Norges teknisk–naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

TDT4105 IT Grunnkurs
Høst 2014

Øving 10

`Merk:` Denne øvingen går over to uker, og teller som enten én eller to øvinger, avhengig av hvor mye du fullfører.

1 Teori

a) Hva er forskjellen på HTTP og HTTPS?

b) Hva er kjøretiden til denne funksjonen uttrykt ved hjelp av `n`?

```
function vector = foo( v )
  n = length( v );

  for i = 1:n
    for j = i:n
      if v(i) > v(j)
        tmp = v(i);
        v(i) = v(j);
        v(j) = tmp;
      end
    end
  end
end
```

2 Rekursjon

a) Fibonaccitallene er definert som følger.

$$f_n = \begin{cases} f_{n-1} + f_{n-2} & \text{hvis } n > 2 \\ 1 & \text{hvis } n = 2 \\ 1 & \text{hvis } n = 1 \end{cases}$$

For eksempel er $f_3 = f_1 + f_2 = 1 + 1 = 2$. Dermed blir begynnelsen av rekken slik: 1, 1, 2, 3, 5, 8, 13, 21...
Lag den rekursive funksjonen `fibonacci` som tar tallet `n` som parameter og returnerer det `n`-te elementet i fibonacci-følgen.

b) Lag funksjonen `factorial` som tar tallet `n` som parameter og returnerer resultatet av den matematiske operasjonen $n!$.
Funksjonen er definert slik:

$$fac(n) = \begin{cases} 1 & n <= 1 \\ n \cdot fac(n-1) & ellers \end{cases}$$

c) Lag funksjonen `des2bin(descimal)` som tar inn et positivt heltall (eller 0) og returnerer den binære representasjonen av tallet som en tekststreng. Funksjonen skal være rekursiv, og for hvert rekursive kall skal funksjonen finne det binære sifferet lengst til høyre i resultatet. Funksjonen kan implementeres etter følgende rekursjonsskjema:

$$des2bin(n) = \begin{cases} '0' & n == 0 \\ '1' & n == 1 \\ [des2bin(n/2)\ '0'] & n\ partall \\ [des2bin((n-1)/2)\ '1'] & n\ oddetall \end{cases}$$

Test funksjonen slik:

```
des2bin(0) % '0'
des2bin(1) % '1'
des2bin(2) % '10'
des2bin(127) % '1111111'
```

d) `(frivillig)` Lag funksjonen `tower_of_hanoi(n, source, dest, temp)`. Funksjonen skal skrive ut løsningen på *Tower of Hanoi* problemet. Se wikipedia for definisjon av problemet. Eksempelutskrift for tower_of_hanoi(3, 1, 3, 2) er:

```
Flytt fra 1 til 3
Flytt fra 1 til 2
Flytt fra 3 til 2
Flytt fra 1 til 3
Flytt fra 2 til 1
Flytt fra 2 til 3
Flytt fra 1 til 3
```

3 Sudoku

Spillet Sudoku handler om å fylle 9 rader, kolonner og kvadrater med alle tallene fra og med 1 til og med 9. Se figur over spillebrettet under.

8								
9								
1								
2						1	3	5
3						2	4	6
4	5	6	1	2	3	9	8	7
5								
6								
7								

Først når alle 9 kolonner, rader og kvadrater er ferdig utfylt er spillet ferdig. Vi kan bruke tallet 0 for å representere en plass som ikke er fylt med et tall enda. For eksempel mangler fortsatt sifrene 4, 6, 7, 8, 9 (i kolonnene 2-6) i raden under.

2	0	0	0	0	0	1	3	5

a) Din oppgave er å implementere Sudoku i Matlab. Du står fritt til å bygge opp løsningen som du vil, men løsningen skal oppfylle følgende krav:

- Brukeren skal kunne skrive inn et tall i en valgfri celle. Dersom tallet ikke er gyldig, dvs ikke mellom 1 og 9, skal en feilmelding skrives ut.
- Brukeren skal ikke kunne fylle inn et tall som allerede finnes i den samme raden, kolonnen eller kvadratet.
- Brukeren skal kunne slette et tall fra en celle.
- Hver gang brukeren fyller inn eller sletter et tall skal det nye brettet skrives ut på en fin måte. Et eksempel kan være som vist under (tallene over og ved siden av brettet angir her henholdsvis kolonne- og radnummer).

```
    1 2 3   4 5 6   7 8 9
   ----------------------
1 |   5   |   4 3 | 2     |
2 | 2     | 6   9 |     4 |
3 |       | 1 2   |       |
   ----------------------
4 |     8 | 7     | 4     |
5 |     6 |     4 | 1 2 9 |
6 |   1   | 3 6   |   8 5 |
   ----------------------
7 | 7     |   1   |   4   |
8 | 1 4 9 |     5 |   6   |
9 |   3 5 | 4 7   |     2 |
   ----------------------
```

- Brukeren skal kunne laste inn et brett fra en tekstfil.
- Et halvutfylt brett skal kunne lagres til fil, slik at man kan fullføre det senere.
- Spillet skal skrive ut en hyggelig gratulasjonsmelding dersom man har klart brettet.
- Alt skal utføres gjennom et brukervennlig grensesnitt. Det vil si at brukeren ikke skal trenge å kalle på funksjonene selv, men at alt gjøres via `input` eller ved å lese fra/skrive til fil.