'''

The laser show at the Lumbini Park is something not to be missed.

But, as per govt rule they have to follow COVID-19 restrictions.

The management planning to guide the audience to sit in the seats

that maximizes the distance to the closest person, in order to

practice the social distance in the auditorium.

Please help usher to guide the audience to sit in a seat by following few rules:

- There are N seats in a single row, seats are numbered from 0 to N-1.

- Maximize the distance from person to the closest person.

- If there are multiple seats with similar distance, they sit in the seat with the lowest number.

- First person always sit in seat number 0.

- If a person leaves the auditorium from a seat, please consider that the seat is vacant

Create a class Auditorium and two functions in it.

1. int seat(): represent the seat number of audience to sit.

2. void leave(int s): person leaves the auditorium from a seat number 's'.

Input Format:

----------------

Line-1 -> two integers N and P, Number of seats N, Number of Operations P

P lines of input -> each line contains funtion number and parameter list (if required).

Output Format:

------------------

Print the alloted seat numbers in one line as output.

Sample Input-1:

-------------------

10 6

1

1

1

1

2 4

1

Sample Output-1:

---------------------

0 9 4 2 5

NOTE:

-----

In the sample input:

- option 1 indicates, calling "int seat()" method.

- option 2 indicates, calling "void leave(seat\_num)" method.

'''

Only 50 percent test cases passed

n,s=list(map(int,input().split()))

l=[]

flag=[False for i in range(n)]

# ll=[]

for i in range(s):

l1=list(map(int,input().split()))

l.append(l1)

for i in l:

if(i[0]==1):

ll=[]

avg=-1

a=0

for k in range(len(flag)):

if(flag[k]==True):

ll.append(k)

# print(ll)

if(len(ll)==0):

flag[0]=True

print(0,end=" ")

elif (len(ll)==1 and flag[0]==True):

flag[len(flag)-1]=True

print(len(flag)-1,end=" ")

else:

dis=0

if(ll[0]>0):

a=(0+ll[0])//2

if(abs(ll[0]-a)>avg):

dis=abs(ll[0]-a)

avg=a

for k in range(len(ll)-1):

a=(ll[k]+ll[k+1])//2

if(abs(ll[k]-a)>avg):

avg=a

dis=abs(ll[k]-a)

if(ll[len(ll)-1]<len(ll)):

a=((len(ll)-1)+ll[len(ll)-1])//2

if(abs(ll[n-1]-a)>avg):

avg=a

dis=abs(ll[n-1]-a)

flag[avg]=True

print(avg,end=" ")

elif(i[0]==2):

flag[i[1]]=False

Gachibowli Diwakar is a tremendous bowler in Indian Cricket Team.

Diwakar has played in N Series of Games, and in each series he has taken some

number of wickets. Now, Gachibowli Diwakar's wickets count is arranged in

ascending order which is a non-negative integer. Create a method to generate

the Gachibowli Diwakar's Wicket-index as W-index

According to the definition of W-index, Gachibowli Diwakar has index W, if W of

his 'N' series have at least W wickets each, and the other 'N-W' series have

no more than W wickets each.

Note:

-----

If there are several possible values for W, the maximum one is taken as the W-index.

Input Format:

-------------

Line-1: An integer N, number of series played by Diwakar.

Line-2: N space separated integers, number of wickets in ascending order.

Output Format:

--------------

Print an integer, value of index -W.

Sample Input-1:

---------------

6

0 1 3 4 5 6

Sample Output-1:

----------------

3

Explanation:

------------

[0, 1, 3, 4, 5, 6] means the Gachibowli Diwakar has 6 series in total and

in each series he had taken 0, 1, 3, 4, 5, 6 wickets respectively.

Since Gachibowli Diwakar has 3 series with at least 3 wickets each and the

remaining three with no more than 3 wickets each, his W index is 3.

Sample Input-2:

---------------

10

1 3 7 10 12 20 21 24 26 32

Sample Output-2:

----------------

7

import java.util.\*;

public class Main{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int[] arr=new int[n];

for(int i=0;i<n;i++){

arr[i]=sc.nextInt();

}

int l=0;

int u=n-1;

// int mid=0;

while(l<=u){

int mid=l+((u-l)/2);

if(arr[n-mid]>=mid){

l=mid+1;

}

else{

u=mid-1;

}

}

System.out.println(u);

}

}

Basanthi interested playing with digits.

