



Sapientia
Hungarian University of Transylvania

Sapientia ECN International Programming Contest – 2015

PROBLEMS



Târgu Mureş / Marosvásárhely, 9th May 2015

PROBLEM A

Fighting against corruption

Input File:	A . IN
Output File:	STANDARD OUTPUT

Andrew is the general prosecutor of Landia and he decided to fight against corruption. As a result of his decision, every time an offender is caught in the act, the personal data of the offender are entered in a special database, together with a qualifier of the offence (common or heavy).

Input

The input file contains two lists. The first list is the database of Landia's people : the total number of persons on the first line ($np \leq 160000$), and the personal data of every person on the next np lines (personal numeric code : nine decimal digits, and the name of person : five to fifty characters).

The second list is the special database which is managed by Andrew. Every record is the personal numeric code of an offender, optionally preceded by the sign + (heavy offence). If the sign + is present, it is not followed by space. This list contains an unspecified number of records. White spaces and empty lines may occur freely in this list. Every occurrence of a particular record is counted as follows : one point for common offence and five points for heavy offence.

Output

Andrew wants to know the number and the percentage of honest people, the number and the percentage of corrupt people (two digits after the decimal point). He also wants to know who the champions of corruption are. You have to display at least two persons who cumulated most points. If the second place and the third place cumulated the same number of points, you must also display the third place, and so on. If two or more persons cumulated the same number of points, order them alphabetically.

Restrictions and refinements

- Response time : 2 seconds

Final remark

- The above story is pure fiction. Any resemblance to real persons, living or dead, is purely coincidental.

Sample Input: A.in
9
19 Alberto Mori
11 Clara Sordi
18 Ella Vats
12 Iannis Korfu
17 Mata Hara
13 Vera Aloe
16 Olive Hank
14 Bill Kido
21 Frenk Tomis
+21 14 +17 11
+14 21 +14 17

Sample Output: Standard output
Honest: 5 persons, 55.56%
Corrupt: 4 persons, 44.44%
Champions of corruption:
14 Bill Kido, 11 points
21 Frenk Tomis, 6 points
17 Mata Hara, 6 points

PROBLEM B

Project Selection

Input File:	B . IN
Output File:	STANDARD OUTPUT

Given a number of n projects with an expected $I[i]$ income and m equipments with $P[i]$ purchase price. Each project requires a number of equipments and each equipment can be shared by several projects. Find the maximum profit that can be realized by selecting the optimal set of projects and equipments, rejecting the lossy ones.

Input

The first line of the input file contains the number t of test cases. The first line of each test case includes the number n of projects and number m of equipments. The next line contains the $I[i]$ values separated by space, while the following line the $P[i]$ values. Next follows number k of dependencies between projects and equipments. The following k lines contain the actual dependencies, the first number represents de project which requires the equipment denoted by the second value.

The numbering of projects and equipments starts at 1.

Output

For each test case your program has to print on a separate line the maximum profit that can be achieved.

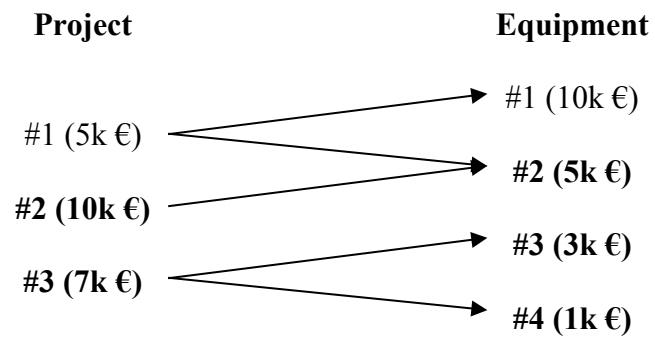
Restrictions and refinements

- $1 \leq t \leq 10$
- $1 \leq n, m \leq 100$
- $1 \leq I[i], P[i] \leq 100$ (thousands of euros)
- $1 \leq k \leq 10000$
- every input and output value will be an integer

Sample Input: B.IN
1 3 4 5 10 7 10 5 3 1 5 1 1 1 2 2 2 3 3 3 4

Sample Output: Standard output
8

Explanation



We choose project 2 and 3 which suppose the acquisition of equipment number 2, 3 and 4. The first equipment is too expensive, project 1 requiring it would generate loss for the company. With this solution the total income would be $10+7=17$, while the total cost $5+3+1=9$, resulting in a profit of $17-9=8$ thousand euros.

