Energy crisis in New Zealand

During the winter of 2011, New Zealand had to cope with a lack of rain. As a consequence, the reservoirs of hydroelectric plants were hardly filled, which led to the country being hit by a severe energy crisis. The Department of Energy had to develop a contingency plan, where the current could be turned off in certain areas of the country in a systematic but fair manner.

In this contingency plan, the country was divided into n areas, where the area around Auckland was assigned the number 1, and the area around Wellington the number 13. First the power would be be turned off in the area with number 1 (this was the fairest starting point) and then successively in each area m positions further. m is a "arbitrary" number (with $m > 0$), which is called the *jump*. The numbering of the areas is applied cyclically by immediately returning to 1 after n. For the sake of fairness, areas where the power was already turned off are excluded when determining the area m positions further. In this way, for example with $n = 17$ areas with jump $m = 5$ the power is turned off in the contingency plan in the following areas:

1, 6, 11, 16, 5, 12, 2, 9, 17, 10, 4, 15, 14, 3, 8, 13, 7

Pay attention to the fact that, for example, after area 16 the power is turned off in area 5. This is the area $m = 5$ positions further, taking into account the fact that the power was already turned off in area 1. To determine the area five positions further, the following areas are thus counted: 17, 2, 3, 4, 5.
Still, for the sake of fairness, now the problem arises that the power in Wellington should be turned off as last. That is, after all the area in which the Department of Energy is located. Therefore, the "arbitrary" jump m for a given number of areas n should be carefully chosen so that region 13 is last selected. A value of m that meets this requirement is called a *valid jump*.

Assignment

1. Write a function contingencyplan which must be passed two arguments: a given number of areas n and a given jump m. The function should return a list as a result that indicates the order of the areas in which the power will successively be turned off. For example, for the values $n = 17$ and $m = 5$ the list [1,6,11,16,5,12,2,9,17,10,4,15,14, 3,8,13,7] should be

returned.
Hint: In this function, use a local list variable to keep track of the n areas where the power has already been turned off (value True), and in what areas it has not (value False).

2. Use the function contingencyplan to write a function jump. To this function, a number of areas n must be passed as argument. This function must return the smallest valid jump m ($1 \leq m \leq n$) for a contingency plan with n regions, which ensures that area 13 is the last area selected. If such an m does not exist, the function must return the value None. For example, for $ n = $ 17, the function must return the value $m = 7$ as a result.

Example

```
>>> contingencyplan(n=17, m=5)
[1, 6, 11, 16, 5, 12, 2, 9, 17, 10, 4, 15, 14, 3, 8, 13, 7]
>>> contingencyplan(n=16, m=11)
[1, 12, 8, 5, 3, 2, 4, 7, 11, 16, 14, 15, 10, 6, 9, 13]
>>> jump(17)
7
>>> print(jump(14))
None
```

Tijdens de winter van 2011 had Nieuw-Zeeland te kampen met een gebrek aan neerslag. Daardoor kwamen de spaarbekkens van waterkrachtcentrales zo laag te staan, dat het land werd getroffen door een ernstige energiecrisis. Het Ministerie van Energie moest dus een noodplan uitwerken, waarbij de stroom in bepaalde gebieden van het land op een systematische maar eerlijke manier kon worden uitgeschakeld.

In dit noodplan werd het land opgedeeld in n gebieden, waarbij het gebied rond Auckland steeds het nummer 1 kreeg toegewezen, en het gebied rond Wellington steeds het nummer 13. Eerst zou de stroom worden uitgeschakeld in het gebied met nummer 1 (niemand stelde in vraag dat dit het eerlijkste startpunt was) en daarna achtereenvolgens in elk gebied m posities verder. Hierbij is m een 'willekeurig' getal (met $m > 0$) dat de *sprong* genoemd wordt. De nummering van de gebieden wordt cyclisch toegepast, door onmiddellijk na n terug 1 te laten volgen. Omwille van de eerlijkheid worden gebieden waar de stroom eerder reeds werd uitgeschakeld niet meer in rekening gebracht bij het bepalen van het gebied m posities verder. Op die manier wordt bijvoorbeeld in het noodplan voor $n=17$ gebieden met sprong $m=5$ achtereenvolgens de stroom uitgeschakeld in de volgende gebieden:

1, 6, 11, 16, 5, 12, 2, 9, 17, 10, 4, 15, 14, 3, 8, 13, 7

Let hierbij op het feit dat bijvoorbeeld na gebied 16 de stroom wordt uitgeschakeld in gebied 5. Dit is het gebied $m=5$ posities verder, rekening houdend met het feit dat de stroom reeds eerder werd uitgeschakeld in gebied 1. Om het gebied vijf posities verder te bepalen, worden dus de volgende gebieden geteld: 17, 2, 3, 4, 5.
Nog omwille van de eerlijkheid stelt zich nu het probleem dat men er in het noodplan ook wil voor zorgen dat de stroom het laatst wordt uitgeschakeld in Wellington. Dat is immers het gebied waarin het Ministerie van Energie gelegen is. Daarom moet voor een gegeven aantal gebieden n de 'willekeurige' sprong m zorgvuldig gekozen worden, zodat gebied 1 3 als laatste geselecteerd wordt. Een waarde m die hieraan voldoet wordt een *geldige sprong* genoemd.

Opgave

1. Schrijf een functie noodplan waaraan twee argumenten moeten doorgegeven worden: een gegeven aantal gebieden n en een gegeven sprong m. De functie moet als resultaat een lijst teruggeven die de volgorde van de gebieden aangeeft waarin achtereenvolgens de stroom zal worden uitgeschakeld. Zo moet de functie bijvoorbeeld voor de waarden $n=17$ en $m=5$ als resultaat de lijst [1,6,11,16,5,12,2,9,17,10,4,15,14,3,8,13,7] teruggeven.
   Hint: Maak in deze functie gebruik van een lokale lijstvariabele om bij te houden in welk van de n gebieden de stroom reeds werd uitgeschakeld (waarde True) en in welke gebieden nog niet (waarde False).

2. Gebruik de functie noodplan om een functie sprong te schrijven. Aan deze functie moet een aantal gebieden n als argument doorgegeven worden. Deze functie moet als resultaat de kleinste geldige sprong m ($1 \leq m \leq n$) teruggeven voor een noodplan met n gebieden, die er dus voor zorgt dat gebied 13 als laatste gebied geselecteerd wordt. Indien een dergelijke m niet bestaat, dan moet de functie als resultaat de waarde None teruggeven. Zo moet de functie voor $n=17$ de waarde $m=7$ als resultaat teruggeven.

Voorbeeld

```
>>> noodplan(n=17, m=5)
[1, 6, 11, 16, 5, 12, 2, 9, 17, 10, 4, 15, 14, 3, 8, 13, 7]
>>> noodplan(n=16, m=11)
[1, 12, 8, 5, 3, 2, 4, 7, 11, 16, 14, 15, 10, 6, 9, 13]
>>> sprong(17)
7
>>> print(sprong(14))
None
```