

INF122 – Obligatorisk oppgave 1, Høst 2021

I denne oppgaven er det ikke lov til å levere felles løsninger. Du kan samarbeide med medstudenter for å komme frem til løsninger, men koden må skrives på egen hånd. Kopiering av kode (fra hverandre eller andre kilder) aksepteres ikke. En **kjørbar** fil med løsning skal leveres elektronisk senest: **ONSDAG, 3. november, kl 23:59**.

Den første innleveringen må bli godkjent for å kunne gå opp til eksamen – det blir ingen mulighet til å forbedre resultatet etter denne fristen.

I denne oppgaven blir du ikke gitt en steg-for-steg oppskrift, det er opp til deg selv hvordan du vil løse oppgaven ut ifra de gitte kriteriene. Kommandoene er gitt i en rekkefølge som kan være naturlig å implementere dem i. Lykke til!

Introduksjon og overblikk

Spillet vi skal lage i denne oppgaven har mange navn, blant annet [The Tower of Hanoi](#). I spillet har vi tre stolper på rekke, og n ringer av forskjellig størrelse som kan plasseres på disse. Spillet starter med at alle ringene er plassert på den første stolpen, og målet med spillet er å få flyttet alle ringene til den siste stolpen. Det er ikke lov til å plassere en større ring oppå en mindre ring.

Generelle kriterier:

- Spillet starter ved at man kjører main-aksjonen i terminalen (dvs. du må ha en aksjon **main::IO**) som kjører spillet når den blir kalt), programmet tar så inn bruker-input og reagerer riktig på de forskjellige kommandoene (se «Kommandoer»-seksjonen).
- Programmet skal ikke krasje, men i stedet skrive ut passende feilmeldinger til brukeren dersom noe «uventet» skjer (f.eks. ugyldig trekk, ukjent kommando, ugyldig format av kjent kommando, etc).
- Brukeren skal hele tiden, etter initialisering av spillet med en gitt størrelse, kunne se den aktuelle plasseringen av ringer på de tre stolpene, og ringene skal være sentrert på stolpen (slik at tårnet får en «juletreform», ref ukesoppgave 8). Stolpene skal hele tiden stå i ro på skjermen og kommandoer skrives alltid samme sted på skjermen.
- Brukeren skal kunne se på skjermen hvor mange trekk de har brukt så langt.
- Spillet skal stoppe når alle ringene er flyttet over til den siste stolpen, da skal det også skrives ut en gratulasjonsmelding som inneholder nivået (antall ringer), og antall trekk spilleren brukte. Brukeren blir så tatt tilbake til startmenyen og kan velge å starte et nytt spill, eller å avslutte programmet.

Kommandoer

Du skal kunne håndtere flere forskjellige kommandoer fra brukeren, her er en oversikt:

b n : starter et nytt spill med n ringer (der den største er n stor, og den minste 1). n må være et positivt heltall, du kan selv bestemme en øvre grense (som ikke kan være mindre enn 10).

Eksempel:

```
Start a new game with: b <nOfRings>, or quit with: q
b 5

      |
      #
     ##
    ###
   ####
  #####
 #####
 #####

      |
      |
      |

Number of moves: 0
█
```

Kommandoen **b n** kan også brukes midt i spillet, for å avslutte det og starte et nytt spill.

q : quit spillet, dvs du skal avslutte programmet. Dette skal kunne gjøres fra alle stadier i spillet.

f t : flytter øverste ring fra stolpe f til stolpe t. Du må sjekke at trekket er gyldig (dvs at både f og t er gyldige stolper (to forskjellige tall 1, 2 eller 3), og at man ikke ender opp med en større ring oppå en mindre ring)