PROBLEM C

The elusive Queen

Input File:	C . IN
Output File:	STANDARD OUTPUT

From the bottom left black corner of an $n \times n$ chessboard, the elusive black Queen can move in arbitrary way to up, to the right, and diagonally up-right, but she can touch only the black squares.

In how many different ways can the elusive black Queen get to a given x, y square on the chessboard?

Input

The first line of the input file contains the number t of test cases. The first line of each test case includes the number n (the dimension of the chessboard), the second line the numbers x and y separated with a space (the coordinates of the square which must be reached by the elusive black Queen).

Output

For each test case your program has to print on a separate line a number o , the number of different ways, in which the elusive black Queen can reach the given square.

Restrictions and refinements

- $1 \leq t \leq 100$
- $1 \leq n \leq 1000$
- $1 \leq x, y \leq n$

Sample Input: C.IN
2
5
5 1
6
6 1

Sample Output: Standard output
40
126

PROBLEM D

The boulangerie

Input File:	D . IN
Output File:	STANDARD OUTPUT

A boulangerie is a French bakery (aka, baker's shop or bake shop) that produces and sells flour-based food baked in an oven, such as bread, cakes, croissants, pastries, and pies. Some of them are also cafés, serving coffee and tea to customers who wish to consume the baked goods on the premises.

In their activity the most important product is flour so every Monday they estimate and buy the quantity (q) of flour they will need for that week.

They can buy it from the wholesaler who sells in bulk with a price of (p) / kg but they also work with small mills who propose interesting prices but they sell the flour in prepackaged bags of different size. Please help the bakery to buy their flour the cheapest they can.

Input

- First line will contain (q), the quantity of flour they want to buy this week.
- In the second line you will find the unitary price (p) they have to pay for 1kg of flour at the wholesaler.
- The next lines describe the mills (*maximum of 200 mills*). Each line will contain the pair of numbers ($bagQuantity$, $bagPrice$) representing the size and the price of the flour bag.

Output

- Should contain the minimal price the bakery has to pay to buy the required quantity **rounded to two decimals**.

Restrictions and refinements

- $0 < q \leq 100$, $0 < p \leq 1000$, $0 < bagQuantity \leq 1000$, $0 < bagPrice \leq 1000$,
- q , p , $bagQuantity$ and $bagPrice$ are real numbers, and they have two decimals maximum.
- **Note:** the wholesaler can sell in all quantities, the mills are small entities and they can sell only with the quantities they have communicated.

Sample Input: D.IN	
12	//kg of flour to be bought
2.20	//wholesaler unit price for 1 kg
8 2	//first mill sells 8kg of flour for 2€
4.4 1.5	//second mill sells 4.4 kg of flour for 1.5€
14 6	//third mill sells 14 kg of flour for 6€

Sample Output: Standard output	
10.04	//minimal price to spend to buy the 12 kg flour
(it can be achieved by buying 2 packs from the second mill: 8.8 kg for 3€ and the remaining 3.2 kg from the wholesaler for a price of 7.04€ (3.2 * 2.2))	

PROBLEM E

3 primes

Input File:	E . IN
Output File:	STANDARD OUTPUT

Given a natural number n , decompose it as a sum of three distinct prime numbers.
How many decompositions in three distinct prime numbers are there?
Two cases are distinct, if they differ by at least one number.

Example

If $n=23$

$23=5+7+11$ correct case

$23=11+5+7$ similar with the previous case

$23=2+2+19$ wrong case, because it has two prime numbers with the same value

$23=3+7+13$ correct case

Number 23 has two correct decompositions.

Input

The first line contains n , the number of test cases. The following n lines contain one v_i number ($i=1,2, \dots, n$), for which we must calculate the number of decompositions of v_i as three distinct prime numbers.

Output

The standard output will contain n lines with one number, the number of decompositions of v_i as a sum of three distinct prime numbers.

