


CODECHEF

 A **Directi** Educational Initiative
Hello **indrajohn** ▾

Logo

PRACTICE

COMPETE

DISCUSS

COMMUNITY

HELP

ABOUT

Home » Practice(Challenge) » Optimal Shopping II

Optimal Shopping II

Problem code: SHOP2

ALL SUBMISSIONS

MY SUBMISSIONS

SUBMIT

Tweet

Like

Share

Be the first of your friends to like this.

All submissions for this problem are available.

SUCCESSFUL SUBMISSIONS Read problems statements in [Mandarin Chinese](#) and [Russian](#) as well.

There are N junctions and M bidirectional roads, connecting these junctions. The junctions are numbered from 1 to N inclusive. For each road the time required to pass it is known.

There's a shop in every junction. Each shop contains some of K types of goods. In i -th shop, the goods of the j -th type cost $A_{i,j}$ rubles.

Any item of the i th type of goods weighs for W_i grams.

Initially, you are standing at the junction 1. You want to get to the junction N and buy **exactly** one unit of each of the goods so that your total budget does not exceed F rubles and the penalty is minimized (see next section for its details). Please note that buying goods take negligible time, so buying goods and switching roads can be considered instantaneous. Also, passing through junctions and roads multiple times is allowed. Also, note that you are also allowed to pass through junction N also, as long as in the end, you end up at this junction.

The penalty is defined as the sum of $T_i * W_i$ over all types of goods, where T_i denotes the time between the moment of the purchase of the item of the goods of the i th type and the finish moment. Intuitively, penalty is the cost you have to pay in carrying the goods.

Input

The first line contain four space-separated integer numbers N, M, K, F denoting the number of junctions, the number of roads, the number of types of goods and the budget respectively.

Each of the following K lines (say, the i th) start with a pair of integer numbers P_i and W_i followed with P_i pairs of integers $B_{i,j}, A_{i,j}$, where $B_{i,j}$ denotes a shop containing i th good, the shop is located at junction $B_{i,j}$ and the cost of i th good at this junction will be $A_{i,j}$.

Each of the following M lines (say, the k th) contains three space-separated integer numbers X_k, Y_k, Z_k , which denotes that there is a road whose two endpoint junctions are given by X_k, Y_k and the time required to pass this junction is given by Z_k .

Output

The first line of the output should contain an integer number D , denoting the number of commands.

The following line should contain D integer numbers, denoting the commands in the following format: Let the number outputted in some command is d .

- If d is positive, then it represents a move from the current junction to the adjacent junction numbered d .
- If d is negative, that it represents buying of the goods of type $-d$ in the current shop.

In the beginning, the current junction is always the first junction.

You will get a wrong answer verdict when:

- You try to move to the junction that is not adjacent to the current one.
- You don't finish at the junction N .
- You spend more than F rubles.
- You try to buy some goods in some shop where they're unavailable.
- You try to buy goods of same type twice or some.
- You don't buy all the goods by the time you finish.
- The number of commands exceeds **2000000**.

Constraints

In order to ensure that the answer will always exist, the value of the sum of the minimal costs over all goods will not exceed F .

- It is possible to get from any junction to any other junction via the given roads.
- $1 \leq Z_k \leq 10^4$
- $1 \leq W_i \leq 10^4$
- $1 \leq F \leq 10^7$
- A shop can contain at most one good of a particular type, Hence for each junction i , $B_{i,j}$'s will be distinct.

Group 1:

- $40 \leq N \leq 50$
- $80 \leq M \leq 100$
- $20 \leq K \leq 40$
- $10 \leq A_{i,j} \leq 30$

Group 2:

- $40 \leq N \leq 50$
- $80 \leq M \leq 100$
- $20 \leq K \leq 40$
- $10000 \leq A_{i,j} \leq 50000$

Group 3:

- $900 \leq N \leq 1000$
- $1000 \leq M \leq 10000$
- $100 \leq K \leq 500$
- $10 \leq A_{i,j} \leq 30$

Group 4:

- $900 \leq N \leq 1000$
- $1000 \leq M \leq 10000$
- $100 \leq K \leq 500$
- $10000 \leq A_{i,j} \leq 50000$

Test data generation

Each of the four groups will contain five test cases. During the contest, you will only see the score on the set of four test cases - one from each group.

The junction-road network structure will be random. The values of **N**, **M**, **K** will be picked from the given intervals randomly and independently.

For the distribution of the goods between shops, there will be picked a non-random interval **[L; R]**, which is common for all goods within a test case. Each of the goods will go to **P** random shops, where **P** is picked randomly from the segment **[L; R]**.

For each of the goods, the weight is picked randomly from the segment **[1; 10^4]**.

Then, the sum **S** of minimal costs over all goods is calculated. The value of **F** will be equal to **S** multiplied by some random number within range **[1.0; 1.1]** (and rounded).