Eksempel gyldig trekk (flytter den øverste ringen fra stolpe 1 til stolpe 3, og øker moves-telleren med 1):

```
      |
      #
     ##
    ###
   ####
  #####
 #####
 #####

      |
      |
      |

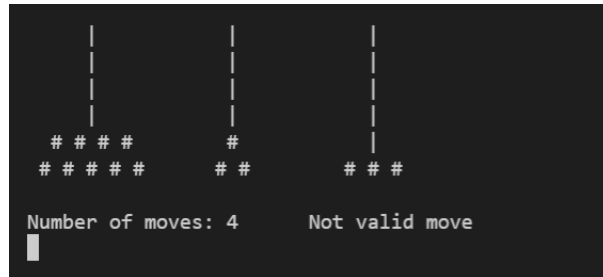
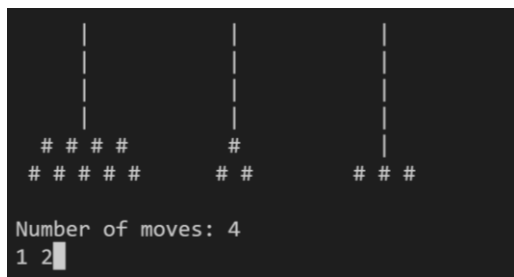
Number of moves: 0
1 3█
```

```
      |
      #
     ##
    ###
   ####
  #####
 #####
 #####

      |
      |
      |
      #

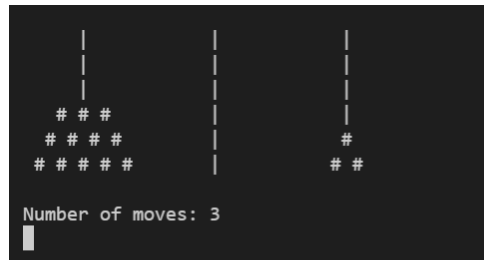
Number of moves: 1
█
```

Eksempel ugyldig trekk (prøver å flytte den øverste ringen fra stolpe 1 (som har størrelse 4) til stolpe 2 (der den øverste ringen har størrelse 1), siden $4 > 1$ så er dette et ugyldig trekk. Brukeren får en passende feilmelding, moves-telleren endres ikke, og spillet fortsetter):

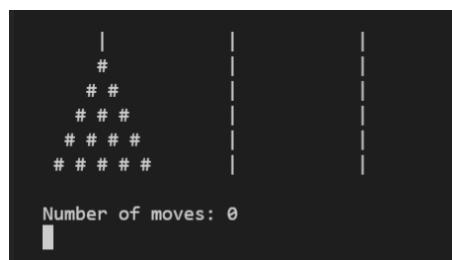
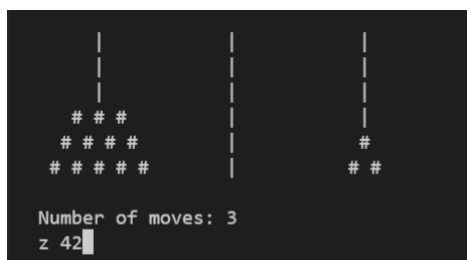


z n : angre n trekk. Hvis n er større enn antall trekk som er gjort så går du bare tilbake til start-brettet. Du skal også trekke det riktige antallet fra moves-telleren.

Eksempel:



Eksempel der n er større enn antall trekk som er gjort:



Følgende kommando kan være litt vanskelig å implementere, men teller kun 10%. Det betyr at du fortsatt kan få en god score uten å gjøre denne oppgaven.

h : gir spilleren hint med det neste «optimale» trekket (som må være basert på en løsning av spillet fra den nåværende tilstanden). Her må du altså programmere en funksjon som løser spillet fra en gitt tilstand, returnerer en liste av trekkene som må gjøres for å vinne, og viser det første trekket fra listen. Følgende pseudokode kan brukes som inspirasjon:

*flytt ring **n** til stolpe **target**:*

*basecase: **n** == 0 -> return*

*rekursjons case 1: **n** er allerede på stolpe **target** -> flytt ring (**n-1**) til stolpe **target***

*rekursjons case 2: **n** er ikke på stolpe **target** ->*

*vi kaller stolpen **n** er på for **source** og den siste stolpen for **helper***

*flytt ring (**n-1**) til stolpe **helper***

*flytt ring **n** fra stolpe **source** til stolpe **target***

*flytt ring (**n-1**) til stolpe **target***