

## ЗАДАЧІ X ВСЕУКРАЇНСЬКОГО ТУРНІРУ ЮНИХ ІНФОРМАТИКІВ

**Мотурнак Є. В.**

У місті Херсоні з 6 по 10 квітня 2012 року відбувся X ювілейний Всеукраїнський турнір юних інформатиків. За час свого існування турнір постійно розширює свою географію і в цьому році в турнірі брали участь десять команд з восьми регіонів України: м. Києва, Дніпропетровської, Донецької, Харківської, Запорізької, Кіровоградської, Херсонської, Волинської областей.

У правилах проведення Всеукраїнського турніру юних інформатиків зазначається, що ТЮІ є командним змаганням школярів, що доводить їх здатність розв'язувати складні наукові і дослідницькі завдання, подавати свої розв'язки у доведеній до впровадження формі, захищати їх під час наукової дискусії.

Серед інших змагань з інформатики ТЮІ виокремлюється своїм форматом. З одного боку, завдання турніру є комплексними, такими що охоплюють широкий спектр інформаційних технологій, які можуть бути використаними під час їх виконання, з іншого, робиться акцент на створенні завершеного програмного продукту, готового до реального використання.

Команди, що складаються з 3–5 учнів беруть участь у трьох турах. Перший тур проходить у такому ж форматі, як більшість інших предметних турнірів. У ході дискусійних «боїв» учасники відстоюють право своїх програмних продуктів, створених заочно, називатися кращими. Водночас, кожна команда виступає в трьох ролях: доповідача, опонента та рецензента.

У другому турі команда отримує завдання на створення програмного продукту. Вона має створити його за п'ять годин, працюючи з двома комп'ютерами, що з'єднані мережею. Після закінчення роботи результати демонструються учасниками в ході публічного захисту.

Третій тур — це АСМ-олімпіада. Маючи один комп'ютер на команду, учні повинні розв'язати сім задач різного рівня складності. Особливістю АСМ-олімпіади є те, що кожна задача повинна бути розв'язана повністю, тобто проходити всі, без виключення, тести журі. У такому випадку, часткові розв'язки не зараховуються. Крім того, фіксується час, коли задачу було зараховано, за кожно невдалу спробу розв'язку нараховується додатковий штрафний час. Перемагає команда, що має найбільшу кількість зарахованих задач, а у разі співпадання цієї кількості в декількох команд, йде ранжування за часом здачі.

Розглянемо завдання кожного туру докладніше.

### ЗАВДАННЯ І ТУРУ

Завдання першого туру публікується заздалегідь, за 3–4 місяці до дати початку турніру. Основою завдання є створення достатньо складної математичної моделі з визначеної предметної області і її реалізація у вигляді програмного продукту. Як правило, розглядаються проблеми реального життя. Так, за останні чотири роки було розглянуто такі предметні галузі як метеорологія, космічні польоти, гідрологічні процеси формування рельєфу, динаміка біологічних популяцій.

Проблеми, які розглядаються в завданнях є відкритими, такими, що не мають абсолютно вірних, класичних розв'язків. Будь-яка модель є лише наближенням до реальності з більшою чи меншою точністю. Тож, будь-який розв'язок має право на існування, і одним із важливих умінь команди є вміння довести правильність своєї точки зору, адекватність створеної моделі процесам, що реально відбуваються.

Під час виконання заочного завдання, учні можуть (і мають) консультуватися з фахівцями у відповідних предметних галузях, отримувати допомогу з боку вчителів, але не повинні забувати, що захищати свій продукт, відповідати на доволі складні запитання їм доведеться самим.

### Завдання №1 (модель «хижак–жертва»)

1.1. Розробити детерміновану математичну модель динаміки двох популяцій — «хижаків» і «жертв», яка б дозволила відстежувати зміну їх кількості у часі. Обов'язкові вхідні параметри моделі: кількість «хижаків» і «жертв» у початковий момент часу, вихідні — їх кількість на кінець досліджуваного періоду. Обґрунтувати вибір параметрів і виведення формул. Модель необхідно представити у текстовому файлі з поясненнями змінних і виведених закономірностей.

