

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
 Студентски клуб „Състезателно програмиране”
 Школа „Състезателно програмиране”
 СЪСТЕЗАНИЕ, 26 март 2011 г.

Зад. А. ИГРА С ПЛОЧКИ

Върху част от клетките на правоъгълна мрежа от квадрати с N реда и M стълба са поставени плочки. Напишете програма, която да провери дали е възможно плочка да бъде преместена от зададено място на дъската до друго зададено място на дъската. Плочката може да бъде местена от квадрата, в който се намира, в някой от съседните празни квадрати отляво, отдясно, отгоре или отдолу и не може да напуска мрежата.

На първия ред на **стандартния вход** е зададен броят на тестовете. Всеки тест започва с ред, на който са зададени целите числа N и M ($3 < N, M \leq 500$). Всеки от следващите N реда съдържа по един низ с дължина M , описващ поредния ред от дъската – знакът ‘*’, означава, че в съответния квадрат има плочка, а знакът ‘.’ (точка) – че квадратът е празен. Последният ред съдържа четири цели положителни числа – координатите на квадрата с плочката, която искаме да преместим, и координатите на празен квадрат, в който искаме да я преместим.

За всеки тест, на отделен ред на **стандартния изход** програмата трябва да изведе YES, ако преместването е възможно или NO, ако не е възможно.

ПРИМЕР

| Вход | Изход |
|-----------|-------|
| 2 | YES |
| 4 5 | NO |
| * * * * * | |
| * . . . * | |
| * * * . * | |
| . . . * . | |
| 3 1 4 3 | |
| 4 5 | |
| * * * * * | |
| * . . . * | |
| * * * . * | |
| . . . * . | |
| 1 2 4 5 | |

Забележка. Редовете са номерирани от 1 до N в реда по който са зададени на входа, а стълбовете от 1 до M – отляво надясно.

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача В. НОВА ТЕХНОЛОГИЯ

Програмистът Станчо обича да решава алгоритмични задачи. Ето една такава задача. Фирмата, за която Станчо от време-навреме работи, въвежда нова технология. Технологията вече е въведена в централата (означена с 1) и сега трябва да бъде разпространена в останалите офиси на фирмата в страната, означени с 2,3,..., N . Фирмата е класифицирала сътрудниците си в K категории (колкото по-нисък е номерът на категорията, толкова по-висока е квалификацията на сътрудника) и съставила за всеки от офисите си списък на категориите на работещи в него сътрудници. За разпространяване на технологията фирмата избрала следната стратегия. Сътрудник от офис, в който технологията е въведена, отива в офис, в който технологията не е въведена още и предава опита от работата с новата технология на домакините. Те оттам нататък се грижат за въвеждането \square в своя офис. Когато технологията е въведена в един офис, сътрудници от този офис могат да я “пренасят“ в други офиси. Ръководството на фирмата преценило, че най-изгодно е пренасянето на технологията от офис в офис да се извърши от **сътрудници в двата офиса с най-нисък еднакъв номер на категория**. В такъв случай, цената на пренасянето на технологията от единия офис в другия е равна на категорията на сътрудниците, които извършват пренасянето. Ако, обаче, такива сътрудници няма – пренасянето на технологията между тези два офиса може да стане между произволни двама сътрудници, но на цена $K + 1$. Най-важно за ръководството на фирмата е, технологията да достигне до всички офиси и пренасянето да стане на най-изгодната цена. На Станчо възложили да напише програмата, която решава тази задача. Опитайте се да я решите и вие.

На първия ред на **стандартния вход** ще бъде зададен боят на тестовете. Всеки тест започва с ред, на който са зададени числата N и K – броят на офисите и броят на категориите, $3 \leq N \leq 1000$, $5 \leq K \leq 32$. На всеки от следващите N реда е зададен списъкът на категориите на сътрудници в един от офисите на фирмата – списъкът за i -тия офис е на i -тия от тези редове. Списъкът започва с броя R , $1 \leq R \leq K$, на различните категории на сътрудниците в съответния офис, последван от R различни цели числа в интервала от 1 до K , подредени в нарастващ ред.

