

## Oppgave 3a: Pseudokode beskriver veien ned

### Naturlig språk:

Her vil det være naturlig å bruke så mye av input-informasjonen som mulig.

1. Vi vet kattens start posisjon
2. Vi vet at inputen tilsvarer et generelt tre og at den foreldrenoden (a) som ikke finnes i noen barnenoder ( $b_n$ ) vil tilsvare roten til treet.

Basert på dette kan man finne stien katten må traversere ved søke etter kattens posisjon i alle barnenoder. Når man finner kattens posisjon i en barnenode kan katten bevege seg til dens foreldrenode. Kattes posisjon blir så oppdatert og søkingen gjentas til katten står i roten.

### Pseudo kode:

**Input:** Flere tekststrenger med heltall, som representerer foreldre- og barnenoder til et generelt tre. Første tekststreng indikerer kattens posisjon i treet. De neste tekststrengene indikerer for hver linje et subtre med tilhørende barnenoder: **a**, **b<sub>0</sub>**, **b<sub>1</sub>** ... **b<sub>n</sub>**. Treet er ferdig beskrevet når input er -1.

**Output:** Et array, path, som indikerer stien katten må traversere for å nå roten til treet

```
proc find_path(input):
  cat_position <- input[0]
  path[0] <- cat_position

  i <- 1
  shared_index <- 0

  while input[i] ≠ -1 do:
    parents[shared_index] <- input[i][0]
    for j <- 0 to |input[i]| - 1 do
      children[shared_index][j] <- input[i][j + 1]
    i <- i + 1
    shared_index <- shared_index + 1

  shared_index <- 0
  k <- 1

  while shared_index < |children|-1 do:
    for i <- 0 to |children[shared_index]| - 1:
      if cat_position = children[shared_index][i]:
        cat_position <- parent[shared_index]
        path[k] <- cat_position
        shared_index <- 0
        k <- k + 1
      else
        shared_index <- shared_index + 1

  return path
```