1.2. Реалізувати математичну модель у вигляді програми. Вхідні дані: кількість «хижаків» і «жертв» на початку досліджуваного періоду, кількість модельованих періодів часу (поколінь). Вихідні дані: кількість «хижаків» і «жертв» у кожному з досліджуваних періодів.

1.3. Розробити імітаційну математичну модель динаміки двох популяцій — «хижаків» і «жертв» з урахуванням випадкових факторів. Обов'язкові вхідні параметри моделі: кількість «хижаків» і «жертв» у початковий момент часу, вихідні — їх кількість на кінець досліджуваного періоду. Обґрунтувати вибір параметрів і виведення формул. Модель необхідно представити у текстовому файлі з поясненнями змінних, формул та правил поведінки істот. Файл повинен містити опис алгоритму розрахунків за моделлю.

1.4. Реалізувати імітаційну модель у вигляді програми. Вхідні дані: кількість «хижаків» і «жертв» на початку досліджуваного періоду, розташування істот у навколишньому середовищі, кількість модельованих періодів часу (поколінь). Вихідні дані: кількість хижаків і жертв у кожному з досліджуваних періодів. Програма повинна давати можливість встановлення значень параметрів випадкових факторів перед запуском.

1.5. Реалізувати можливість автоматичного проведення серії експериментів з одними й тими самими наборами значень вхідних даних.

### Завдання №2 (візуалізація)

2.1. Додати у програму можливість спостереження за динамікою розвитку популяції у вигляді графіків кількості хижаків і жертв у часі й фазового портрету системи.

2.2. Візуалізувати імітаційну модель динаміки двох популяцій s «хижаків» і «жертв» s у вигляді двовимір-



ної паралельної/ізометричної проекції середовища, де живуть організми.

2.3. Візуалізувати імітаційну модель зміни динаміки двох популяцій  $s$  «хижаків» і «жертв»  $s$  у тривимірному вигляді з можливістю зміни ракурсу.

2.4. Додати можливість масштабування, уповільнення роботи, паузи та перезапуску моделі.

#### Завдання №3 (навколишнє середовище)

3.1. Додати до моделі можливість зміни початкових значень параметрів навколишнього середовища.

3.2. Врахувати можливість неоднорідності навколишнього середовища.

3.3. Додати до моделі можливість зміни параметрів навколишнього середовища з часом, обумовлені процесами в екосистемі і зовнішніми чинниками.

3.4. Врахувати вичерпність ресурсів навколишнього середовища і швидкість їх відновлення.

3.5. Врахувати шкідливий вплив екосистеми на навколишнє середовище.

3.6. Врахувати можливість зовнішнього забруднення навколишнього середовища і вплив його на екосистему.

3.7. Реалізувати всі зміни моделі у програмі.

#### Завдання №4 (псевдогенетичний код)

4.1. Додати до моделі можливість зміни початкових параметрів істот у вигляді псевдогенетичного коду. Псевдогенетичний код складається з генів, кожен з яких кодує морфологічну особливість дорослої істоти, аспект поведінки тощо.

4.2. Додати можливість співіснування в екосистемі більше, ніж двох видів.

4.3. Додати можливість установки паузи в моделі, зміни генетичного коду істот, і продовження роботи.

4.4. Реалізувати у системі візуалізації відображення графіків зміни у часі параметрів популяцій всіх видів.

#### Завдання №5 (еволюція)

Розробити модель, що враховує можливість мутацій і еволюції видів.

5.1. Додати до моделі механізм мутацій псевдогенетичного коду протягом життєвого циклу істот, що буде переданий наступним поколінням.

5.2. Додати до моделі можливість обміну генетичним матеріалом між істотами генетично-близьких видів.

5.3. Реалізувати оптимальну систему візуалізації динаміки розвитку всіх істот з можливістю відстеження зміни генетичних кодів кожної істоти і популяцій генетично.

#### Додаток 1. Визначення

Детермінована математична модель — модель, результати якої визначені через відомі відношення станів і подій, і в якій заданий набір вхідних даних буде завжди визначати один і той самий результат (наприклад, модель, що зображує відому хімічну реакцію).