За всеки тест, на отделен ред на **стандартния изход** програмата трябва да изведе минималната цена, която фирмата трябва да плати за разпространяването на технологията.

ПРИМЕР

| Вход | Изход |
|---------|-------|
| 1 | 7 |
| 3 6 | |
| 3 1 2 3 | |
| 3 3 4 5 | |
| 3 4 5 6 | |

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача С. СНЯГ

И тази година падналият сняг създава проблеми в града X , с неговите N кръстовища (номерирани с числата от 1 до N) и M улици. Всяка улица свързва две кръстовища, като движението по всички улици е двупосочно и някои кръстовища може да са свързани с повече от една улица. Улиците свързват кръстовищата така, че от всяко кръстовище може да се стигне до всяко друго кръстовище. Улиците са доста широки и за да бъде почистена една улица от снега, чистещата машина трябва да премине по нея два пъти – по един път във всяка от двете посоки на улицата. Естествено, градската управа би искала да почисти всички улици на града с колкото може по-малко разходи и поставила задачата: да се намери маршрут на чистещата машина, който да започва от най-близкото до гаража кръстовище (номерирано с числото 1), минава по всяка улица точно един път във всяка от двете посоки и завършва в кръстовището, от което е тръгнала, без да губи време и гориво за да се придвижва по вече изчистеното. Напишете програма, която решава поставената задача.

На първия ред на **стандартния вход** ще бъде зададен броят на тестовете. На първия ред за всеки тест са зададени целите числа N и M , разделени с един интервал ($5 \leq N \leq 1000$, $5 \leq M \leq 100000$). Всеки от следващите M реда съдържа по две числа, разделени с един интервал – два номера на кръстовища, които са свързани с улица.

За всеки тестов пример програмата трябва да изведе на един ред на **стандартния изход** намерения маршрут – списък от номерата на кръстовища, през които минава маршрутът, започвайки от върха с номер 1 и завършвайки във върха с номер 1.

ПРИМЕР

| Вход | Изход |
|------|---------------------------|
| 1 | 1 2 3 5 1 4 3 2 1 5 3 4 1 |
| 5 6 | |
| 1 2 | |
| 1 4 | |
| 1 5 | |
| 2 3 | |
| 3 4 | |
| 3 5 | |

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача D. БЪРЗО ПРИДВИЖВАНЕ

В големите градове бързината на придвижването не винаги зависи само от дължината на маршрута, който трябва да бъде изминат, а също и от натовареността на движението и пропускливостта на отделните улици. Пропускливостта се определя от ширината на улицата, от качеството на настилката, от нейната безопасност и т.н. и, в края на краищата, се измерва с някакво цяло положително число, което можем да наричаме *ширина*. Колкото по-голяма е ширината, толкова по-голяма е пропускливостта на улицата. Ширината на един маршрут в града, обаче, е равна на ширината на най-тясната улица по маршрута. Напишете програма, която в зададен град с N кръстовища (номерирани с числата от 1 до N) и M улици, всяка от които свързва две от кръстовищата, намира маршрут с най-голяма ширина от кръстовището с номер 1 до кръстовището с номер N , ако такъв маршрут съществува.

На първия ред на **стандартния вход** ще бъде зададен броят на тестовите примери. Всеки тест започва с ред, на който са зададени, разделени с един интервал, числата N и M , $2 \leq N \leq 1000$, $1 \leq M \leq 100000$. Всеки от следващите M реда ще съдържа по три цели положителни числа I , J и W , $1 \leq I \neq J \leq N$, $1 \leq W \leq 30000$, разделени с по един интервал, които означават, че кръстовищата с номера I и J са свързани с улица с ширина W .

