**Problem Statement**

DIRECTIONS:

a. Use any programming language of your choice

b. Your program must be fast, efficient and, most importantly, produce the correct output. Credit if you have taken care of error handling and show us some really interesting aspect of software design or coding with your solution.

Website owners are willing to show ads on their websites for revenue and companies are willing to pay to display ads of their financial products (Bonds, IPO, etc) on websites. The purpose of an Adserver is to match ads to sites. We want you to develop a mini-adserver program which returns the most appropriate financial products ad for a given domain website. The Adserver program would be given two input files.

|  |  |
| --- | --- |
| Input file 1: sites.csv  These are the sites which the adserver recognizes. Each line in the file describes a website domain.   1. The first field is a 32-bit unsigned integer which is the unique identifier for the site(also referred as site\_id) 2. The second field is an unsigned 32-bit integer which is the minimum amount in cents(also referred to as the reserve\_price) the site demands for displaying an ad 3. The third field is a string that is the site’s URL (referred to as site\_url).  An example follows: | Input file 2: ads.csv  The ads which the adserver considers. Each line in the file describes an ad.   1. The first field is a 32-bit unsigned integer which is the unique identifier for the ad (referred to as ad\_id) 2. The second field is a 32-bit unsigned integer which is the maximum amount in cents the ad is willing to pay to show on a site(referred to as bid\_price) 3. The third field is a 32-bit integer that specifies the number of sites the ad wants to display on, followed by the site\_ids of the sites. An example follows: |
| SiteID, ResAmt, SiteUrl  23181, 320, abc.com  34288, 450, pqr.com  98662, 567, xyz.com  22675, 721, lmn.com  66434, 500, rst.com  86123, 5000, jjj.com | AdID, BidAmt, NumSites, Site1, Site2, Site3  9822, 450, 3, 23181, 98662, 66434  3421, 897, 3, 22675, 98662, 34288  8961, 342, 1, 98662, ,  7623, 2000, 3, 23181, 22675, 98662 |
|  |  |

In the above example, ad 9822 is willing to play on 3 websites (abc.com, xyz.com and rst.com) and can pay a maximum amount of 450 cents

For a given site, the adserver decides the appropriate ad by applying the following rules

1. An ad can be shown on this site only if the site is in the list of sites that the ad is interested in. If there are no ads for a given website, then no ad is returned.
2. The ad that is returned is chosen using second price auction. For ads to qualify for the second price auction for a site their bid\_price should be greater than or equal to the reserve\_price of the site. The winner of the auction is the ad that has the maximum bid\_price among these ads, and this winning ad pays the second highest ad’s bid\_price. For instance, if two ads with bid\_price 500 and 600 are competing for a site that has a reserve\_price of 400, then the ad that bid 600 wins, but pays 500 (the second highest ad’s bid price) to the site.
3. In case of a tie between two or more ads, the ad with higher value of number of sites to display (NumSites) wins and pays its bid\_price. In case, the auction has only one ad that qualifies, then that ad wins and pays reserve\_price of that site.
4. In case there are no ads that qualify for the auction for a site - either because no ad expresses interest in playing on that site or because none of the ads have bid\_price greater than or equal to reserve\_price of that site then no ad is returned.

OUTPUT:

Your adserver should accept the input file paths (first is the sites.csv and the second is the ads.csv) as arguments and wait for input. The input is the site\_id and the adserver should return the ad\_id of the ad that wins the second price auction and price the ad pays for the display. An input of -1 ends the program. For example,

$ ./adserver sites.csv ads.csv

22675

7623 897

34288

3421 450

66434

0 0

99999

0 0

86123

0 0

-1

$