XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача А. ПОЛИНОМ

Напишете програма, която намира коефициента пред *k*-тата степен на *x* в полинома $f(x)=(x-1)^n(x+1)^m$.

Вход

От първия ред на стандартния вход се въвежда броят *p* на тестовите примери (0<*p*<11). Следват *p* реда, на всеки от които са зададени по три цели числа – стойностите на *n*, *m* и *k* за съответния тестов пример.

Изход

За всеки тестов пример програмата трябва да изведе на отделен ред коефициента пред *k*-тата степен на *x* в съответния полином.

Ограничения

$n, m \geq 0,\ n+m \leq 60,\ 0 \leq k \leq n+m.$

Примерен вход	Примерен изход
2 3 1 3 4 0 2	-2 6

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача В. СУПЕР

Бойко е суперпрограмист в най-големия супермаркет на Балканите. Отговорен е за супербързото информиране на супершефа за състоянието на цените на стоките в супермаркета по видове. Супершефът се интересува от суперцените (най-висока и най-ниска цена) от даден вид стока и колко артикула се продават на тези суперцени. Бойко си има един супермасив с цените на всички стоки по видове. Например всички цени на всички видове велосипеди са елементи на супермасива с индекси от $\mathbf{v_1}$ до $\mathbf{v_2}$ и в този интервал са само цени на велосипеди.

Поради кризата и други причини, непрекъснато постъпват данни за промяна на цената на някой артикул. Бойко трябва да направи суперпрограма, която да следи за промяна на цените на артикулите и да дава супербързо отговори на запитвания на супершефа за суперцените на даден вид стока.

Супермасивът на Бойко се състои от **N** (1 <= **N** <= 100 000) числа – цените на стоките. За заблуда на конкуренцията, Бойко кодира всяка цена така, че елементите на супермасива се оказват цели числа в интервала [-1 000 000, 1 000 000]. Заявките за промяна на някоя цена и за информиране на супершефа са оформени както следва:

Заявка 1:

1 x v

– **x** е новата цена на артикул, записан в супермасива с индекс **v** (1 <= **v** <= **N**)

Заявка 2:

2 $\mathbf{v_1}$ $\mathbf{v_2}$

– да се дадат на супершефа суперцените, и колко артикула се продават на тези суперцени за вида стока, описана в елементите на супермасива с индекси от $\mathbf{v_1}$ до $\mathbf{v_2}$ (1 <= $\mathbf{v_1}$ <= $\mathbf{v_2}$ <= **N**)

Вход: На стандартния вход се задават няколко теста. Всеки тест започва с двойка естествени числа на отделен ред - **0 N**. Следват не повече от 300 000 заявки от двата описани вида. След последната заявка на поредния тест има число **3** на отделен ред.

Изход: За всяка заявка от втория вид на стандартния изход на отделен ред да се изведат данните за супершефа – максималната стойност на елементите на супермасива в затворения индексен интервал [$\mathbf{v_1}$, $\mathbf{v_2}$], колко пъти се среща тази стойност в този интервал и същото за минималната стойност.

Примерен вход	Примерен изход
1 0 10 -1 2 2 2 3 4 5 7 1 3 2 1 10 2 2 3 1 9 3 2 6 10 1 6 -2 2 1 10 3	-1 1 7 1 2 2 2 2 3 2 7 1 -2 1 7 1

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача С. ДЪРВОТО

Дадено е дърво, състоящо се от N върха ($2 \le N \le 150\,000$), номерирани с числата от 1 до N и ребра с дължина 1. Върховете са оцветени с K цвята ($1 \le K \le 50\,000$), като всеки цвят има номер – число в интервала [1, K]. Всеки връх е оцветен само в един цвят. С всеки от цветовете са оцветени най-малко два върха. Да се намери разстоянието между двата най-далечни върха от група върхове, оцветени с еднакъв цвят.

Вход: На първия ред на стандартния вход е зададен броят на тестовете. За всеки тест на първия ред са зададени числата N и K, разделени с един интервал. Следват N реда с по две числа, разделени с по един интервал. *i*-тия ред съдържа предшественика на връх с номер *i*, следван от число – номера на цвета, с който е оцветен този връх. Коренът на дървото е даден с предшественик 0.

Изход: За всеки тест програмата трябва да изведе K числа, всяко на отделен ред на стандартния изход. *k*-тия ред трябва да съдържа най-дългото разстояние между кои да е два върха от групата върхове, оцветени с цвят *k*.

