| <pre>1&lt;=N&lt;=K&lt;=10^8 Sample Input: 521 Sample Output: 2  def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))</pre>   | 0.5  | '5' ~\v                             | Logo                | 'ɔ'         | <u>~</u>  | - ۱۰                 | .0             |
|--|--|-------------------------------------|---------------------|-------------|-----------|----------------------|----------------|
| EXPERIMENT Title  CANDIES  CANDIES  Let's consider a scenario where there are K candies to be distributed among N children, each uniquely numbered from 1 to N. The distribution commences with Child A followed by a sequential allocation to the subsequent children in the order: A, A+1, A+2, N. The query at hand is to identify which child will be the last recipient of a candy. In more explicit terms, after Child x (where 1-s x x N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child Necesiva a candy, the distribution cycle restarts, and Child 1 becomes the next recipient.  The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution.  Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  I <= N=K=K=10*8  Sample Input:  5 2 1  Sample Output:  2  Source Code:  Source Code:  A Example usage:  N, K, A = app(int, input().strip().split())  print(last, candy, recipient(N, K, A))   | AO 3BR   | 3clove STI                          | UDENT REP           | ORT         | aBR13CV   | ONO 3BR              | 3CD040         |
| Roll Number  38R23CD040  EXPERIMENT  Title  CANDIES  Let's consider a scenario where there are K candies to be distributed among N children, each uniquely numbered from 1 to N.  The distribution commences with Child A followed by a sequential allocation to the subsequent children in the order: A A+1,  A+2, N. The query at hand is to identify which child will be the last recipient of a candy.  In more explicit terms, after Child x (where 1 < x × N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child Necetiving a candy, the distribution cycle restarts, and Child 1 becomes the next recipient.  The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution.  Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1-e-Nc=Kc=10^8  Sample Input:  5 2 1  Sample Output:  2  Source Code:  # Example usage:  N, K, A = map(int, input().strip().split())  print(1281, candy_recipient(N, K, A))  | 273000   | 58R2 5                              | 22300               | 500AC       |           | 5° 58°               | V              |
| Roll Number  3BR23CD040  EXPERIMENT  Title  CANDIES  Lets consider a scenario where there are K candies to be distributed among N children, each uniquely numbered from 1 to N. The distribution commences with Child A, followed by a sequential allocation to the subsequent children in the order: A, A+1, A+2,, N. The query at hand is to identify which child will be the last recipient of a candy. In more explicit terms, after Child x, (where 1<6 x < N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child Necetiving a candy, the distribution cycle restarts, and Child 1 becomes the next recipient. The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution. Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1-eNe=K<=10*8  Sample Input:  5 2 1  Sample Output:  2  Source Code:  N, K, A = map(int, input().strip().split()) print(last, candy_recipient(N, K, A))  # Example usage: N, K, A = map(int, input().strip().split()) print(last, candy_recipient(N, K, A))  | DETAILS 3000   | 38273° 04038                        | , 3cDoAs            | 382230      | 040 3B'   | 03CDOM               | 38273          |
| Roll Number  38R23CD040  EXPERIMENT  Title  CANDIES  Lets consider a scenario where there are K candies to be distributed among N children, each uniquely numbered from 1 to N. The distribution commences with Child A, followed by a sequential allocation to the subsequent children in the order: A, A+1, A+2,, N. The query at hand is to identify which child will be the last recipient of a candy. In more explicit terms, after Child x (where 1<0 x < N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child N receiving a candy, the distribution cycle restarts, and Child 1 becomes the next recipient.  The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution.  Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  1:=N=K<=10*8  Sample Input:  5 2.1  Sample Output:  2  Source Code:  N, K, A = map(int, input().strip().split()) print(last, candy_recipient(N, K, A))  # Example usage: N, K, A = map(int, input().strip().split()) print(last, candy_recipient(N, K, A))   | Nama   | CDO40 5 223CD                       | 3BR1                | 0040        | 23cD°     | 3BRIV                | 040 s          |
| Roll Number  3BR23C0040  EXPERIMENT  Intel  CANDIES  Description  Let's consider a scenario where there are K candies to be distributed among N children, each uniquely numbered from 1 to N. The distribution commences with Child A, followed by a sequential allocation to the subsequent children in the order: A, A+1, A+2, N The query at hand is to identify which child will be the last recipient of a candy, In more explicit terms, after Child x (where 1<= x < N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child N receiving a candy, the distribution cycle restarts, and Child 1 becomes the next recipient.  The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution. Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1 <a href="I &lt;= N=&lt;= 10^8 Sample Input: 52 1">I &lt;= N=&lt;= 10^8 Sample Input: 52 1</a> Sample Output:  2  Source Code:  N K, A = map(int, input().strip().split()) print(last, candy_recipient(N, K, A):    I = St_maple usage: N, K, A = map(int, input().strip().split()) print(last, candy_recipient(N, K, A))    I = St_maple usage: N, K, A = map(int, input().strip().split()) print(last, candy_recipient(N, K, A))   | KALYANI  | 39                                  | 236                 | 30          | 00k       |                      | 201            |
| EXPERIMENT  Title  CANDIES  Description  Let's consider a scenario where there are K candies to be distributed among N children, each uniquely numbered from 1 to N. The distribution commences with child A, followed by a sequential allocation to the subsequent children in the order. A, A+1, A+2, N. The query at hand is to identify which child will be the last recipient of a candy. In more explicit terms, after child x (where 1-8 x x N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child N receiving a candy, the distribution cycle restarts, and Child 1 becomes the next recipient. The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution. Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1 <= N <= K <= 10^3 8  Sample Input:  5 2 1  Sample Output:  2  Source Code:  W  | Roll Number  | 223                                 |                     | 2873        | 30        | 20 n                 |                |
