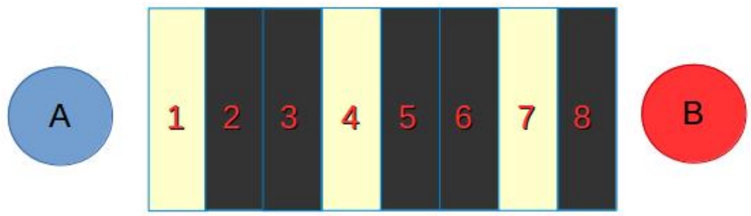
Problema a resolver:

El problema esta dado por la siguiente situación: nos encontramos en un extremo de un puente (afuera de él) y queremos llegar al otro extremo, bajo ciertas circunstancias y, si es posible hacer esto, hacerlo de la "mejor manera" (mas adelante se detallará qué significa esto y por qué podría no ser posible atravesar el puente). El puente está hecho con una cantidad n de tablones seguidos y, algunos de ellos pueden estar rotos. Cuando esto suceda, vamos a querer saltar estos tablones rotos al atravesar el puente, pisando siempre tablones sanos cuando estemos avanzando.
Tenemos una cantidad fija tablones seguidos que podemos saltar, la denominaremos c, así podemos ver que, si el puente tiene, en algún momento, una cantidad $k > c$ de tablones rotos **seguidos**, entonces claramente no tendremos manera de atravesarlo ya que, intuitivamente, podemos pensar que, en el mejor de los casos (donde mejor significa estar lo más alejados posibles del comienzo del puente) podríamos estar en el tablon sano anterior (anterior inmediato) al primero de esos k tablones rotos y, aún así no podríamos atravesar el puente ya que no podemos saltar más de c tablones seguidos, por lo tanto en cualquier otro caso (donde nos encontráramos en un tablón anterior al anterior inmediato del primero de los k rotos por ejemplo), estaríamos en una situación similar porque eventualmente llegaríamos al tablón sano que es el anterior inmediato al primero de los k y quedaríamos estancados de la misma manera.
En el caso en el que no se de la situación descripta anteriormente (es decir, en el caso en el cual sí podamos atravesar el puente), vamos a querer dar la menor cantidad de saltos (la "mejor manera").

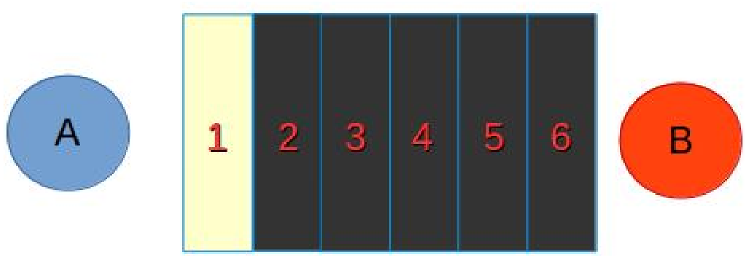
Presentamos algunos ejemplos graficos junto a sus soluciones y las secuencias que lo representan. Los circulos con el A y el B determinan el punto de partida y el punto de llegada, ambos fuera del puente. Los tablones rotos están pintados de negro.

En este ejemplo, $n = 10$ y el puente se escribe como antes 10 ahora n c 0 1 0 0 1 1 0 1 1 1.
Si $c = 3$, entonces la solución está dada por caer en los tablones 4 7 11. Notar que en realidad no existe un trablón numerado con el 11, pero cuando exista una solución, para indicar que llegamos al punto de llegada, diremos que

saltamos a un tablón mayor estricto que la cantidad de tablones del puente (o sea, que estamos efectivamente fuera del puente). Si tuvieramos el mismo puente pero con $antesnahorac = 2$, claramente no existiría una solución ya que, si bien podríamos llegar al tablón 7 sin problemas, una vez ahí no tendríamos manera de saltar el 8, 9 y 10 que están rotos.



Este otro puente se representa como 8 c 0 1 1 0 1 1 0 1.
Si $c = 2$ entonces la solución es 1 4 7 9.
Si $c = 1$ entonces no habría solución ya que hay 2 tablones rotos seguidos (esto ocurre dos veces en este caso particular pero con que ocurra una ya no hay solución posible).
Si $c = 3$ la solución es 4 7 9.