Примерен вход	Примерен изход
1 15 4 0 2 1 1 1 3 2 4 2 4 3 2 3 3 4 2 4 3 5 3 5 4 6 1 6 2 7 2 7 1	4 6 5 3

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача D. РЕДИЦА

Жълтото пате непрекъснато измисля шантави редици. Днес в училище то научило операцията „повдигане на степен”, и веднага измислило нова редица.

Първо, патето записва на дъската естествено число A. Всяко следващо число, написано от него на дъската, ще бъде равно на степен с основа A и показател, равен на предходното число. С други думи, редицата ще изглежда така:

$$x[1] = A,$$
$$x[k + 1] = A^{x[k]}, k > 0$$

Доволно от изобретението си, патето започнало да изследва свойствата на своята редица. Ето защо то иска да разбере кой е елементът на тази редица с минимален номер, който ще се дели без остатък на дадено число **N**. Тъй като числата на дъската са огромни, отново ще е необходима вашата помощ. Напишете програма, която по дадени числа **A** и **N** намира номера на търсения минимален елемент на редицата, който се дели на **N.**

Вход

От стандартния вход се въвежда цяло число **t** – брой тестови примери. За всеки тестов пример се въвеждат две естествени числа **A** и **N** ($1 \le A \le 10^9, 1 \le N \le 10^9$).

Изход

За всеки тестов пример изведете резултата по следния начин: ако нито един елемент на редицата не се дели на **N**, изведете **0**. Иначе изведете минималния номер на елемент от разглежданата редица, който се дели на **N**.

Примерен вход	Примерен изход
2 2 2 2 4	1 2

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача Е. ЛОДКИ

Жълтите патета организират традиционната олимпиада по програмиране. Тази година тя ще се проведе на най-отдалеченото място на езерото и патетата ще трябва да използват лодки, за да не се уморят, плувайки до там. Организаторите предвидили това и осигурили дълъг прав пристан, на който имало множество места за връзване на лодките. Ако се приеме най-лявото място за нулево, всички останали места се определят с разстоянието им до него. Да наречем това разстояние координата на мястото. Патетата са много конфликтни личности и спирането на лодките близо една до друга може да доведе до инциденти. Затова, след като всички заявки за участие били приети, организаторите решили да разпределят местата за връзване на лодките на различните отбори така, че те да са възможно най-отдалечени една от друга. Тъй като заявките са много, ще се наложи да се напише програма **B**, която по зададени координати на местата за връзване и брой отбори, разпределя местата на лодките така, че най-малкото възможно разстояние между две лодки да е максимално.

Вход

От първия ред на стандартния вход се въвежда цяло число **t** – брой тестови примери. За всеки тестов пример данните се въвеждат на два реда: на първия ред са две цели числа **N (2<N<10001)** – броя на местата за връзване на лодките и **K(1<K<N)** – броя лодки. На втория ред се задават **N** естествени числа в нарастващ ред– координатите на местата за лодките (координатите не надвишават 10^9).

Изход

За всеки тестов пример изведете едно число – най-голямото възможно допустимо разстояние между лодки на пристана.

Примерен вход	Примерен изход
1 5 3 1 2 3 100 1000	99

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача F. КЛАСИРАНЕ

За поредната Републиканска студентска олимпиада по програмиране, треньорът на един университет записал пет отбора. Какво било учудването му, когато след края на олимпиадата се оказало, че всичките му отбори имат един и същ брой решени задачи и ще трябва да бъдат класирани според сумарното времето за предаване на решените задачи. Затова треньорът поискал да му списък с времената. Студентите, развеселени от ситуацията, скроили на треньора си следния номер: вместо да му напишат петте времена му дали лист, на който били написани десет числа – всички възможни суми от времената на някои два от отборите – и му предложили сам да си пресметне петте времена.

Треньорът изобщо не се стреснал, направил някаква сметка и съобщил на студентите, че когато са пресмятали сумите вероятно са сбъркали, защото десетте числа, които са му дали, не могат да бъдат получени по описания начин от никои 5 положителни цели. Студентите направили бърза проверка и се оказало, че треньорът е прав. Дали му нови 10 положителни цели числа и той бързо намерил петте времена и класирането на отборите.

Напишете програма, която по зададени 10 положителни цели числа да намира такива 5 положителни числа, че сумите им по двойки да са точно зададените 10 числа, или да съобщава че такива 5 положителни цели числа не съществуват.

На първия ред на **стандартния вход** ще бъде зададен броят на различните случаи, които програмата трябва да разреши. За всеки от тези случаи, на отделен ред на входния файл, ще бъдат зададени по 10 положителни цели числа, всяко от които не е по-голямо от 2 000 000 и разделени едно от друго с интервал.