| EXPERIMENT Title  CANDIES  Let's consider a scenario where there are K candies to be distributed among N children, each uniquely numbered from 1 to N. The distribution commences with Child A, followed by a sequential allocation to the subsequent children in the order: A, A+1, A+2,, N. The query at hand is to identify which child will be the last recipient of a candy. In more explicit terms, after Child x (where the child x) where the child x (where the child x) when the child will be the last recipient of a candy. In primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution. Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1 c=N<=K<=10*8  Sample Input:  5 2 1  Sample Output:  2  Source Code:  Where the condition of the candy is the condition of the candy is granted to Child x +1. Upon Child x | )X   |                                     |                     |             | . 1x      | ~~                   |                |
| The distribution commences with Child A, followed by a sequential allocation to the subsequent children in the order: A, A+1, A+2,, N. The query at hand is to identify which child will be the last recipient of a candy. In more explicit terms, after Child x (where 1-e x < N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child N receiving a candy, the distribution cycle restarts, and Child 1 becomes the next recipient. The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution.  Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1<=N<=K<=10^*8  Sample Input:  5 2 1  Sample Output:  2  def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))   | ~ M  | O3BRIV                              | 22 <sup>2</sup> 3c0 | ,03BR1      | CDONO     | 227300               | ,03            |
| The distribution commences with Child A, followed by a sequential allocation to the subsequent children in the order: A, A+1, A+2,, N. The query at hand is to identify which child will be the last recipient of a candy. In more explicit terms, after Child x (where 1-e x < N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child N receiving a candy, the distribution cycle restarts, and Child 1 becomes the next recipient. The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution.  Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1<=N<=K<=10^*8  Sample Input:  5 2 1  Sample Output:  2  def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))   | Title CANDIES  | 3BR13CDOW 3CDOMO 3BR133C            | 13cD0A03b           | 3CDOW 3CDOW | 38R135    | 30 AO 38 P           | BCDOK          |
| The distribution commences with Child A, followed by a sequential allocation to the subsequent children in the order: A, A+1, A+2,, N. The query at hand is to identify which child will be the last recipient of a candy. In more explicit terms, after Child x (where 1-e x < N) receives a candy, the subsequent candy is granted to Child x+1. Upon Child N receiving a candy, the distribution cycle restarts, and Child 1 becomes the next recipient. The primary objective is to ascertain the identity of the child who will receive the last candy in this cyclic distribution.  Note: Each child receives only 1 candy.  Input Format:  The first line of input contains 3 space seperated integers N, K and A.  Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1<=N<=K<=10^*8  Sample Input:  5 2 1  Sample Output:  2  def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))   | Description, St. 250   | -000 3822° 223C10203                | 38823500            | OOAO BRAS   | 23CD040.3 | 38R13CDV             | ) N.<br>+1,    |
| Output Format:  Print the friend who will be the final recipient of the candy.  Constraints:  1<=N<=K<=10^8  Sample Input:  521  Sample Output:  2  def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))   | Note: Fach child recei   | is to ascertain the identity of the |                     |             |           | distribution.        | n<br>ે         |
| Print the friend who will be the final recipient of the candy.  Constraints:  1<=N<=K<=10^8  Sample Input:  521  Sample Output:  2  def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))   | The first line of input of   | contains 3 space seperated into     | egers N, K and A.   |             |           |                      | C.             |
| <pre>def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))</pre>  | Output Format:   |                                     |                     |             |           |                      |                |
| <pre>def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))</pre>  | Print the friend who w   | ill be the final recipient of the c | andy.               |             |           |                      | 0              |
| Sample Input:  521  Source Code:  action (N, K, A):  last_child = (A - 1 + K - 1) % N + 1  return last_child  # Example usage:  N, K, A = map(int, input().strip().split())  print(last_candy_recipient(N, K, A))  | d N. K. 1010   |                                     |                     |             |           |                      | ,01            |
| Source Code:  def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))   | Sample Input:  |                                     |                     |             |           |                      |                |
| <pre>def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))</pre>  |  |                                     |                     |             |           |                      | 3              |
| <pre>def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage:     N, K, A = map(int, input().strip().split())     print(last_candy_recipient(N, K, A))</pre>  | Sample Output:   |                                     |                     |             |           |                      |                |
| <pre>def last_candy_recipient(N, K, A):     last_child = (A - 1 + K - 1) % N + 1     return last_child  # Example usage: N, K, A = map(int, input().strip().split()) print(last_candy_recipient(N, K, A))</pre>  | 2  |                                     |                     |             |           |                      | .33            |
| <pre>return last_child  # Example usage: N, K, A = map(int, input().strip().split()) print(last_candy_recipient(N, K, A))</pre>  | Source Code:   | -8038gr 20040                       | -8F13CV             | AO3BR'      | achoko    | -8 <sup>273</sup> CV | AO 3BRI        |
| <pre># Example usage: N, K, A = map(int, input().strip().split()) print(last_candy_recipient(N, K, A))</pre>   | return last_chi  | A - 1 + K - 1) % N + 1              |                     |             |           |                      | o <sup>r</sup> |
| BE ON SCA SEE ON BEEN VE   | <pre># Example usage: N, K, A = map(int, print(last_candy_re</pre> | ecipient(N, K, A))                  |                     |             |           |                      | 8              |
|  | 362  | 8813C10A0 0A038R13CV                | 2cD0k0 38k.         | 23CDOAO     | 3BR13CV   | 38,382               | CDODONO        |

38th. 2000 53ch. 38ther 2000 . 5550 36