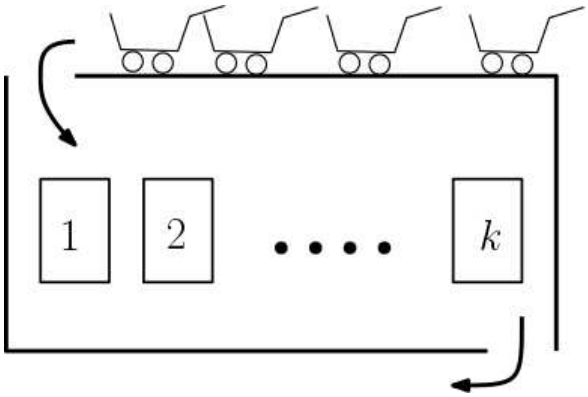
몰(Mall)

[문제] 대형 쇼핑몰에서 고객은 구매할 물건을 쇼핑 카트(cart)에 담아 계산을 위하여 준비된 계산대로 간다. 이 쇼핑몰에는 아래 그림과 같이 k개의 계산대가 병렬로 배치되어 있다. 그런데 사람들은 보통 가장 짧은 줄로 가는 경향이 있다. 그러나 짧은 줄이라고 하더라도 한 카트에 산더미 같이 많은 물건이 있다면 도리어, 이 "짧은 줄"에 서는 것은 더 불리할 경우가 있다.

이런 상황을 해결하기 위한 AI 시스템을 개발하고자 한다. 이 시스템은 각 고객의 카트에 담긴 물건의 갯수까지를 고려하여 각 고객이 가장 빠르게 쇼핑몰을 나갈 수 있도록 특정 계산대를 지정해준다. 즉 현 상황에서 각 고객이 가장 빠르게 나갈 수 있는 계산대를 지정해준다. 즉 여러 계산대 줄에 선 고객의 수와 각 카트 속 물건을 기준으로 가장 빨리 나갈 줄을 지정한다.

만일 대기 계산 시간이 같을 경우에는 계산대 번호가 더 빠른 쪽을 우선하여 지정한다. 예를 들어 3번과 7번, 이 두 계산대에서 기다리는 시간이 같다면 출구 쪽에 더 가까운 3번을 지정한다. 일종의 tie-break rule이다. 만일 두 고객이 두 계산대에서 동시에 마친 경우에는 출구에 가까운 k번 계산대 쪽 고객부터 먼저 쇼핑몰을 빠져나간다. 고객이 계산하는 시간은 물건 종류에 상관없이 동일하게 모두 t초(seconds)가 걸린다고 가정한다.



여러분에게는 k개의 계산대가 준비된 쇼핑몰에서 계산을 위하여 대기 중인 N명 고객의 정보가 [id , 카트 물건의 수]의 형식 record로 주어진다. 여러분은 이 정보를 바탕으로 앞서 설명한 AI 시스템을 활용할 때, 쇼핑몰을 완전히 빠져나오는 고객의 순서를 계산해야 한다. 단 대기중인 고객의 계산대로 갈 때 걸리는 시간은 0, 즉 없는 것으로 계산한다.

[입출력] 입력파일 **stdin** 첫줄에는 2개의 정수 'N k'가 주어진다. N은 계산대 배정을 위하여 기다리

고 있는 고객의 수이며 k는 계산대의 개수이다. 단 $4 \le N \le 100$, $2 \le k \le 10$ 이다. 이어지는 N개의 줄에는 각 고객의 회원 번호 u_i와 카트 속 물건의 갯수 w_i가 2개의 정수로 주어진다.

쇼핑몰 회원 번호 u_i는 유일하며(unique), u_i 카트에 담긴 상품 수는 $1 \le w_i \le 20$ 이다. 같은 시각에 서로 다른 계산대에서 동시에 결재를 마치는 경우라면, 쇼핑몰을 나가는 순서는 **계산대 번호가 큰 순서로 결정된다.** 예를 들어 4번, 6번, 9번 계산대에서 같은 시간에 계산을 마친다면, 나가는 회원의 순서는 9번, 6번, 4번 계산대의 순으로 결정된다. 여러분은 주어진 N명 회원의 구매 정보로부터 쇼핑몰을 완전히 빠져나가는 회원들의 순서를 **stdout** 파일의 N개 줄에 하나씩 순서대로 출력해야 한다.

[예제] 10명의 고객이 $k=3$개의 계산대 $\{C_1, C_2, C_3\}$가 준비된 쇼핑몰에서 대기 중이다. 시작할 때 3개의 계산대 모두가 비어있으므로 123은 C_1, 21은 C_2, 34는 C_3로 배정된다. 이를 Python list로 나타내면 다음과 같다. 각 t별 C_i의 상황은 다음과 같다. 박스에 표시된 (ui, wi)은 제일 앞에 선 고객을 의미한다. 그 뒤에 줄을 쭉 서있는 상황이다.

t	C1	C2	C3	Exct
0	∅	∅	∅	∅
1	123(4)	21(5)	34(14)	
4	123(4)	21(5)	34(14)	
5	56(1)	21(5)	34(14)	123
6	45(7)	723(5)	34(14)	21, 56
7	45(7)	723(5)	34(14)	
...	45(7)	723(5)	34(14)	
11	45(7)	55(7)	34(14)	723
12			34(14)	
13			34(14)	45
14			34(14)	
15				34

stdin	stdout
10 3 //N,k 123 4 21 5 34 14 56 1 45 7 723 5 55 7 13 5 910 10 73 3	123 21 56 723 45 34 55 13 73 910

위 도표에서는 C_i에 한 고객만 표시했는데, 기다리는 고객의 t 시점 이후에 대기 계산대에서 기다리고 있는 고객들의 카트에 담긴 물건의 수가 제일 적은 쪽으로 줄을 서면 된다. 그리고 계산이 끝난 고객을 바로 바로 exit 시켜서 각 queue를 정리해야 한다. 즉 전체 쇼핑몰의 시계가 t=0부터 $t=t+1$로 진행하도록 하고, 계산을 한 고객은 각 계산대 Queue에서 제거(?)한다. 단 tie-break를 조심해야 한다.

[제한조건] 프로그램의 이름은 **mall.{c,cpp,py}**이다. 이번 문제는 어렵지는 않지만 token 수를 맞추려면 STL queue를 이용하여 compact하게 작성해야 할 것이다.