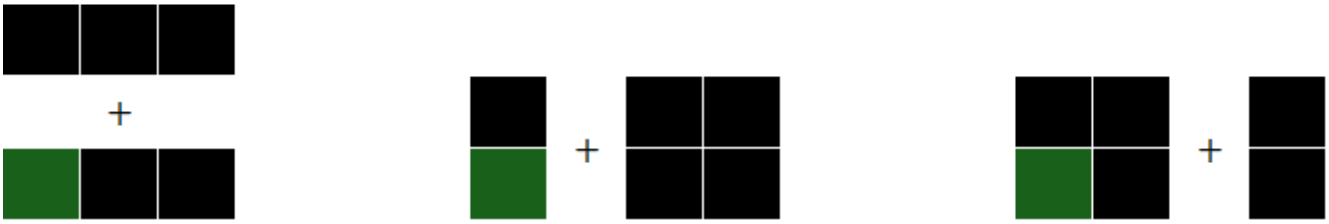
Problem C. Chocolate

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

有一块$n \times m$的巧克力，恰好有一个1×1的小块长了绿毛，为了公平分配，你需要进行一场博弈，每次对于一块巧克力，可以沿着行或者列掰开，即对于如图一块2行3列的巧克力，可以有3种掰法。



每次掰下后，将不含坏巧克力的部分分配给自己，含有坏巧克力的部分给对方继续掰，两人轮流进行，由你先手，直到某一方拿到只有1×1的巧克力(当然是剩下的那块坏的)为输。

本题是交互题，获得AC的条件有且仅有一条：让对方拿到最后的坏巧克力。

Input

初始仅包含一行4个正整数$n\ m\ x\ y$，表示巧克力初始有n行m列，且坏的1×1小块的位置为从上到下第x行，从左到右第y列，保证您有至少一种策略可以在任意情况下都不吃到坏的。

$(1 \le x \le n \le 16,\ 1 \le y \le m \le 16)$

对于您的每一次输出，判题程序会给出它的下一次切割结果，使用通常方式读入即可，交互格式在输出描述中。

Output

您有两种与判题程序交互的格式:

- 输出一行"$r\ x$"(不含双引号)，表示在第x行和第$x+1$行之间进行切割
- 输出一行"$c\ y$"(不含双引号)，表示在第y列和第$y+1$列之间进行切割

其中$1 \le x <$当前巧克力剩余行数，$1 \le y <$当前巧克力剩余列数。

在每一次输出中确保带有换行符'\n'，并且之后立即刷新缓冲区，不同语言的刷新方式见Note

之后判题程序会在您的切割基础上进行它的切割，并将含有坏块的巧克力按与输入相同的格式返回，4个正整数$n\ m\ x\ y$，表示巧克力在切割后剩余n行m列，且坏的1×1小块的位置为从上到下第x行，从左到右第y列，只需要使用正常的读入方式进行读入即可。

特别地，为了方便，如果经过您的切割后，巧克力只剩下1×1的大小，判题程序会返回"0 0 0 0"，您在读到这样的返回时，可以在读入后直接结束程序，并获得Accepted。

Sample

standard input	standard output
2 3 2 2 1 2 1 2 0 0 0 0	r 1 c 1

Note

常用语言刷新缓冲区方式:

- C/C++: fflush(stdout)
- Java: System.out.flush()
- Python: stdout.flush()

交互格式举例:

- C/C++:
  - printf("r %d\n" ,x);//表示要在第x行与第$x+1$行之间切割
  - fflush(stdout);//刷新缓冲区
  - scanf("%d%d%d%d" ,&n,&m,&x,&y);//获取判题程序切割后的结果
  - $\cdots$(重复上述过程，直到胜利)
- Java:假设已有Scanner sc=new Scanner(System.in);
  - System.out.println("r "+x);//表示要在第x行与第$x+1$行之间切割
  - System.out.flush();//刷新缓冲区
  - n=sc.nextInt();m=sc.nextInt();x=sc.nextInt();y=sc.nextInt();//获取判题程序切割后的结果
  - $\cdots$(重复上述过程，直到胜利)
- Python:
  - print("r" ,x)#表示要在第x行与第$x+1$行之间切割
  - stdout.flush()#刷新缓冲区
  - n,m,x,y=map(int,input().split())#获取判题程序切割后的结果
  - $\cdots$(重复上述过程，直到胜利)

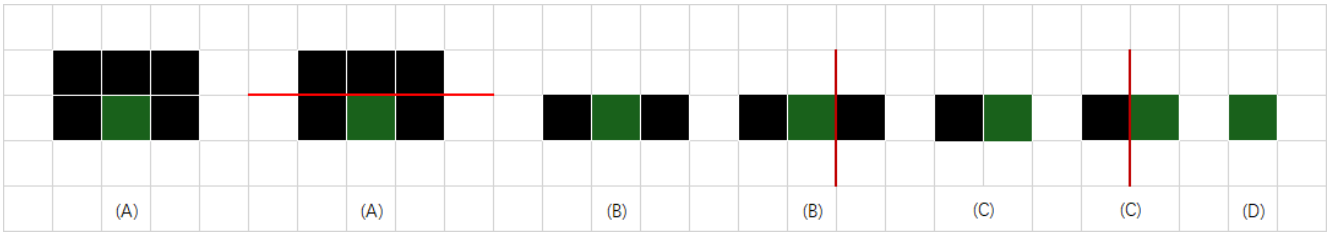
样例解释:

Sample Input中，只有第一行是初始您能够读到的$n\ m\ x\ y$，后续的都是交互过程中产生的

1. 用户程序通过输入得知初始巧克力为2行3列，坏块在第2行第2列
2. 用户做出第一次决策：从第1行和第2行中间切开(剩余1行3列的巧克力，坏块在第1行第2列)

3. 用户程序输出"r 1"，换行后刷新缓冲区
4. 判题程序做出决策：从第2列和第3列中间切开(剩余1行2列的巧克力，坏块在第1行第2列)
5. 判题程序将结果返回给用户程序"1 2 1 2"
6. 用户程序读入结果，得知剩余巧克力的情况，做出决策"c 1"并输出
7. 判题程序输了，输出"0 0 0 0"
8. 用户程序读入结果得知已经赢了，直接结束程序

切割过程如下图：



交互题的错误结果有多种可能，例如Wrong Answer可能代表交互格式不对(例如输出的x和y超出范围)，也可能代表交互正常但是最后您输了，TimeLimit可能表示忘记刷新缓冲区，或者在您的程序的决策过程中出现了死循环等情况，造成判题程序没有在时限内获得您的输出。