За всеки тестов пример програмата трябва да изведе на отделен ред на **стандартния изход** най-голямата ширина на маршрут от кръстовището с номер 1 до кръстовището с номер N , при условие, че такива маршрути съществуват. В противен случай програмата трябва да изведе на стандартния изход числото 0.

ПРИМЕР

| Вход | Изход |
|--------|-------|
| 2 | 3 |
| 5 6 | 0 |
| 1 2 1 | |
| 1 3 3 | |
| 1 4 9 | |
| 2 5 10 | |
| 3 5 4 | |
| 4 5 2 | |
| 6 5 | |
| 1 3 4 | |
| 1 4 3 | |
| 3 4 5 | |
| 2 6 6 | |
| 5 6 7 | |

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача Е. ВЛОЖЕНИ ОТСЕЧКИ

Върху една права лежат няколко отсечки. Всяка от тях е зададена чрез координатите на двата си края. Отсечките са номерирани от 1 до N ($0 < N \leq 500$). Казваме, че една отсечка е *вложена* в друга, ако двете отсечки са различни и първата се съдържа изцяло във втората (това не изключва по един от двата им края да съвпадат). Напишете програма, която извежда номерата на отсечките от най-дългата последователност от вложени една в друга отсечки, т.е. такава последователност, че всяка отсечка в нея, с изключение на последната, е вложена в следващата отсечка.

На първия ред на **стандартния вход** ще бъде зададен броят на тестовите примери. Всеки тест започва с ред, съдържащ числото N . Следват N реда, на всеки от които са зададени по две числа – координатите на левия и десния край на поредната отсечка. Координатите са цели числа от интервала $[-10000, 10000]$. Номерацията на отсечките е по реда на срещането им във входа.

За всеки тестов пример, на отделен ред на **стандартния изход** програмата трябва да изведе едно цяло число – броя на отсечките в максималната последователност от вложени една в друга отсечки.

ПРИМЕР

| Вход | Изход |
|--------|-------|
| 1 4 | 3 |
| -2 2 | |
| -1 1 | |
| -3 3 | |
| 4 5 | |

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача F. ПРИЯТЕЛИ

В един град живеят N жители, за някои двойки от които се знае, че са приятели. От известната максима “Приятелите на моите приятели са и мои приятели” следва, че ако A и B са приятели, и B и C са приятели, то A и C също са приятели. Напишете програма, която намира броя на гражданите в най-голямата група от приятели.

На първия ред на **стандартния вход** е зададен броят на тестовете. Всеки тест започва с ред, на който са зададени числата N и M , където N е броят на жителите ($10 \leq N \leq 10000$), а M е броят на двойките, за които се знае, че са приятели ($0 < M \leq 50000$). На всеки от следващите M реда има по две числа – номерата на двойка приятели A, B ($1 \leq A \leq N, 1 \leq B \leq N, A \neq B$), като между дадените двойки може да има и повтарящи се.

За всеки тестов пример програмата трябва да изведе на **стандартния изход** едно число – броя на гражданите в най-многобройната група от приятели.

ПРИМЕР

| Вход | Изход |
|--|-------|
| 1 10 12 1 2 3 1 3 4 5 4 3 5 4 6 5 2 2 1 7 10 1 2 9 10 8 9 | 6 |

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача G. БРОЙ НА ЦИФРИТЕ

Да се намери броя на цифрите в десетичния запис на естествено число. Числото е представено като произведение на простите си множители. Основната теорема на аритметиката гласи, че всяко цяло число, по-голямо от 1, може да се представи по единствен начин (с точност до реда на множителите) като произведение на прости числа.

На **стандартния вход** са зададени много примери, всеки на отделен ред. Умножението се означава с "x" (латинско хикс), а степенуването - с "^" (колибка), при записва няма шпации. Множителите на числото и техните степени са числа, не по-големи от 10^5 .

За всеки пример от входа, на **стандартния изход** се отпечават по едно число на отделен ред - броят на десетичните цифри на съответното число от входа.