Restrictions and refinements

- $3 \leq n \leq 32010$;
- $1 \leq k \leq 32000$;
- Time limit: 1 sec/test
- Source file limit: 10k

Sample Input: E.IN
3 10 23 88

Sample Output: Standard output
1 2 4

PROBLEM F

Apocalypse

Input File:	F.IN
Output File:	STANDARD OUTPUT

The PaMayan calendar says that the end of the world is coming. Again. The exact date is not known, but after many research articles have been written, somebody proved that there is an ancient method, that given any text one can calculate whether a day of a year is dangerous or not.

The method is as follows: count the words and the letters in a document, and calculate the average word length in the text (we call *word* a sequence of letters ('a'..'z', 'A'..'Z')). Raise the number to the power of 3. Then take the integer part of the resulting $number \bmod 12+1$, this will be the month of the predicted event. The day can be calculated by taking the $integer\ part\ of\ the\ number \bmod (days\ of\ the\ selected\ month)+1$. If the day is today then... well, bad luck.

Input

The first line of the input file contains the number t of test cases, and then the input text file contains t paragraphs, each paragraph being a text to analyze. The last line of the file is empty.

There will be no newline characters inside one paragraph. A paragraph is closed by the characters LF+CR or just CR.

Output

Write in separate lines to the standard output for every test case:

YES – if today is the day,

NO – if it will be another day.

Restrictions and refinements

- $1 \leq t \leq 20$
- The size of a paragraph can be up to 10MB of text.

Sample Input: F.IN
2 There is a strong tradition of "world ages" in Mayan literature, but the record has been distorted, leaving several possibilities open to interpretation. The 2012 "doomsday" phenomenon was a range of eschatological beliefs that cataclysmic or otherwise transformative events would occur around date 21 December 2012.

Sample Output: Standard output
NO YES

PROBLEM G

Longest common palindromes

Input File:	G . IN
Output File:	STANDARD OUTPUT

Find the longest palindrome substrings common to two given strings. A palindrome string spelled backwards is the same.

Input

The first line stores the number of the test-cases. The two strings of a test-case are stored in two consecutive lines.

Output

For each test-case print to a new line the number of the case, and to the next lines all distinct longest common palindrome substrings in alphabetical order, putting one palindrome per line.

Print NO when there are no common palindromes.

Restrictions and refinements

- The length of each string is between 1 and 80.
- All input lines are terminated by ENTER.

Sample Input: G.IN
4 abaabbab aababbaab aaaaaa bbbbbbbbbb ababababab babababababab abaaaba bbbb

Sample Output: Standard output
1 abba baab 2 NO 3 ababababa babababab 4 b

PROBLEM H

Mathematics by Experiment

Input File:	No
Output File:	STANDARD OUTPUT

In the standard mathematics – at least at the level of the usual presentation – one starts taking some properties and formulas already proved and by logical reasoning and straightforward computation steps one deduce new properties, formulas.

It is less obvious, and even paradoxical, that properties and formulas can be discovered by pure computer experiment, without any reasoning and any use of previously established properties! All is done by using the computer's numerical computation capabilities only.

Let us consider a tiny but illustrative example: discover an interesting, highly nontrivial formula yourself.

Take the sequence of cubes of the natural numbers:

1	2	3	4	5	6	7	8	9	10	11	...
1	8	27	64	125	216	343	512	729	1000	1331	...

Take a prime number p , take the integer division of cubes by p up to the $(p-1)$ -th cube, and then take the sum of all these quotients. For example if $p = 7$, you have

0	1	3	9	17	30					$\Sigma :$	60
---	---	---	---	----	----	--	--	--	--	------------	----

The sum obviously is a function f of p . The question is: has this function f any *closed form*? We can look for the simplest one, a *polynomial formula*. If such a polynomial function exists, it has only a finite number of terms, say $n+1$ if the degree of the polynomial $f(p)$ is n , but it will give the sum Σ for all – infinitely many – values of primes p . This would be indeed remarkable!

Your task is to search for such a formula just up to degree 3. The formula you guess – if there exists – must give the correct value of the sum for all primes up to the 100-th prime (only)! Remember that the sequence of primes is 2, 3, 5, 7, 11, ...

