Programming Design, Spring 2016
Lab Exam 2

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

For all the problems in this exam, you are allowed to use any technique.

Problem 1

(30 points) Given n integers in binary and decimal encoding, find their sum in hexadecimal encoding.

Input/output formats

There are 15 input files. In each file, there are $n+1$ lines. The first line contains n, which is a positive integer no greater than 100. Starting from the second line, each line contains an integer $b \in \{2, 10\}$ and a string s. s is a binary integer if $b = 2$ or a decimal integer if $b = 10$. In any case, s will be nonnegative and no greater than 9999. There is a white space between b and s. Given the input, your program should find their sum and print it out in hexadecimal encoding. For example, the following input

```
3
2 101
2 1101
10 12
```

requires you to print out

```
1E
```

because $5 + 13 + 12 = 30$, which is represented as 1E in hexadecimal encoding.

Grading criteria

30 points will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. Each fully correct set of outputs gives you 2 points.

Problem 2

(30 points) Adam is going to travel through n cities. For the road connecting cities i and j directly, we record its distance as $d_{ij} > 0$. By defining $d_{ii} = 0$ for all i, we have an $n \times n$ matrix D recording the distances among all cities. Starting from city 1, Adam wants to find a shortest way to travel through the n cities and then be back to city 1. His algorithm is the following:

1. From city 1, find city k_1 that is closet to city 1.
2. From city k_1, find city k_2 that has not been visited and is closet to city k_1.
3. Keep doing so to find city k_3, k_4, ..., and k_{n-1}.

At any iteration, if multiple cities are all the closet to the source city, choose the one with the smallest index. Then the route is $(1, k_1, k_2, ..., k_{n-1}, 1)$.

Input/output formats

There are 15 input files. In each file, there are $n+1$ lines. The first line contains a single integer n. It is known that $1 \leq n \leq 100$. The $(i+1)$th lines contains n integers, d_{i1}, d_{i2}, ..., and d_{in}. Two consecutive integers are separated by a white space. It is known that $d_{ii} = 0$ for all i and $d_{ij} \in \{1, 2, ..., 100\}$. Your program should find a route from city 1 to travel through all cities exactly once and then back to city 1 by the algorithm defined in this problem. The route should be printed out as n integers 1, k_1, k_2, ..., and k_{n-1}. Two consecutive integers should be separated by a white space. For example, the following input

```
4
0 2 3 2
1 0 5 4
1 3 0 5
1 2 3 0
```

requires you to output

```
1 2 4 3
```

Grading criteria

30 points will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. Each fully correct set of outputs gives you 2 points.

Problem 3

(40 points) Given a list of n product sales transactions, Eve would like to do some statistics. A transaction contains three attributes: the sales date, product ID of the sold product, and category ID of the sold product. Given all the transactions and a month, Eve wants to find the category whose total number of transactions in that month is the highest.

Input/output formats

There are 20 input files. In each file, there are $n+2$ lines. The first line contains a single integer n which is between 1 and 1000. Each of the second to the $(n+1)$th lines containing a sales date, a product ID, and a category ID. The sales date is provided as three integers, separated by two white spaces. For example, a transaction on May 1, 2016 is recorded as `2016 5 1`. The product ID is an integer between 1 and 1000. Finally, the category ID is a single English capital letter. Two consecutive attributes are separated by a white space. The last line contains a month in the format yyyy-mm. For example, May 2016 is represented as `2016-05`.

Your program should print out the category whose total number of transactions in that month is the highest, that total number of transactions, and the number of distinct products sold in that category. If multiple categories all have the highest sales quantity, pick the one with the smaller number of distinct products sold. If there is still a tie, pick the one that is ranked the first in the alphabetical order (A before B, B before C, etc.). Two consecutive values should be separated by a white space.

For example, the following input

```
10
2016 5 1 1 A
2016 5 6 1 A
2016 5 16 2 A
```

```
2016 6 3 2 A
2016 6 9 2 A
2016 5 2 3 B
2016 5 13 4 B
2016 5 23 5 B
2016 5 1 6 C
2016 5 1 6 C
2016-05
```

shows ten transactions, and the target month is May, 2016. In May, 2016, there are 3, 3, and 2 transactions in categories A, B, and C, respectively. However, as 2 and 3 distinct products are sold in categories A and B, the winner is A, and the output should be

```
A 3 2
```

Grading criteria

40 points will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. Each fully correct set of outputs gives you 2 points.