Імітаційна математична модель — модель процесу, що описує, як цей процес проходив би насправді. Таку модель можна «програти» в часі як для одного випробування, так і заданої їх кількості. При цьому результати визначатимуться випадковим характером процесів. За цими даними можна отримати достатньо стійку статистику.

У вузькому значенні, імітаційна модель — це логіко-математичний опис об'єкта, який може бути використаний для експериментування на комп'ютері в цілях проектування, аналізу й оцінки функціонування об'єкта.

Фазовий простір — багатовимірний простір змінних динамічної системи. Кількість вимірів цього простору відповідає кількості відстежуваних змінних у системі. Наприклад, для моделі «хижак-жертва» фазовий простір може мати два виміри — кількість хи-

жаків і кількість жертв. Тобто фазовий простір у цьому випадку — це координатна площина, на осі абсцис якої відстежується кількість хижаків, а на осі ординат — жертв. Точка в такому просторі відповідає стану системи у певний момент часу. Наприклад, точка з координатами (5;16) вказує на те, що у момент часу, відповідний цій точці, у системі 5 хижаків і 16 жертв. Час у фазовому просторі представлений неявно.

Фазовий портрет системи — зображення траєкторій динамічної системи у фазовому просторі. Кожен стан системи відповідає певній точці на фазовому портреті. Фазові портрети служать для наочного відображення особливостей еволюції динамічної системи: стаціонарних точок, циклів, басейнів притягання. У прикладі з моделлю «хижак — жертва» фазовий портрет — це послідовне відображення точок, відповідних стану системи, у фазовому просторі. Точки можуть бути поєднані лініями або ні, залежно від цілей дослідника.

Завдання були побудовані за принципом «від простого до складного». Перше завдання було доступно всім командам, незалежно від мови програмування, достатньо було лише вміти робити обчислення і працювати з файлами. Друге завдання вже потребувало роботи з технологіями візуалізації, не додаючи математичної складності до моделі. Третє — розширення моделі за рахунок відстеження навколишнього середовища. Основну складність (і кінцеву мету) всього завдання було поміщено в четверте і п'яте завдання.

У фінальному вигляді, за умови виконання всіх завдань, програмний продукт повинен був представляти із себе систему, що дозволяла б досліджувати задані екосистеми. Водночас, екосистема повинна була б складатися з певної кількості видів. Кожен із видів характеризувався б псевдогенетичним кодом, що задавав би як морфологічні особливості істот, так і стратегії їх поведінки. За рахунок механізму обміну генами під час розмноження, мутацій, можливо було б простежити.

Під час виконання цього завдання виникало дві окремі складні задачі. Це розробка і документування математичної моделі, з одного боку, і реалізація цієї моделі у вигляді програмного продукту, з іншої. Насправді, друга задача викликала набагато менше складнощів, у роботах було використано різні технології 2D та 3D-візуалізації, створено досить складні інтерфейси.

На жаль, далеко не в усіх роботах була присутня докладна документація з описом параметрів моделей, формул, що їх пов'язують. Над підготовкою проекту для I туру працює команда. Зазвичай, безпосереднім програмуванням займається один чи два учні, а інші як раз можуть зайнятися підготовкою текстової документації, опису моделей, створенню інструкцій користувача. Крім того, для виконання таких складних завдань необхідно чітко спланувати свою роботу.

Учні часто намагаються уникнути етапу створення математичної моделі, змішуючи його з безпосереднім програмуванням. Це, безперечно, недалекоглядно. Уявимо ситуацію. У програмі вже написано кількості або й тисячі рядків коду, аж тут, програміст розуміє, що все працює не так, як дано, треба вносити корективи у модель. А корективи в моделі можуть потребувати зміни 80% коду. А якщо таких змін буде не одна? Тому спочатку розробляється мо-

дель, кілька разів перевіряється, і тільки після цього слід приступати до програмування.

Для можливості проведення повного аналізу, наведемо критерії, за якими оцінювались роботи учасників за досконалість й ефективність роботи програмного продукту (табл. 1).

Нагадаємо, що оцінка за перший тур складається з оцінки за роботу й оцінки за бої. Критерії оцінювання доповіді, опонування і рецензування наведено у правилах проведення ТЮІ: професійність, чіткість, зрозумілість, логічність, коректність.