He wants create a set of integers of length N, using the digits[0-9].

The rules to create the integers are as follows:

- digits in each integer are like d0,d1,d2...dn-1

- and |d0-d1| = |d1-d2| = |d2-d3| ... |dn-2 - dn-1|= D

Basanthi is given two integers N and D, where N is length of the integer and

D is the difference. Can you help Basanthi, to create such list of integers

and print all the possible integers in ascending order

Note:

-----

Integers with leading 0's are not allowed

Input Format:

-------------

Two space separated integers N and K.

Output Format:

--------------

Print all the possible integers in ascending order.

Sample Input-1:

---------------

3 6

Sample Output-1:

----------------

[171, 282, 393, 606, 717, 828, 939]

Sample Input-2:

---------------

2 3

Sample Output-2:

----------------

[14, 25, 30, 36, 41, 47, 52, 58, 63, 69, 74, 85, 96]

import java.util.\*;

public class Main{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int d=sc.nextInt();

ArrayList<Integer> res=new ArrayList<>();

backtrack(n,d,0,0,res);

System.out.println(res);

}

public static void backtrack(int n,int k,int num,int ctr,ArrayList<Integer> res){

if(ctr==n){

res.add(num);

return;

}

for(int i=0;i<=9;i++){

if(num==0 && i==0){

continue;

}

else if(num==0 || Math.abs((num%10)-i)==k){

num=num\*10+i;

backtrack(n,k,num,ctr+1,res);

num=num/10;

}

}

}

}

Venkat wants to create a unique name for his child, to do that he is referring

two names, one is traditonal name(TN) and other is modern name(MN)

He is planning to create unique name(UN), using the following ways:

- if traditional name TN is non empty, then add the first letter L of TN

to unique name UN and remove the letter L from traditional name TN

e.g., if TN = "havi" and UN="anvika", then after adding L to UN and remove L

from TN, the names updated as UN="hanvika", TN="avi".

- if modern name MN is non empty, then add the first letter L of MN

to unique name UN and remove the letter L from modern name MN

e.g., if MN = "ram" and UN="ao", then after adding L to UN and remove L

from MN, the names updated as UN="rao", MN="am".

You are given two names, TN and MN,

Your task is to help venkat to generate the unique name for his child,

and the name should be largest in the dictionary order.

For example, "kmit" is larger than "kmec", as third letter is greater in "kmit".

Input Format:

-------------

Two space separated names, TN and MN.

Output Format:

--------------

Print a string, the laregst unique name possible.

Sample Input-1:

---------------

sudha vivid

Sample Output-1:

----------------

vsuividhda

Sample Input-2:

---------------

appa banana

Sample Output-2:

----------------

bappananaa

import java.util.\*;

public class Main{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

String s1=sc.next();

String s2=sc.next();

int i=0;

int j=0;

StringBuilder un=new StringBuilder("");

while(i<s1.length() && j <s2.length()){

if(s1.charAt(i)>s2.charAt(j)){

un.append(s1.charAt(i));

// System.out.println(un);

i+=1;

}

else if(s1.charAt(i)<s2.charAt(j)){

un.append(s2.charAt(j));

// System.out.println(un);

j+=1;

}

else {

int k = i + 1;

int l = j + 1;

while (k < s1.length() && l < s2.length() && s1.charAt(k) == s2.charAt(l)) {

k++;

l++;

}

if(k==s1.length() && l!=s2.length()){

un.append(s2.charAt(j));

j++;

}

else if(l==s2.length() && k!=s1.length()){

un.append(s1.charAt(i));

i++;

}

else if(k==s1.length() && l==s2.length() ){

un.append(s2.charAt(j));

j++;

}

else if(s1.charAt(k) > s2.charAt(l)){

un.append(s1.charAt(i));

i++;

}

else if(s2.charAt(l)> s1.charAt(k) ){

un.append(s2.charAt(j));

j++;

}

}

}

if(i<s1.length()){

while(i<s1.length()){

un.append(s1.charAt(i));

i+=1;

}

}

if(j<s2.length()){

while(j<s2.length()){

un.append(s2.charAt(j));

j+=1;

}

}

System.out.println(un);

}

}