

## Budgeting a Supercomputer

---

A Supercomputer can beat any ordinary computer in the processing-speed. In general, the speed of a Supercomputer is measured in “FLOPS” (Floating Point Operations Per Second). For simplicity, we assume that a Supercomputer is made up of 4 major components (e.g. A,B,C,D) where the processing speed of the Supercomputer is the smallest processing speed of all of its major components.

Suppose, you are given some data about the prices and the processing speeds of various types of those major components which exist in the market. The table of the collected data is as follows.

$A_1(p_1, s_1)$	$A_2(p_2, s_2)$	$A_3(p_3, s_3)$	$A_4(p_4, s_4)$	$A_5(p_5, s_5)$
$B_1(p_1, s_1)$	$B_2(p_2, s_2)$	$B_3(p_3, s_3)$		
$C_1(p_1, s_1)$	$C_2(p_2, s_2)$			
$D_1(p_1, s_1)$	$D_2(p_2, s_2)$	$D_3(p_3, s_3)$	$D_4(p_4, s_4)$	

The first row denotes that there exist  $i = \{1, 2, \dots, 5\}$  types of the component A in the market. The price and the processing speed of the  $i$ -th type of the component A is denoted by  $A_i(p_i, s_i)$ . The similar explanation holds for the other rows of the table.

You have to buy exactly one type of each major components. Your total budget is limited. Now you are wondering what is the fastest operating speed you can achieve.

For example consider following scenario

$A_1(1010, 20)$	$A_2(1050, 30)$	$A_3(900, 10)$	$A_4(2000, 17)$	$A_5(1800, 35)$
$B_1(100, 25)$	$B_2(200, 30)$	$B_3(200, 27)$		
$C_1(1000, 21)$	$C_2(1100, 25)$			
$D_1(700, 22)$	$D_2(900, 26)$	$D_3(1000, 31)$	$D_4(1001, 35)$	

Now if your budget is 2830 then maximum speed that can be achieved is 20 by buying  $A_1, B_1, C_1, D_1$ , but if your budget is 2860 then the speed you can achieve is 21 by buying  $A_2, B_1, C_1, D_1$ .

### Input

The input consists of several test cases. Each test case describes the prices and the processing speeds of various types of components. The first line of

each test case contains a single integer  $1 \leq n \leq 10000$  that indicates the number of types of component A. The next  $n$  lines contain pairs of integers that represent the prices and processing speeds of the  $n$  types of the component A. Subsequent lines are for other three components. Let  $p$  be the price and  $s$  be the processing speed of any component. Then,  $1 \leq p, s \leq 5 \times 10^8$ . The last line of a test case consists a number  $b \leq 10^9$

The input is terminated by the number zero on a line by itself.

## Output Specification

For each of the cases in the input file, print one line which contains the maximum processing speed achievable. You may assume that it will always possible to buy a supercomputer with your available budget.

## Sample Input

```
5
1010 20
1050 30
900 10
2000 17
1800 35
3
100 25
200 30
200 27
2
1000 21
1100 25
4
700 22
900 26
1000 31
1001 35
2830

5
1010 20
1050 30
900 10
2000 17
1800 35
```

3  
100 25  
200 30  
200 27  
2  
1000 21  
1100 25  
4  
700 22  
900 26  
1000 31  
1001 35  
2860

5  
1010 20  
1050 30  
900 10  
2000 17  
1800 35

3  
100 25  
200 30  
200 27  
2  
1000 21  
1100 25  
4  
700 22  
900 26  
1000 31  
1001 35  
1000000000

0

## Sample Output

20  
21  
25

## Remarks

**Algorithm** Use binary search on solution and greedily choose best component to find cost for given processing. Expected time complexity is  $O(n \log(10^9))$ .

**Input Output Explanation** In the sample input given, there are three cases. The cases are separated with new line. However, the last line of input is 0. Obviously, you should not process this as a case, rather just exit from your program. So your code should take input of cases in a loop and process them. The first two sample are the example described. It is required to follow the input output format exactly. Otherwise you will not get the mark of passing the judge data.

```
while(scanf("%d",&an)==2)
{
    if(an==0) break;

    // some initialization here...

    //taken input

    // .... process and print output
}
```

---

Tanaeem M Moosa  
tanaeem.moosa@csebuuet.org