Як показує практика проведення турнірів, якісний захист, опонування та рецензування можуть висвітлити сильні сторони власної програми і помилки суперників. Але буває і навпаки. Тож, під час підготовки до першого туру слід приділяти велику увагу технології публічного виступу і ведення дискусії.

**ЗАВДАННЯ II ТУРУ**

Завдання другого туру також полягає у створенні завершеного програмного продукту. Але, на відміну від першого туру, виконання завдання є очним. Тобто, за п'ять годин необхідно створити програму, що відповідає умовам завдання. При цьому перевіряється, як учні володіють програмними засобами за відсутності зовнішніх консультацій.

Завдання може формулюватися у загальному вигляді й носити дослідницький характер. Так у попередні два роки учням пропонувалося дослідити і створити власні версії відомих у минулому ігрових програм Paratroopers і Toppler (Перебудова).

Цього року вся країна живе в очікуванні Євро-2012, тож і ми не лишилися осторонь і запропонували учням створити комп'ютерну версію гри «Футбол на аркуші».

**Футбол на аркуші**

Футбол на аркуші паперу — дуже популярна гра. У неї грають і школярі, і студенти.

Для гри потрібний аркуш паперу в клітинку і ручка. На аркуші, відступивши від країв одну клітинку, малюється рамка — ігрове поле 32x40 клітинок. По центру коротких сторін аркуша вертикальними лініями малюються ворота (рис. 1) шириною 4 клітинки. По центру ігрового поля ставиться крапка — початок гри.

Двоє гравців роблять удари по черзі різними кольорами. Перший гравець робить удар м'ячем — малює лінію з 3 відрізків, які послідовно проходять через сусідні вузли сітки. Лінія може бути прямою або ламаною. Відрізки можуть проходити по межах або по діагоналі клітинок (рис. 2).

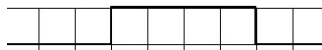


Рис. 1

Перетинати і торкатися ліній попередніх ударів і бічної лінії поля не можна. Якщо після удару першого гравця другому гравцеві нікуди бити (тобто, він не може побудувати лінію з трьох відрізків, що відповідає правилам), то перший гравець отримує право на штрафний удар.

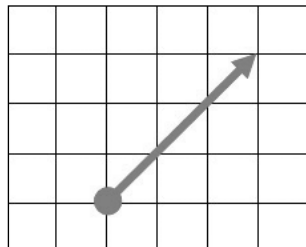


Рис. 2

На наступному прикладі (рис. 3) «червоний» гравець (1) виконує удар (хід) так, що «синьому» гравцеві (2) нікуди бити.

Штрафний удар можна виконувати в будь-якому напрямі. Штрафний удар — лінія з 6 відрізків може бу-

Критерії	
Завдання 1	
1.1	Відповідність опису детермінованої матмоделі вимогам
	Опис вхідних і вихідних параметрів детерм. матмоделі
	Опис формул, що пов'язують параметри моделі
	Обґрунтування детермінованої моделі
1.2	Програмна реалізація відповідає описаній моделі
	Відповідність введення і виведення даних програмою завданню
1.3	Відповідність опису імітаційної матмоделі вимогам
	Опис вхідних і вихідних параметрів імітаційної матмоделі
	Опис формул, що пов'язують параметри моделі
	Врахування випадкових факторів
	Опис правил поведінки істот
	Опис алгоритму розрахунків за моделлю
	Обґрунтування моделі
1.4	Програмна реалізація відповідає описаній моделі
	Відповідність введення і виведення даних програмою завданню
	Керовані параметри випадкових факторів
1.5	Реалізована можливість проведення серій експериментів
	Реалізована можливість збереження результатів експериментів
Завдання 2	
2.1	Візуалізація динаміки популяцій у вигляді графіків
	Візуалізація фазового портрету системи
2.2	Двовимірна візуалізація моделі
2.3	Тривимірна візуалізація моделі
	Можливість зміни ракурсу
2.4	Масштабування зображень
	Наявність режиму паузи
	Уповільнення/прискорення імітації
	Перезапуск моделі з поточними значеннями параметрів
Завдання 3	
3.1	Можливість зміни параметрів навколишнього середовища
	Обґрунтування параметрів навколишнього середовища
3.2	Неоднорідності навколишнього середовища
3.3	Реалізовано зміну навколишнього середовища з часом
3.4	Враховано вичерпність ресурсів навколишнього середовища
3.5	Враховано шкідливий вплив екосистеми на навколишнє середовище
3.6	Враховано можливість зовнішнього забруднення
Завдання 4	
4.1	Моделюється псевдогенетичний код
	Обґрунтування набору "генів"
4.2	Реалізовано можливість співіснування багатьох видів
4.3	Реалізовано можливість зміни псевдогенетичного коду користувачем
4.4	Візуалізація у вигляді графіків відображає всі види в екосистемі
Завдання 5	
5.1	Наявність механізму мутації генів під час моделювання
	Обґрунтування механізму мутації
5.2	Наявність можливості обміну генетичним матеріалом
5.3.	Реалізовано систему відстеження кожної істоти та її мутацій

