The CENTRE for EDUCATION
in MATHEMATICS and COMPUTING

2010 Canadian Computing Competition: Senior Division

Sponsor:

University of Waterloo

Canadian Computing Competition
Student Instructions for the Senior Problems

1. You may only compete in one competition. If you wish to write the Junior paper, see the other problem set.

2. Be sure to indicate on your **Student Information Form** that you are competing in the **Senior** competition.

3. You have three (3) hours to complete this competition.

4. You should assume that

   - all input is from a file named `sX.in`, where `X` is the problem number ($1 \leq \texttt{X} \leq 5$).
   - all output is to the screen

   Since your program will read from a file, there is no need for prompting. Be sure your output matches the output in terms of order, spacing, etc. IT MUST MATCH EXACTLY!

5. Do your own work. Cheating will be dealt with harshly.

6. Do not use any features that the judge (your teacher) will not be able to use while evaluating your programs.

7. Books and written materials are allowed. Any machine-readable materials (like other programs which you have written) are *not* allowed. However, you are allowed to use "standard" libraries for your programming languages; for example, the STL for C++, java.util.*, java.io.*, etc. for Java, and so on.

8. Applications other than editors, compilers, debuggers or other standard programming tools are **not** allowed. Any use of other applications will lead to disqualification.

9. Please use file names that are unique to each problem: for example, please use `s1.pas` or `s1.c` or `s1.java` (or some other appropriate extension) for Problem S1. This will make the evaluator's task a little easier.

10. Your program will be run against test cases other than the sample ones. Be sure you test your program on other test cases. Inefficient solutions may lose marks for some problems, especially Problems 4 and 5. Be sure your code is as efficient (in terms of time) as possible.

11. Note that the top 2 Senior competitors in each region of the country will get a plaque and $100, and the schools of these competitors will also get a plaque. The regions are:

    - West (BC to Manitoba)
    - Ontario North and East
    - Metro Toronto area

- Ontario Central and West
- Quebec and Atlantic

12. If you finish in the top 20 competitors on this competition, you will be invited to participate in CCC Stage 2, held at the University of Waterloo in May 2010. Should you finish in the top 4 at Stage 2, you will be a member of the Canadian IOI team at IOI 2010, held in Canada. Note that you will need to know C, C++ or Pascal if you are invited to Stage 2. But first, do well on this contest!

13. Check the CCC website at the end of March to see how you did on this contest, and to see who the prize winners are. The CCC website is:

    `www.cemc.uwaterloo.ca/ccc`

Problem S1: Computer Purchase

Problem Description

In order to increase your performance on the ABC (Another Buying Contest), you decide that you need a new computer. When determining which computer to buy, you narrow your search categories to:

- RAM (in gigabytes), denoted as R;
- CPU speed (in megahertz), denoted as S;
- disk drive space (in gigabytes), denoted as D.

You perform some analysis and determine that the most preferred machine is the machine that has the largest value of the formula

$$2 * R + 3 * S + D.$$

Your task is to read a given list of computers and output the top two computers in order of preference, from highest preference to lowest preference.

Input Specification

The first line of input will be an integer n ($0 \leq n \leq 10000$). Each of the remaining n lines of input will contain a computer specification. A computer specification is of the form:

- computer name (a string of less than 20 characters)
- the RAM available (an integer R with $1 \leq R \leq 128$)
- the CPU speed (an integer S with $1 \leq S \leq 4000$)
- the disk drive space (an integer D with $1 \leq D \leq 3000$)

There is one space between the name, RAM, CPU speed and disk drive space on each line.

Output Specification

The output is the name of the top two preferred computers, one name per line, sorted in decreasing order of preference. If there is a tie in the rankings, pick the computer(s) whose name(s) are lexicographically smallest (i.e., "Apple" is smaller than "Dell"). If there is only one computer, output that computer on one line (i.e., do not print it twice).

Sample Input

```
4
ABC 13 22 1
```

```
DEF 10 20 30
GHI 11 2 2
JKL 20 20 20
```

Output for Sample Input

```
JKL
DEF
```

Explanation of Output for Sample Input

Computer ABC has a computed value of 93. Computer DEF has a computed value of 110. Computer GHI has a computed value of 30. Computer JKL has a computed value of 120. Therefore, computer JKL is the most preferred, followed by computer DEF.

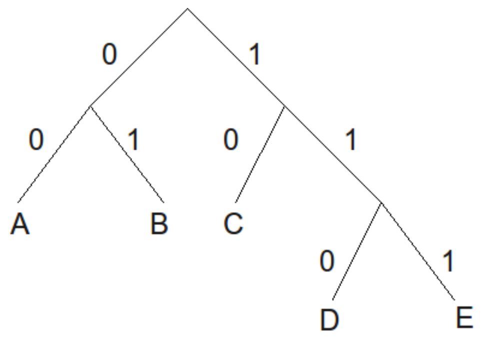
Problem S2: Huffman Encoding

Problem Description

There is an ingenious text-compression algorithm called *Huffman coding*, designed by David Huffman in 1952.