За всеки от случаите, програмата трябва да изведе на един ред на **стандартния изход** намерените пет положителни цели, сортирани в намаляващ ред и разделени с по един интервал, или низа IMPOSSIBLE, ако не съществуват пет положителни цели с указаните свойства.

Примерен вход	Примерен изход
2 1 2 3 4 5 6 7 8 9 10 34 12 48 6 20 36 10 18 24 40	IMPOSSIBLE 32 16 8 4 2

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача G. МРАВКИ

Мравките Анти и Панти били номинирани за разузнавачи на отбора на Термитград, участващ в ежегодната мравешка олимпиада по събиране на зърна. Те решили да си разделят работата по претърсване на градината на фермера бай Иван ред по ред по следния начин: започват да търсят от две противоположни посоки докато се срещнат по средата на реда. Много скоро Анти и Панти открили, че в доста редове зърната жито, овес, ечемик са симетрично разположени спрямо средата от грижливия бай Иван и решили да претърсват само половината редове. За нещастие обаче небрежният син на бай Иван е объркал позициите на зърната в някои редове и те са почти симетрични, но се различават в една позиция. За още по-голямо нещастие палавият внук на бай Иван напълно е разместил зърната в други редове и те съвсем не са симетрични спрямо средата. Тъй като нивата се намира близо до боровата гора, редовете в нивата не са еднакво дълги.

Съставете програма, която да помогне на Анти и Панти да определят дали даден ред е симетричен, почти симетричен или не е симетричен. Ако е симетричен, изведете 1, ако не е симетричен – 0, а ако е почти симетричен - двата възможни симетрични реда, съдържащи еднаквите знаци на същите симетрични позиции, разделени с интервал

Примерен вход	Примерен изход
abccba abcba abcda abcd	1 1 abcba adcda 0

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача Н. МРЕЖАТА

Двама приятели Ани и Митко съвместно използват интернет канал. Системата за балансиране на натоварването е така конфигурирана, че ако в даден момент каналът се използва само от един човек, то изтеглянето на файла се извършва със скорост равна на честотната лента, а ако каналът се използва от няколко души, то пропускателната способност на канала се разделя поравно между потребителите. Загубата на скорост при предаване на управляваща информация по канала не се отчита. В процеса на сваляне на файлове няма допълнително активност в мрежата. В отговор се получава цяло число в секунди.

Например нека честотната лента е 256 Kbytes. Ани започва да изтегля файл с размер 10 Mbytes. Известно време след това Митко започва да изтегля файл, с размер 6 MBytes. Митко завършва изтеглянето на своя файл 8 секунди по-късно от Ани. Изчислете за колко секунди всеки един от потребителите Ани и Митко изтеглят файловете си?

Вход

На стандартният вход са дадени: на първи ред скоростта на канала, на втори ред са размера на файловете на потребителите, на трети ред е времето закъснението на втория потребител, зададено в цяло число секунди, за което втория поробител завършва файла си след като първия е приключил.

Изход

На стандартният изход изведете времената за изтегляне на файловете на потребителите в цяло число - секунди.

Примерен вход	Примерен изход
256 10 6 8	24 32

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача I. БАНИЦА

Освен ценител на българската минерална вода, Станчо е и голям почитател на традиционната българска кухня. Поради това той решил да събере екип и да направят рекордна баница от N парчета (до тогава били правени само баници с M и с X парчета).

След като проектът бил реализиран и документиран, се появил обаче един сериозен проблем – баницата трябвало да се изяде. За да бъде яденето честно трябва всеки от K-те човека участвали в приготвянето да получи равен брой парчета от различните видове (например, ако Станчо получи 1 парче със сирене, 3 с праз и 2 със спанак, то и всички останали K-1 човека трябва да получат същите). За да не се развали конструктивната цялост на баницата обаче е важно тя да не бъде нарязвана прекалено много, т.е. парчетата да си останат слепени така както са били. Всеки трябва да получи точно един цял отрез от баницата, която представлява линейна последователност от парчета с различна плънка. Вие трябва да напишете програма, която по дадена баница и брой участници да определи колко най-дълъг отрез от баницата може да получи всеки, така че да са изпълнени горните условия.

Вход: На стандартния вход са дадени няколко конфигурации на баници и хора. Всяка от тях съдържа един низ от не-повече от 4000 малки латински букви и броят на хора в екипа K ($1 \le K \le 10$). Всяка буква кодира различен вид парче от баницата.

Изход: За всяка от дадените конфигурации вашата програма трябва да изведе на стандартния изход ред съдържащ едно число – максималния брой парчета, които може да получи всеки, така че яденето да е честно.