ти прямою або складатися з двох прямих ліній по три відрізки. Лінія штрафного удару може перетинати лінії попередніх ударів, але не може перетинати або торкатися рамки ігрового поля. Голом вважається перетин лінії воріт. Якщо після виконання штрафного удару іншому гравцеві знову нікуди бити або лінія штрафного удару закінчилась на іншій лінії, виконується ще один штрафний удар тим самим гравцем. Приклад штрафного удару — коричнева лінія (3) на рис. 4.

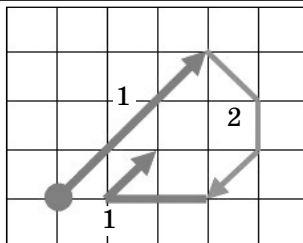


Рис. 3

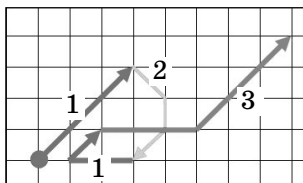


Рис. 4

Гра ведеться до гола. Після цього на новому аркуші малюється нове ігрове поле і починається новий тайм.

**Завдання**

1. Створіть програму гри в «футбол на аркуші».
2. Реалізуйте можливість гри у режимах людина — людина і людина — комп'ютер.

Після завершення II туру до журі Турніру капітаном кожної команди-учасниці подаються:

- текстові файли з кодами програмних продуктів;
- ехе-файл програмного продукту;
- документація до програмного продукту в електронному вигляді.

Повної реалізації не було в жодній з команд, хоча багато з них реалізували режим гри людина-комп'ютер більш-менш досконало. Деяко складними виявилися задача визначення, чи перетинає новий шлях м'яча вже існуючі лінії, та створення зручного інтерфейсу керування. Більшість команд обрали керування за допомогою миші, що не є досить зручним з огляду на невеликий розмір клітинки і необхідність точного наведення курсору на кути.

Роботи II туру оцінюються за критеріями: повнота реалізації поставленої задачі, досконалість інтерфейсу, досконалість роботи програмного продукту, документація до програмного продукту, презентація власної реалізації задачі-проекту.

**ЗАВДАННЯ III ТУРУ**

Як правило, у третьому турі із семи задач одна є дуже легкою, одна чи дві досить складними, а решта — середніми за складністю. Щоб надати змогу оцінити складність і цікавість задач наведемо їх умови повністю.

**1. Кола**

Завод з виробництва коли готує її не тільки для магазинів, але і для всесвітньо відомої мережі ресторанів швидкого харчування. Кожного дня завод відвантажує один і той же об'єм коли у літрах. Служба доставки мережі ресторанів, зазвичай, використовує для транспортування коли ємності об'ємом або тільки 50 літрів, або тільки 70 літрів. Якщо доставка здійснюється за допомогою ємностей у 50 літрів, то для перевезення об'єму коли необхідно  $A$  ємностей. А якщо за допомогою ємностей у 70 літрів, то необхідно  $B$  ємностей. Водночас у кожному із випадків одна з ємностей може бути заповнена не повністю.

