# OJ02746：约瑟夫问题, implementation

# <http://cs101.openjudge.cn/practice/02746>

while True:  
 n, m = map(int, input().split())  
 k = m  
 if n == 0:  
 exit()  
 lst = [i for i in range(1, n + 1)]  
 while len(lst) != 1:  
 del lst[k - 1]  
 n -= 1  
 k = (m + k - 2) % n + 1  
 print(lst[0])

# OJ02810: 完美立方, brute force

# <http://cs101.openjudge.cn/practice/02810>

n = int(input()) + 1  
for a in range(2, n):  
 for b in range(2, a):  
 for c in range(b, a):  
 for d in range(c, a):  
 if a \*\* 3 == b \*\* 3 + c \*\* 3 + d \*\* 3:  
 print("Cube = " + str(a) + ", Triple = (" + str(b) + "," + str(c) + "," + str(d) + ")")

# OJ04110: 圣诞老人的礼物, greedy

# <http://cs101.openjudge.cn/practice/04110>

n, m = map(int, input().split())  
dct, lst, ans = {}, [], 0  
for i in range(n):  
 v, w = map(int, input().split())  
 if lst.count(v / w) == 0:  
 dct[v / w] = w  
 lst.append(v / w)  
 else:  
 dct[v / w] += w  
lst.sort(reverse=True)  
for i in lst:  
 if m > dct[i]:  
 ans += dct[i] \* i  
 m -= dct[i]  
 else:  
 ans += i \* m  
 break  
print(int(ans \* 10 + 0.5) / 10)

# CF545C: Woodcutters, dp/greedy, 1500

# <https://codeforces.com/problemset/problem/545/C>

lstx, lsth, ans = [], [], 2  
for i in range(int(input())):  
 x, h = map(int, input().split())  
 lstx.append(x)  
 lsth.append(h)  
right = lstx[0]  
if len(lstx) == 1:  
 print(1)  
 exit()  
for i in range(1, len(lsth) - 1):  
 if lsth[i] < lstx[i] - right:  
 ans += 1  
 right = lstx[i]  
 elif lsth[i] < lstx[i + 1] - lstx[i]:  
 ans += 1  
 right = lstx[i] + lsth[i]  
 else:  
 right = lstx[i]  
print(ans)

# OJ16528：充实的寒假生活, cs10117 Final Exam, greedy

# <http://cs101.openjudge.cn/practice/16528>

dct, keylst, ans = {}, [], 1  
for i in range(int(input())):  
 start, end = map(int, input().split())  
 if keylst.count(start) == 0:  
 dct[start] = end  
 keylst.append(start)  
 else:  
 if dct[start] > end:  
 dct[start] = end  
keylst.sort()  
for i in range(len(keylst)):  
 for j in keylst[i + 1:]:  
 if dct[keylst[i]] >= dct[j]:  
 del dct[keylst[i]]  
 del keylst[i]  
for i in range(len(keylst)):  
 for j in keylst[i + 1:]:  
 if dct[keylst[i]] >= dct[j]:  
 del dct[keylst[i]]  
 del keylst[i]  
end = dct[keylst[0]]  
for i in keylst[1:]:  
 if i > end:  
 ans += 1  
 end = dct[i]  
print(ans)

这里好几次因为忘记把用来debug的print语句删除WA。。。我的主要思路是把区间预处理，删除大区间套小区间的情况，保留相交情况，再遍历一遍得结果。代码的具体问题是预处理不干净：左边有一段重复出现两次的代码就是为了处理这个问题。

排列倒是想清楚之后就一遍过了。

# OJ01833: 排列, math

# <http://cs101.openjudge.cn/practice/01833>

for i1 in range(int(input())):  
 n, k = map(int, input().split())  
 lst = [int(i) for i in input().split()]  
 for i2 in range(k):  
 i3 = n - 1  
 while lst[i3 - 1] > lst[i3]:  
 i3 -= 1  
 if i3 == 0:  
 lst = [i for i in range(1, n + 1)]  
 break  
 if i3 != 0:  
 temp = lst[i3 - 1:]  
 temp.sort()  
 index = temp.index(lst[i3 - 1]) + 1  
 t = temp[index]  
 del temp[index]  
 lst = lst[:i3 - 1] + [t] + temp  
 for i2 in lst[:-1]:  
 print(i2, end=" ")  
 print(lst[-1])