The basic idea is that each character is associated with a binary sequence (i.e., a sequence of 0s and 1s). These binary sequences satisfy the *prefix-free property*: a binary sequence for one character is never a prefix of another character's binary sequence.

It is worth noting that to construct a prefix-free binary sequence, simply put the characters as the leaves of a binary tree, and label the "left" edge as 0 and the "right" edge as 1. The path from the root to a leaf node forms the code for the character at that leaf node. For example, the following binary tree constructs a prefix-free binary sequence for the characters $\{$A, B, C, D, E$\}$:



That is, A is encoded as 00, B is encoded as 01, C is encoded as 10, D is encoded as 110 and E is encoded as 111.

The benefit of a set of codes having the prefix-free property is that any sequence of these codes can be uniquely decoded into the original characters.

Your task is to read a Huffman code (i.e., a set of characters and associated binary sequences) along with a binary sequence, and decode the binary sequence to its character representation.

Input Specification

The first line of input will be an integer k ($1 \leq k \leq 20$), representing the number of characters and associated codes. The next k lines each contain a single character, followed by a space, followed by the binary sequence (of length at most 10) representing the associated code of that character. You may assume that the character is an alphabet character (i.e., 'a'...'z' and 'A'..'Z'). You may assume that the sequence of binary codes has the prefix-free property. On the $k+2$nd line is the binary sequence which is to be decoded. You may assume the binary sequence contains codes associated with the given characters, and that the $k+2$nd line contains no more than 250 binary digits.

Output Specification
On one line, output the characters that correspond to the given binary sequence.

Sample Input
```
5
A 00
B 01
C 10
D 110
E 111
00000101111
```

Output for Sample Input
```
AABBE
```

Problem S3: Firehose

Problem Description
There is a very unusual street in your neighbourhood. This street forms a perfect circle, and the circumference of the circle is 1,000,000. There are H ($1 \leq H \leq 1000$) houses on the street. The address of each house is the clockwise arc-length from the northern-most point of the circle. The address of the house at the northern-most point of the circle is 0.

You also have special firehoses which follow the curve of the street. However, you wish to keep the length of the longest hose you require to a minimum.

Your task is to place k ($1 \leq k \leq 1000$) fire hydrants on this street so that the maximum length of hose required to connect a house to a fire hydrant is as small as possible.

Input Specification
The first line of input will be an integer H, the number of houses. The next H lines each contain one integer, which is the address of that particular house, and each house address is at least 0 and less than 1,000,000. On the $H + 2$nd line is the number k, which is the number of fire hydrants that can be placed around the circle. Note that a fire hydrant can be placed at the same position as a house. You may assume that no two houses are at the same address. Note: at least 40% of the marks for this question have $H \leq 10$.

Output Specification
On one line, output the length of hose required so that every house can connect to its nearest fire hydrant with that length of hose.

Sample Input
```
4
0
67000
68000
77000
2
```

Output for Sample Input
```
5000
```

Problem S4: Animal Farm

Problem Description
You are running a farm which has N ($1 \leq N \leq 100$) animals. You went to the store and bought $M = N$ pre-made pens that will house your animals. Pens satisfy the following conditions:

- pens have between 3 and 8 edges;
- an edge that is specified by two pens connects the two pens;
- an edge that is specified only once connects that pen to the outside;
- there is exactly one animal in each pen and no animals outside the pens, initially.

The animals, however, have a game they like to play called "Escape from the pen." They assign a cost to each edge of the pen, and they determine the minimum cost for all of the animals to meet in the same area by trampling over the edge of various pens. The animals may meet inside a particular pen or outside of all the pens. Also note that once an edge has been trampled, any animal may pass over it without incurring any cost.

You will be given a description of the pens, along with the placement of animals, and you are to figure out what the smallest cost is to move all the animals into the same area.

Input Specification
The first line of input will be the integer M, the number of pens. On the next M lines, there will be a description of each pen, with one description per line. The description is composed of three components, with each component separated by one space, as follows:

- the first component is an integer e_p ($3 \leq e_p \leq 8$), which describes the number of edges for this particular pen p;
- the second component is a sequence of e_p integers describing the corners of each pen, where each integer is less than or equal to 1000;
- the third component is a sequence of e_p integers describing the cost of each edge, where each integer is less than or equal to 5000.