Пример

Вход	Изход	Пояснение (примерни отрези)
banitsabgsriba 3	2	(ba)nits(ab)gsri(ba)
hellohelloinbs 2	5	(hello)(hello)inbs
abcdedcbacdb 3	3	a(bcd)e(dcb)a(cdb)
mnogohora 10	0	

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача J. НАЙ-ДОБЪР КАБИНЕТ

Поради ремонтни дейности и преструктуриране, както и поради рязкото повишаване на средната продължителност на живота, във факултета на Станчо са решили да разместят кабинетите на катедрите и Станчо трябва да избере нов кабинет. Той обаче иска да пробва различни кабинети преди да се спре на най-подходящия. Тъй като е супер умен, не му било трудно да измисли план как да го постигне.

Тъй като факултетците са широко скроени, във факултета се използват най-различни бройни системи за записване на числа. От администрацията обаче хич не се справят с това, от което Станчо мисли да се възползва по следния начин:

1. Да заяви и пробва номер кабинет в някаква бройна система (например $100_{(10)}$).
2. Ако не му хареса да заяви, че записът е бил в друга бройна система (напр. осмична) и съответно да вземе друг кабинет ($100_{(8)} = 64$).
3. да поиска от администрацията бележка за назначения му кабинет, но в друга бройна система (например $64_{(10)}$).
4. отново да интерпретира записа в друга бройна система (например $64_{(8)} = 52$) и т.н.

Кабинетите във факутета са неограничени (изброима безкрайност), но поради изчислителните възможности на администрацията, не е прието да се използват числа с повече от *L* цифри и бройни системи с основа повече от *R*. Тъй като Станчо иска да пробва максимален брой кабинети преди да реши, напишете програма, която определя дали даден начален номер на кабинет е подходящ за тази цел.

Вход: На стандартния вход са дадени няколко факултетостанчови конфигурации. Всяка от тях представлява ред с целите числа *R* ($2 \le R \le 10$), L ($2 \le L \le 10$) и X (с не-повече от *L* цифри в *R*-ична бройна система).

Изход: За всяка от дадените конфигурации изведете на стандартния изход ред съдържащ YES ако началния номер е такъв, че с него може да изпробва възможно най-голям брой кабинети и NO в случай, че има по-добър.

Примерен вход	Примерен изход
10 2 99 10 2 33 8 8 1	NO YES NO

XXIV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
11 – 12 май 2012, БСУ, Бургас

Задача К. БИБЛИОТЕКА

В университетската библиотека винаги е съществувал проблем с бързото намиране на книга по нейното наименование. Въпреки че книгите са подредени по азбучен ред, намирането на заглавие става все по-трудно поради големия брой на книгите.

Управителката на университетската библиотека реши да организира каталог с имената на книгите в библиотеката, като за всеки стелаж състави по два азбучни индекса на заглавията. Всеки от индексите тя състави като използва възможно най-кратката началната част от името на книгата, намираща се на първа и последна позиция на всеки от стелажите. Индексите все пак направи достатъчни по дължина, за да може да определи еднозначно стелажа, на който трябва да се намира заглавието, която даден студент търси, ако е налично в библиотеката. Така тя може да гарантира, че ако името на книгата по азбучен ред е между два индекса на даден стелаж, то тя трябва да бъде намерена на този стелаж и не присъства на никой от останалите стелажи. Вие трябва да помогнете на управителката да състави каталог на библиотеката.

Вход: Първи ред съдържа броя на стелажите ($1 \le S \le 10000$) и броя на книгите ($1 \le B \le 10000$), които могат да се поставят на всеки от стелажите. Следват N ($1 \le N \le 10000$) брой реда, съдържащи по едно име на книга на всеки ред. Името на книгата е съставено от ASCII символи на латиница без разделител и не може да е празно. Имената на книгите са не-по дълги от 250 символа и не могат се повтарят. Последният ред съдържа името на последната книга и символ за край на файла.

Изход: За всеки вход се извеждат $1 \le R \le S$ реда с резултати. Всеки ред съдържа номера на стелаж, на който има поне една книга, начален и краен индекси на книги, разделени с интервал. Празните стелажите не се извеждат. Ако книгите не могат да се разположат на стелажите се извежда ред със съобщението `DATA ERROR`.

Примерен вход:	**Примерен изход:**
4 5 Book20 Book12 Book13 Book11 Book03 Book04 Book02 Book05 Book01 Book19 Book18 Book17 Book16 Book14 Book10 Book06 Book09 Book08 Book07	1 Book01 Book05 2 Book06 Book10 3 Book11 Book16 4 Book17 Book20
Примерен вход:	**Примерен изход:**
1 1 Book1 Book2	DATA_ERROR