KTH, NADA, Linda Kann

2D1320, Tilda, Tentamenslösning 3 juni 2000

1. *Aktieautomat*

i	next[i]
1	0
2	1
3	1
4	0
5	1
6	3
7	1

2. *Orderstack*

1. Poppa alla order i stacken och lägg dom i kön.
2. Ta alla order från kön och pusha på stacken (nu har vi vänt stacken).
3. Poppa en order i taget från stacken, skriv ut den och lägg den i kön - fortsätt tills stacken är tom.
4. Ta alla order från kön och pusha på stacken (nu har vi återställt stacken!).

Rekursiv tanke

1. Om stacken inte är tom: Skriv ut resten av stacken.
2. Skriv ut första ordern.

3. *Valutaspekulation* Breddenförstsökning (eller bästaförstsökning) i ett träd där rotposten innehåller startbelopp och valuta. Söner skapas genom växling (en procedur går igenom alla möjliga växlingar) och läggs dessutom i en kö (eller prioritetskö). Högst uppnådda belopp i varje valuta lagras i en vektor och dumsöner som inte överträffar detta läggs aldrig in i kön.

Man avbryter när en post med startvalutan och ett belopp som överskrider startbeloppet dyker upp. Med hjälp av faderspekarna kan man skriva ut växlingskedjan (en rekursiv utskriftsprocedur). Om det inte finns någon lönsam växlingskedja tar kön så småningom slut.

En trädmodul och en kömodul (eller heapmodul) är bra att ha!

4. *Transaktionssyntax*

```
<konto>       ::= <kontonr> <transaktion>
<transaktion> ::= <datum> <saldo> \n | <datum> <saldo> \n <transaktion>
<kontonr>     ::= <nr><nr><nr><nr><nr><nr>-<nr><nr><nr><nr>
<datum>       ::= <år>-<månad>-<dag>
<år>          ::= <nr><nr><nr><nr>
<månad>       ::= 01|02|...|12
<dag>         ::= 01|02|...31
<nr>          ::= 0|1|2...|9
<saldo>       ::= <nr>,00 | <nr><saldo>
```

5. *Rekursiv ränta*

Rekursiv tanke: Värdet på kapitalet i år är värdet på förra årets kapital multiplicerat med årets ränta.

6. *Anbud på Telia*

Lagra anbuden i en prioritetskö! Trappvillkoret ser till att nya anbud alltid hamnar på rätt ställe. När tiden gått ut plockar man helt enkelt ut ett anbud i taget och säljer motsvarande antal aktier.