Finalmente introducimos este ultimo ejemplo del puente 6 c 0 1 1 1 1 1. Siguiendo la misma lógica que veníamos teniendo, podemos ver que este puente no tiene solución para $c < 5$. Si $c = 5$ entonces la solución es 1 7 y sí $c > 5$ la solución es 7.

Resolución:

La idea basicamente es ir recorriendo el puente (tablón por tablón) e ir almacenanando cual es el último tablón sano (lo llamaremos *ultimosano*) desde el comienzo del puente hasta el tablón de la iteración actual i (notar que *ultimosano* podría ser i), los tablones que conformarían la solución en el caso de que exista y la cantidad de tablones por los que pasamos desde el último tablón que agregué a la solución (sin contarlo) hasta el tablón de la iteración actual (contándolo), llamaremos a esta última variable *saltados*.
Eventualmente podemos llegar a una iteración donde $saltados = c + 1$ (c es dato y representa la cantidad de tablones como máximo que puedo saltar) y, si no llegamos a esta iteración quiere decir que terminamos de recorrer todos los tablones del puente antes de llegar a pasar por $c + 1$ tablones desde el último sano.
Analicemos el primer caso donde efectivamente llegamos a esa iteración donde vale $saltados = c + 1$: si llegamos a este punto significa que estamos parados en el tablón mas lejano al cual puedo llegar saltando desde el último tablón que agregué a la solución (porque si puedo saltar c tablones seguidos desde donde estoy, entonces caigo en el tablón $c + 1$). Entonces nuevamente pueden pasar dos cosas: 1) $i - ultimosano > c$ (que la cantidad de tablones entre el tablón de la iteración actual i hasta *ultimosano* sea mas grande que c) 2) $i - ultimosano <= c$. Si estamos en 1) voy a tener una cantidad mayor estricta que c de tablones seguidos donde ninguno de ellos es un tablón sano (la última vez que asigné un valor a *ultimosano* fue o bien antes de empezar a iterar, o sea cuando estoy fuera del puente, o en un tablón que dista a más de c tablones del actual), esto quiere decir que están todos rotos y, como no puedo saltar una cantidad mayor que c de tablones seguidos, concluimos que no existe una solución al problema. Ahora bien si estamos en 2), entonces quiere decir que existe un tablón sano (y está almacenado en *ultimosano* entre el último que agregué a la solución (si no agregué ninguno, entonces este tablón representa el punto de partida fuera del puente) y el tablón representado por la iteración actual i y este tablón NO es el último que agregué a la solución (de ser así no estaríamos en este caso). Entonces lo que hacemos es agregar *ultimosano* a la solución y actualizar el valor de *saltados* para que represente la cantidad de tablones que salté desde el último que agregué a la solución (que es el que agregamos recién, *ultimosano*) hasta llegar al tablón de la iteración actual i. Una vez hecho esto, repetimos el proceso descripto con la siguiente iteración.
Dado que los tablones que deben ser solución se agregan a la misma cuando nos encontramos en una iteración i en la cual vale $saltados = c + 1$, podría suceder que terminemos de iterar y no hayamos pasado por esta iteración i. Esto quiere decir que a partir del último tablón que agregué a la solución, hay una cantidad menor estricta que $c + 1$ tablones y por lo tanto, puedo saltarlos todos, alcanzando así el punto de llegada fuera del puente.
Cuando terminamos de iterar todos los tablones del puente, agregamos como el último "tablón" de la solución, un número mayor estricto que n para indicar

que estamos fuera del puente.

A continuación, presentamos el pseudocódigo que hace lo que describimos arriba:

```
procedure ResolverPuente(puente, c)
    ultimosano ← −1
    saltados ← 0
    solucion ← ∅
    n ← cantidadTablones(puente)
    for i ← 1, n do
        saltados + +
        if puente[i] ← 0 then
            ultimosano ← i
        end if
        if saltados == c + 1 then
            if i − ultimosano > c ∨ ultimosano == −1 then
                no hay solución, termino
            end if
            agregar(solución, i)
            saltados = i − ultimosalto
        end if
    end for
    agregar(solución, size(puente) + 1)
end procedure
```