Scoring

Your score for a single test case will be calculated as a square root of the value of the penalty function, described in the statement.

Your score for the problem will be calculated as a sum of individual scores for each test case.

Example

Input:

```
4 6 2 28
3 50 2 13 4 17 3 15
2 91 3 19 2 15
2 1 34
1 3 46
3 4 18
1 4 54
2 4 90
3 2 38
```

Output:

```
5
2 -1 -2 3 4
```

Explanation

Input

There are **4** junctions, **6** roads and **2** types of goods. The total budget is **28**.

The goods of the first type does weigh **50** and are sold in the shops at the junctions **2, 4, 3** with costs **13, 17, 15** respectively.

The goods of the second type does weigh **91** and are sold in the junctions **3, 2** with costs **19, 15** respectively.

Output

Let's perform the outputted commands:

- Command **2** means to move to junction 2. There is such a road, so the command is correct.
- Command **-1** means to buy goods of the type 1. So we buy goods of the type 1, and spend **13** rubles.
- Command **-2** means to buy goods of the type 2. So we buy goods of the type 2 and spend **15** burles. Now we've spent **28** burles in total, which is OK, since we didn't exceed the budget of **28** burles.
- Command **3** means to move to the third junction. Meanwhile, the penalty becomes equal to $(50 + 91) * 38 = 5358$.
- Command **4** means to move to the fourth junction. The penalty function becomes equal to $(50 + 91) * (38 + 18) = 7896$. We can finish now, since all goods are bought and we've reached the final junction.

Author: xcwgf666

Tester: kevinso

Editorial: <http://discuss.codechef.com/problems/SHOP2>

Tags: challenge mav16 travel salesman xcwgf666

regu. challenge may be over soon! Don't miss it!

Date Added: 29-03-2016

Time Limit: 5 sec

Source Limit: 50000 Bytes

Languages: ADA, ASM, BASH, BF, C, C99 strict, CAML, CLOJ, CLPS, CPP 4.3.2, CPP 4.9.2, CPP14, CS2, D, ERL, FORT, FS, GO, HASK, ICK, ICON, JAVA, JS, LISP clisp, LISP sbcl, LUA, NEM, NICE, NODEJS, PAS fpc, PAS gpc, PERL, PERL6, PHP, PIKE, PRLG, PYPY, PYTH, PYTH 3.1.2, RUBY, SCALA, SCM chicken, SCM guile, SCM qobi, ST, TCL, TEXT, WSPC

SUBMIT

Comments ▾

[CodeChef is a non-commercial competitive programming community](#)

[About CodeChef](#) | [About Directi](#) | [CEO's Corner](#) | [C-Programming](#) | [Programming Languages](#) | [Contact Us](#)

© 2009 **Directi Group**. All Rights Reserved. CodeChef uses SPOJ © by **Sphere Research Labs**

In order to report copyright violations of any kind, send in an email to copyright@codechef.com

Directi
Intelligent People. Uncommon Ideas.

The time now is: 03:29:55 PM
Your IP: 172.22.224.201, 144.49.50.1

CodeChef - A Platform for Aspiring Programmers

CodeChef was created as a platform to help programmers make it big in the world of algorithms, **computer programming** and **programming contests**. At CodeChef we work hard to revive the geek in you by hosting a **programming contest** at the start of the month and another smaller programming challenge in the middle of the month. We also aim to have training sessions and discussions related to **algorithms**, **binary search**, technicalities like **array size** and the likes. Apart from providing a platform for **programming competitions**, CodeChef also has various algorithm tutorials and forum discussions to help those who are new to the world of **computer programming**.

Practice Section - A Place to hone your 'Computer Programming Skills'

Try your hand at one of our many practice problems and submit your solution in a language of your choice. Our **programming contest** judge accepts solutions in over 35+ programming languages. Preparing for coding contests were never this much fun! Receive points, and move up through the CodeChef ranks. Use our practice section to better prepare yourself for the multiple **programming challenges** that take place through-out the month on CodeChef.

Compete - Monthly Programming Contests and Cook-offs

Here is where you can show off your **computer programming** skills. Take part in our 10 day long monthly **coding contest** and the shorter format Cook-off **coding contest**. Put yourself up for recognition and win great prizes. Our **programming contests** have prizes worth up to INR 20,000 (for Indian Community), \$700 (for Global Community) and lots more CodeChef goodies up for grabs.

Programming Tools

[Online IDE](#)

[Upcoming Coding Contests](#)

[Contest Hosting](#)

[Problem Setting](#)

[CodeChef Tutorials](#)

[CodeChef Wiki](#)

Practice Problems

[Easy](#)

[Medium](#)

[Hard](#)

[Challenge](#)

[Peer](#)

[School](#)

[FAQ's](#)

Initiatives

[Go for Gold](#)

[CodeChef for Schools](#)

[Campus Chapters](#)