Недавно мережа ресторанів вирішила затвердити новий об'єм ємностей для доставки коли — 60 літрів. Скільки ємностей тепер знадобиться для доставки того ж самого об'єму коли?

**Формат вхідних даних**

Два числа  $A$  і  $B$  ( $1 \leq A, B \leq 10\,000\,000$ ), які записані кожне в окремому рядку.

**Формат вихідних даних**

Виведіть усі можливі значення для ємностей по 60 літрів, як будуть заповнені (у тому числі одна можливо частково), у порядку зростання або число  $-1$ , якщо значення  $A$  і  $B$  суперечать одне одному, тобто вони були записані неправильно.

**Приклад вхідних і вихідних даних**

Введення	Виведення
3 2	2 3
1 2	-1

У першому прикладі коли могло бути, наприклад, 115 літрів, у цьому випадку потрібно дві ємності по 60 літрів, а могло бути — 135 літрів, у цьому випадку знадобиться уже три ємності по 60 літрів. Чотири ємності не можуть знадобитись ніколи.

**2. Часовий на посту**

Доблесні прикордонники Анчурії затримали на кордоні контейнер з підривною літературою — багатотомною *Encyclopaedia Britannica*. Відомо, що томи енциклопедії пронумеровані по порядку, починаючи з 1, і по крайньому разі один екземпляр кожного тому в контейнері був.

Напишіть програму, яка визначає:

- 1) номер останнього тому (тобто кількість томів у енциклопедії);
- 2) скільки повних комплектів енциклопедії можна зібрати із затриманих книг;
- 3) який том присутній у контейнері в найбільшій кількості (і яка це кількість); якщо є кілька варіантів різних томів, то визначити найменший із таких номерів;
- 4) яку максимально можливу кількість повних комплектів можна зібрати, якщо дозволяється замінити любі два томи на любий том по вибору.
- 5) програмі надається список томів у тому порядку, у якому їх витягували із контейнера — необхідно визначити, скільки книг довелося витягувати із контейнера, перш ніж вперше з них вдалося зібрати повний комплект енциклопедії.

**Формат вхідних даних**

У першому рядку міститься натуральне число  $N$  ( $2 \leq N \leq 10000$ ) — кількість книг у контейнері. У наступних  $N$  рядках містяться номери томів. Кожен рядок містить номер одного тому — одне натуральне число, яке не перевищує  $N$ .

**Формат вихідних даних**

Шість рядків. У кожному рядку повинно знаходитися одне ціле число: у першому рядку — відповідь на перше запитання, у другому — на друге, у третьому рядку повинна знаходитися найбільша кількість екземплярів одного будь-якого тому, що повторюється, а у четвертому рядку — номер цього тому (якщо таких номерів кілька, то виводьте найменший із них), у п'ятому рядку має знаходитися відповідь на четверте запитання, у шостому рядку — на п'яте запитання.

**Приклад вхідних і вихідних даних**

Введення	Виведення
7	3
3	1
2	3
2	1
1	2
1	4
2	
1	

**3. З почуттям впевненості у завтрашньому дні**

Вовочка дуже любив піклуватися про бабусь і про їхній добробут. Кожного  $N$ -го дня він переводив бабусь через дорогу, кожного  $K$ -го дня він оголошував, що добробут бабусь підвищився, а кожного  $M$ -го дня він обіцяв, що скоро їм буде ще краще. Сьогодні Вовочка робив всі три справи: і переводив, і оголошував, і обіцяв, і дуже втомився, але був задоволений собою.

Напишіть програму, яка визначає:

1) через скільки днів у Вовочки знову трапиться такий змістовний день?

2) скільки до цього моменту буде у Вовочки нудних днів, коли він не займався ні однією з улюблених справ?

3) чи трапиться коли-небудь так, що у Вовочки буде три цікавих дні поспіль (цікавим Вовочка вважає день, коли він займався хоча б однією з трьох улюблених справ); якщо так, то коли це трапиться вперше?

**Формат вхідних даних**

Три рядки: перший рядок містить натуральне число  $N$ , другий — натуральне число  $K$ , і третій — натуральне число  $M$ . Кожне із цих чисел не більше 1000.

