

Programmering i C/C++

Räkneövning 1. (5.11.2009)

Deadline för lösningar: **12.11.2009, kl.23.59**

Lösningar till uppgifterna inlämnas elektroniskt på adressen: <https://xprog28.cs.abo.fi/ro.nsf>

OBS!! Denna räkneövning löses i C, dvs INTE i C++!!

En löst uppgift utgörs av en dokumentation i PDF- eller Word-format. Denna dokumentation ska innehålla:

1. En introduktion till problemet och en beskrivning (i ord) hur den är löst
2. Listning av programkod
3. Skärmdump av utskrift från programkörning

Kontakta kursassistenten ifall du kör fast eller undrar över något!

E-mail: tsoinine@abo.fi

Uppgift 1 (2p) Skriv ett program som läser in en text från standard input och skriver ut texten på skärmen.

Uppgift 2 (3p) Skriv ett program som tar som inargument ett filnamn från command prompten.

Använd detta filnamn till att öppna/skapa en fil för utskrift. Läs in från standard input data till strukturen:

```
struct address {
    char forNamn[MAX_NAMN_LANGD];
    char efterNamn[MAX_NAMN_LANGD];
    char gatuAddress[MAX_ADDRESS_LANGD];
    int postNummer;
    char postAnstalt[MAX_POST_LANGD];
}
```

Skriv ut dessa fält till filen. Exempel på filens innehåll efter utskrift:

```
Axel Eklund
Joukahainengatan 3-5 A
20250 Åbo
```

Uppgift 3 (5p) Använd strukturen från uppgift 2 och skapa en länkad lista som innehåller personadresser som element. Listan skall stöda följande funktioner:

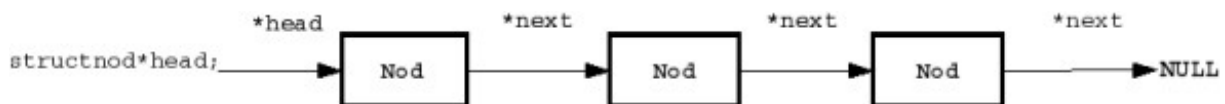
- Initialisera listan
- Dynamiskt sätta in nya element till listan (i slutet)
- Skriv ut listans element
- Töm listan (och frigör allokerat minne!)

Skriv ett program som använder alla dessa funktioner för en lista med minst 3 element.

Använd följande definition för listans element:

```
struct nod {
    struct nod *next;
    struct address data;
}
```

Exempel på en list-struktur med 3 element:



Man måste se till att det alltid finns en nod (den sista noden) vars `*next = NULL`. En länkad lista går egentligen bara att läsa i en riktning. Börja från head-noden och använd `*next` pekaren för att hitta nästa element ända tills man stöter på NULL. Vill man "stega" i listan i båda riktningarna gör man en dubbellänkad lista som utöver `*next` pekaren har även en `*prev` pekare som ger det föregående elementet.

Kompilera programmen så här: **gcc -Wall program.c -o program**

För ytterligare information angående kompileringen, skriv "man gcc" i prompten.