом (таке вже хитре правило ніхто не розкусить):

Факторіалом числа n, n - натуральне, Настоятель називає число, що визначається за правилом: n! = 1, якщо n = 1

n! = n \* (n - 1)!, Якщо n > 1

Функція F(x), x >= 0 обчислюється за таким правилом:

F(x) = x, якщо x <= 9 F(x) = F(S(x)), якщо x > 9

де S(x) - сума цифр у десятковому записі числа x.

Правило це Настоятель застосовує так: у рік із номером n від народження Будди мудрий Настоятель повідомляє, що вони зберуть F(n!) мішків зерна з одного посадженого мішка навесні. З кожним роком все складніше і складніше Настоятелю обчислювати F(n!), ваша мета - допомогти йому в цьому.

Введення. Єдине число n – номер року, 1 <= n < 200 Висновок. F(n!) - відповідь Наставника

Приклад input.txt #1

3

Приклад output.txt #1

6

Приклад input.txt #2

4

Приклад output.txt #2

6

# Варіант 14.

Легендарний учитель математики Юрій Петрович вигадав кумедну гру з числами. А саме, взявши довільне ціле число, він переводить його в двійкову систему числення, отримуючи деяку послідовність із нулів та одиниць, що починається з одиниці. (Наприклад, десяткове число 1910 = 1\*24+0\*23+0\*22+1\*21+1\*20 у двійковій системі запишеться як 100112.) Потім вчитель починає зрушувати цифри отриманого двійкового числа за циклом (так, що остання цифра стає першою, проте інші зрушуються однією позицію вправо), виписуючи які утворюються у своїй послідовності з нулів і одиниць в стовпчик - він зауважив, що незалежно від вибору вихідного числа послідовності починають з деякого моменту повторюватися. І, нарешті, Юрій Петрович шукає максимальне з виписаних чисел і переводить його назад у десяткову систему числення, вважаючи це число результатом виконаних маніпуляцій. Так, для числа 19 список послідовностей буде таким:

10011

11001

11100

01110

00111

10011

і результатом гри, отже, виявиться число 1 \* 24 + 1 \* 23 + 1 \* 22 + 0 \* 21 + 0 \* 20 = 28. Оскільки придумана гра з числами все більше займає уяву вчителя, відволікаючи тим самим його від роботи з ну дуже обдарованими школярами, Вас просять написати програму, яка допомогла б Юрію Петровичу отримувати результат гри без стомлюючих ручних обчислень.

Формат вхідних даних

Вхідний файл містить одне ціле число N (0 <= N <= 32767). Формат вихідних даних

Ваша програма повинна вивести у вихідний файл одне ціле число, що дорівнює результату гри.

Приклад d.in

19

d.out 28

# Варіант 15.

Вхідний файл містить набір слів (словом вважається послідовність символів, що не містить символу перекладу рядка). 1-а літера може бути як великої, і малої, інші - малі. Нехай деяка послідовність літер зустрічається обох словах (регістр літер враховується).

Відсотком збігу назвемо максимальну довжину такої послідовності, поділену на довжину більш короткого з цих слів, помножену на 100% і заокруглене до найближчого цілого. Потрібно розподілити слова по парах, щоб сумарний відсоток збігу всім пар був найбільшим.

Формат вхідного файлу:

N

слово1

...

словоN

де N – число слів від 2 до 12 (завжди парне). Довжина кожного слова не перевищує 20 символів.

Формат вихідного файлу:

слово слово процент\_збігу

...

слово слово процент\_збігу де:

у кожній парі слова стоять за абеткою.

пари відсортовані в алфавітному порядку за першими словами в них. Приклад вхідного файлу:

8

Оля Іра Катя Наташа Сергій Ілля Коля Сашко

Файл результату для цього прикладу: Ілля Оля 67

Іра Сергія 67

Катя Коля 50

Наталя Саша 75