**Формат вихідних даних**

Три рядки, кожен із яких має містити одне ціле число — відповідь на відповідне запитання. Якщо такого особливо вдального дня, коли Вовочка займався всіма трьома улюбленими справами одразу, більше ніколи не станеться, то перший рядок має містити  $-1$ . Якщо у Вовочки ніколи не трапиться трьох цікавих днів поспіль, то третій рядок має містити  $-1$ , якщо ж трапиться, то слід виводити номер першого з тих трьох днів, коли таке трапиться вперше. Вважаємо, що сьогоднішній день має номер 0.

**Приклад вхідних і вихідних даних**

Введення	Виведення
5	30
2	8
3	2

**4. ПДР**

Одного разу тато Петрика вирішив покатати його на автомобілі. Петрик зайняв переднє місце пасажира. На що тато одразу процитував правила дорожнього руху: *забороняється перевозити дітей, зріст яких менше 145 см на передньому сидінні.*

На скільки сантиметрів слід підрости Петрику, щоб сидіти на передньому сидінні, якщо зараз його зріст складає  $N$  см.

**Формат вхідних даних**

Єдиний рядок з даним числом  $N$  ( $1 \leq N \leq 145$ ).

**Формат вихідних даних**

Необхідно вивести одне число – на скільки сантиметрів слід підрости Петрику.

**Приклади вхідних і вихідних даних**

Введення	Виведення
100	45

**5. Одинички**

Дмитрик вивчив цифру 1 і сьогодні цілу годину виводив одинички у зошиті. Старший брат Дмитрика Петрик навчається в математичному класі. Він щойно вивчив алгоритм знаходження НСД (найбільший спільний дільник) двох чисел. Тепер його цікавить, як знайти НСД двох чисел, які записав Дмитрик у своєму зошиті з одиниць у двох різних рядках.

**Формат вхідних даних**

Два рядки, у кожному з яких записано по одному числу, кожне з яких складається тільки з одиниць. Кількість цифр у кожному числі не більше 20000.

**Формат вихідних даних**

Одне число – кількість цифр, що містить найбільший спільний дільник даних чисел.

**Приклади вхідних і вихідних даних**

Введення	Виведення
1 11	1
111 11	1
11 1111	2

**6. Новорічні іграшки**

У кожного свята є один недолік – рано чи пізно, але воно закінчується. Ось і новорічні свята завершилися і малому Дмитрику необхідно скласти іграшки у коробки. Частину іграшок він склав у одну коробку, а частину в іншу. Старший брат Дмитрика Петрик навчається в математичному класі. І його цікавить, чи можна перекласти всі іграшки у одну з коробок, якщо з одної коробки у іншу можна перекладати стільки іграшок, скільки є у іншій коробці.

**Формат вхідних даних**

Два числа  $N$  і  $M$  — кількість іграшок у першій і другій коробці ( $1 \leq N, M \leq 2000000000$ ).

**Формат вихідних даних**

Виведіть 1 – якщо можна перекласти іграшки в одну коробку, або 0 – якщо такої можливості немає.

**Приклади вхідних і вихідних даних**

Введення	Виведення
2 6	1
1 5	0
20 5	0

**7. Числа**

Саша і Катя навчаються у початковій школі. Для вивчення арифметики використовуються картки, на яких написані цифри (на кожній картці написана рівно одна цифра). Одного разу вони прийшли на урок арифметики, і Саша, використовуючи свої картки, показав число  $A$ , а Катя показала число  $B$ . Учитель тоді вирішив дати їм таку задачу, щоб відповідь на неї змогли показати і Саша, і Катя, кожен використовуючи свої картки. До того ж учитель хоче, щоб шукане число було максимальним.

**Формат вхідних даних**

Два невід'ємних числа  $A$  і  $B$ , кожне записане у окремому рядку. Довжина кожного із чисел не перевищує 100 000 цифр.

**Формат вихідних даних**

Виведіть одне число – максимальне ціле число, яке можна отримати, як за допомогою цифр першого числа, так і за допомогою цифр другого числа. Якщо ж ні одного такого числа скласти не можна, виведіть  $-1$ .

**Приклад вхідних і вихідних даних**

Введення	Виведення
280138 798081	8810
123 456	-1