Input

no input file

Output

It should contain either: NO, if no polynomial formula up to degree 3 exists, or else, the value of the sum $\sum_{i=1}^{100} 4f'(p_i)$, where p_i is the i -th prime, and f' is the derivative of the polynomial f .

Example: It is not necessary.

PROBLEM I

How Old Are You?

Input File:	I . IN
Output File:	STANDARD OUTPUT

...

- *Here is the filled form.*
- *Thank you. Let me check... hum... OK, OK, OK... Wait, how old are you?*
- *20. Did I forget to fill it?*
- *No. It says here that you'll be born next month! The year is wrong...*
- *Oh... Sorry!*

The process is going to be automatic, and to avoid some human errors, there will be a calculated field that informs the age based on the current date and the birth date given. This is your task: calculate the age, or say if there's something wrong.

Input

The first line of the input gives the number of test cases, T ($1 \leq T \leq 200$). T test cases follow. Each test case starts with a blank line, then you will have two lines corresponding to the current date and the birth date, respectively. The dates are in the format $DD/MM/YYYY$, where DD is the day, MM is the month, and $YYYY$ is the year. All dates will be valid.

Output

The output is comprised of one line for each input data set and should be as follows (quotes for clarifying only): “Case # N : AGE “, where N is the number of the current test case, and AGE is one of the following three options:

- “Invalid birth date”, if the calculated age is impossible (still going to be born).
- “Check birth date”, if the calculated age is more than 130.
- the calculated age (whole years only), otherwise.

Sample Input: I.IN
4
01/01/2007
10/02/2007
09/06/2007
28/02/1871
12/11/2007
01/01/1984
28/02/2005
29/02/2004

Sample Output: Standard output
Case #1: Invalid birth date
Case #2: Check birth date
Case #3: 23
Case #4: 0

PROBLEM J

The Pentagon

Input File:	J . IN
Output File:	STANDARD OUTPUT

The American army had n ($1 < n < 100$) military units in Iraq (identified by $1, 2, \dots, n$). The Pentagon was informed that k units had been eliminated by Iraqi rebels. Is the following reaction of the Pentagon justified: “Don’t worry! The remaining units are still in communication range with each other!”?

Input

The first line of the input file contains the number t of test cases. The first line of each test case includes the numbers n, k, m (m is the number of directly connected units). The next m lines contain the directly connected pairs of units.

Output

For each test case your program has to print on a separate line YES or NO.

Sample Input: J.IN
2 3 1 3 1 2 1 3 2 3 3 1 2 1 2 1 3

Sample Output: Standard output
YES NO

PROBLEM K

Hash values

Input File:	K.IN
Output File:	STANDARD OUTPUT

Several methods are known to calculate the hash value of a given password. In this problem you will use a baby variant:

1. We suppose that a password is formed using only the lowercase letters of the English alphabet. Along the password a salt value is known, formed using only the lowercase letters of English alphabet, too.
2. First calculate a normalized value: after the password, if needed repeatedly append the salt value to reach the necessary length. If the resulting string is longer than the necessary length cut as many characters as you need.

Example:

"passw" "salt", 10 -> "passwsalts"

"passw" "salt", 3 -> "pas"

3. Use the following formula on $c_1c_2\dots c_n$ normalized value: $\sum_{i=1}^n i \cdot 2^{\text{code}(c_i)}$, where $\text{code}(c_i) = \text{ASCII code}(c_i) - 97$.

Input

The first two rows of K.IN file contain a password and the length of the normalized value. The next rows contain several hash values in base26, followed by \$ as separation, and the corresponding salt value. The base26 form of a number can be obtained by converting the number to base 26 and adding 97 to each digit. This will correspond to an 'a'...'z' character for each digit.

Output

Your task is to write a program that examines which hash value belongs to the given password, writing on standard output the position of the valid hash value. If doesn't belongs to any hash value your program will be write on standard output INVALID.

Restriction and refinements

- Max length of normalized value = 10
- output value will be an integer

Sample Input: K.IN	Sample Output: Standard output
passw 10 wuxigg\$xyzxyz jptce\$abcefg rymnr\$salt sqjce\$abcefg	3
Sample Input: K.IN	Sample Output: Standard output
passw 7 wuxigg\$xyzxyz jptce\$abcefg rymnr\$salt sqjce\$abcefg	INVALID