For the corner and edge cost description, the descriptions are given in cyclical order. For example, the following description of a pen

$$3\ 1\ 2\ 3\ 7\ 4\ 6$$

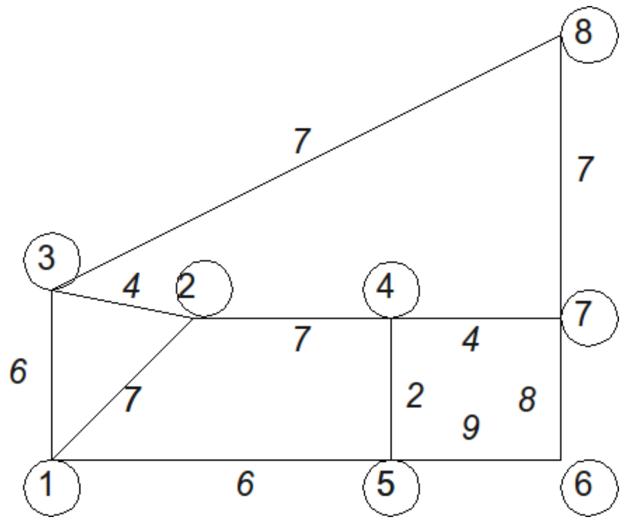
means that there are three corners, and thus, three edges, where the edge $(1, 2)$ has cost 7, the edge $(2, 3)$ has cost 4 and the edge $(3, 1)$ has cost 6. Note: at least 20% of the marks for this question have $N \leq 10$ and no pen will have more than four edges in these cases.

Output Specification
On one line, output the minimal cost that will allow all the animals to gather in one pen or outside all of the pens.

Sample Input
```
4
3 1 2 3 7 4 6
4 1 2 4 5 7 7 2 6
4 4 7 6 5 4 8 9 2
5 3 2 4 7 8 4 7 4 7 7
```

Output for Sample Input
```
10
```

Explanation of Output for Sample Input
The diagram below explains the input data:



where the circled numbers are the corners, and the numbers in *italics* are the edge costs. Notice that if the edges $(2, 3)$, $(4, 5)$ and $(4, 7)$ are removed, all the animals can meet in the pen which has five sides.

Problem S5: Nutrient Tree

Problem Description
There is a binary tree with l leaves $(1 \leq l \leq 50)$ where each leaf k has an amount of nutrients n_k $(1 \leq n_k \leq 10,000)$ that it produces.

The branches (which you can think of as edges) of this binary tree limit the maximum amount of nutrients that can flow to the root of the tree. You have X growth agents $(1 \leq X \leq 2500)$ that can be used to increase the thickness of an edge or increase the nutrient production of a leaf node. Initially, each edge has a weight of 1 and if you give it w growth agents then it can transport $(1+w)^2$ nutrients. Increasing the nutrient production of a leaf with initial value n_k by s raises the nutrient production of that leaf to $n_k + s$.

Notice that when edges meet, the amount of nutrient that flows is the sum of nutrients flowing along the incoming edges.

Find the maximum amount of nutrients you can transport to the root.

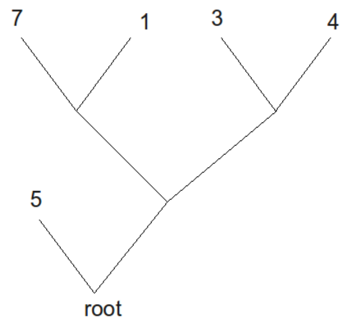
Input Specification
The first line of input will be a description of the tree. This description can be defined recursively as either an integer n_k $(1 \leq n_k \leq 10,000)$ or as $(T_L T_R)$ where T_L and T_R are descriptions of the left and right subtrees, respectively. The second line of input will be the integer X, the amount of growth agents you have. Note: at least 30% of the marks for this question have $l \leq 5$ and $X \leq 50$.

Output Specification
On one line, output the maximum amount of nutrients that can flow into the root of the tree.

Sample Input
```
(5 ((7 1) (3 4)))
3
```

Output for Sample Input
```
7
```

Explanation of Output for Sample Input
The tree description can be illustrated as:



Notice that if we put two growth factors on the left edge of the root, and one growth factor on the right edge of the root. Then, there will be 5 nutrients flowing from the leaf labelled 5 to the root (since the capacity is $(1+2)^2 = 9$ units of nutrients) and 2 nutrients flowing from the right subtree (since the capacity is $(1+1)^2 = 4$, but there are only 2 units of nutrients due to limits of 1 on the edges attached to all other leaves).

Alternatively, notice that increasing the nutrient production of the left leaf of the root to 6, and increasing the capacity of the root to the left leaf by 2 (to give a capacity of $(1+2)^2 = 9$ units) would cause 6 units of nutrients to flow into the root from the left tree, and the right subtree of the root would produce 1 unit of nutrient.

In both cases, the maximum amount of nutrients that can reach the root node is 7.