ПРИМЕР

| Вход | Изход |
|--------------------------|--------------|
| 19 | 2 |
| $2^2 \times 3^2$ | 2 |
| $19 \times 5^2 \times 3$ | 4 |

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача Н. ПРОСТИ ЧИСЛА

Взаимно прости числа в се наричат две или повече цели числа, чиито единствени общи делители са 1 и -1 или, изразено по друг начин, чиито най-голям общ делител е единица. Един бърз начин за определяне дали две числа са взаимно прости е най-древният известен алгоритъм - алгоритъмът на Евклид, с който се намира най-големият общ делител на две числа. В частност алгоритъмът разпознава дали две числа са взаимно прости. Някои алгоритми за кодиране на предаване на данни използват произведение на две прости числа. За да бъде разбит подобен канал за предаване на данни, трябва да бъдат намерени самите прости числа, като се знае тяхното произведение. Да се намери броя на всички числа в даден интервал, които са произведение на точно две прости числа.

На **стандартния вход** са зададени много примери, всеки на отделен ред. Всеки пример се определя от две числа - долна и горна граница на интервала. Числата са не по-големи от 10^5 .

За всеки пример от входа, на **стандартния изход** се отпечават по едно число на отделен ред - броят на търсените числа в съответния интервал.

ПРИМЕР

| Вход | Изход |
|-------|-------|
| 1 10 | 4 |
| 23 43 | 7 |

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача I. БРОЙ НУЛИ

Факториел е функция на естествено число n , която изразява произведението на всички естествени числа, по-малки или равни на n ($n! = 1.2.3.4...n$ - произведение на числата от 1 до n). Използва се в теорията на числата (например за изразяване на коефициентите на Нютоновия бином), при разлагането на аналитичните функции, например синус и косинус, в ред на Тейлър, което позволява практическото им изчисление с дадена точност, и т.н. Да се напише програма за пресмятане на броя на нулите, които се съдържат в десетичния запис на числото $n!$ (n е цяло положително число).

На **стандартния вход** е зададена редица от числа, разделени с интервал, не по-големи от 10^5 .

За всяко число от входа, на **стандартния изход** се отпечатва броя на нулите на факториела му. Разделител между числата е интервал.

ПРИМЕР

| Вход | Изход |
|-------------|--------------|
| 2 5 11 | 0 1 2 |

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
Студентски клуб „Състезателно програмиране”
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 26 март 2011 г.

Задача J. ДВОИЧНО ПРЕДСТАВЯНЕ

Двете устойчиви състояния на електронните цифрови елементи се означават условно с 1 и 0. Те обуславят и основната единица за представяне на информацията в компютъра, наречена бит (от Binary digit). Понятието разрядност на числата означава броя на цифрите в тях. При записа и обработването им в компютър на всеки разряд на двоичното число съответства двоичен (електронен или магнитен) елемент. Следователно, може да се счита, че информацията се представя в компютъра като съвкупности (масиви, множества) от двоични (дискретни) елементи. Тези масиви образуват т. нар. машинни думи (разрядни решетки). Броят на едновременно обработваните разряди определя разрядността на компютъра и дължината на думата. Първите компютри - 2-разрядни, 4-разрядни, след това - 8-, 16-, а понастоящем 32- и 64-разрядни. Образуването на двоични числа и операциите с тях са обект на двоичната аритметика, която се основава на двоичната бройна система. Да се намери броят на цифрите със стойност 1 в двоичното представяне в компютъра на цяло положително число.

На **стандартния вход** е зададена редица от цели числа (по-малки от 10^9 , записани в десетична бройна система) която завършва с числото 0.

За всеки елемент на редицата (без последния) на **стандартния изход** се извежда броят на единиците за съответното число от входа - на един ред с разделител интервал.

ПРИМЕР

| Вход | Изход |
|---------------------|--------|
| 1024 99999999 255 0 | 1 19 8 |