First problem：

Problem description: (简单堆操作)

Robin likes sugar very much. Therefore, Robin has an extremely large sugar pot. He has two operations to the pot. One is to put a sugar into the pot. The other one is to pick the heaviest sugar from the pot. Now, you are given the operation sequence of Robin and your mission is to output the weight of the heaviest sugar each time Robin picks.

Input:

There will be n (0 < n < 1, 000, 000) lines representing the operation sequence of Robin. Each line there will be two integers A and B. If A equals 0, then it means Robin puts a sugar into the pot and B is the weight of the sugar. If A equals 1, then it means Robin pick up the heaviest sugar from the pot and B is a random number.

Output:

Each time Robin picks up a sugar, you should output the weight of that sugar.

Sample input:

0 10

1 1

0 1

0 2

0 3

1 2

1 3

Sample output:

10

3

2

Second problem:

Problem description: (栈操作，加上取最小值)

Try to simulate the operation of the stack structure. There are three operations: Pop, Push and Min. Pop and Push operation is just the original operation of a stack. Min operation is to output the present minimum element in the stack. When it comes to Pop and Min operation, there will always be elements in the stack.

Input:

There are n (0 < n < 1, 000, 000) lines representing the stack operations. Each line there will be two integers A and B. If A equals 0, then it means pushing an integer B into the stack. If A equals 1, then it means popping the top integer from the stack and B is a random number. If A equals 2, then you should output the minimum element in the stack (you do not need to pop this element) and B is a random number.

Output:

Output the minimum element when the operation is Min.

Sample input:

0 1

0 2

0 3

1 3

2 3

2 4

Sample output:

1

1

Third problem：

Problem description: (单源最短路，加第二权重)

There are n cities connected by m roads. Each road has a distance D and a cost V. Suppose Road X connects city A and city B. Then the distance between A and B is X.D and the cost between A and B is X.V. Now, Robin wants to travel from city S to city T. He hopes that distance he walks through is the shortest one. If the two distances are the same, then he wants the cost is the least one. Could you help him calculate the shortest distance from city S to city T and the minimum cost along with the shortest distance?

Input:

The first line is composed of two integers n and m (0 < n <= 1,000, 0 < m < 1, 000, 000). In the following m lines, each line is composed of four integers A B D V meaning the distance between A and B is D and the cost between A and B is V (0 < A, B <= n).

Output:

Only one line with two integers D and V. D is the shortest distance from S to T (S = 1 and T = n). V is the minimum cost along with the shortest distance.

Sample input:

4 5

1 2 1 3

2 4 2 2

1 3 2 2

3 4 1 2

1 4 4 1

Sample output:

3 4

Forth problem.

Problem description: (简单二维动态规划)

There is a convex polygon with N vertices (code from 1 to N). Every vertex has a weight. Your mission is to divide the convex polygon into N-2 non-intersect (不相交) triangles. There are many ways to do the division. However, you need to find a way to divide the convex polygon with the minimum sum of the product of the triangle vertices.

Input:

There is an integer N (3 <= N <=50) on the first line representing the number of the convex polygon vertex.

There are N integers on the second line meaning the weight of each vertex (from 1 to N). The weight of vertex is no larger than 32768.

Output:

The minimum value of the sum of the product of the triangle vertices.

Sample input:

5

121 122 123 245 231

Sample output:

12214884