

Programmering i C/C++

Labb 1

- Deadline för lösningar: **Ons. 22.9**
- Lämna in på:
<https://xprog28.cs.abo.fi/ro.nsf>

Uppgift 1

- Initialisera en char räckta som buffer för texten.
 - Ex. `char racka[256];`
- Läs sedan in text från standard input (`stdin`) till räckan.
 - Tips: `fgets()`
- Skriv sedan ut texten på skärmen
 - `printf()`
- Inkludera bibliotek som behövs för dessa funktioner.

Uppgift 2

- Använd `int main(int argv, char *argv[])` för main funktionen.
 - `argv` innehåller antalet argument
 - `argv[1]` pekar på en sträng med första argumentet osv.
- `Fopen()` öppnar en filström för läsning eller skrivning
 - `FILE *fp = NULL; fp = fopen(...);`
 - Kolla gärna att `fp` inte fortfarande är `NULL` efter `fopen()`!
 - `fprintf()` kan användas för att skriva till filströmmar
 - Kom också ihåg att stänga strömmen med `fclose`
- `sscanf()` kan användas på en inläst sträng för att få ut en integer, t.ex. postnumret i adressen.
- Obs! `fgets()` behåller newline tecknet i en inmatad sträng.
 - Kan t.ex. skriva en funktion som ersätter newline med null tecknet, `\0`
 - `strlen` kan användas för att få längden på en sträng (exklusive `\0` tecknet). Dvs. `strang[strlen(strang)-1]` är troligen ett newline, `\n`, tecken efter `fgets`.

Uppgift 3

- Noder kan allokeras med dynamisk minnesallokering
 - t.ex. `malloc(2*sizeof(int))` returnerar en pointer till ett minnesområde som rymmer två heltal (`int`).
 - Allokera en integer array t.ex. så här:

```
int *array;  
array = (int *) malloc(2*sizeof(int));
```
 - `free()` frigör allokerat minne. Detta sköts ju inte automatiskt i C!
- Skriv helst skilda funktioner för listoperationerna (lägga till element osv.)
- Pekare till strukturer (`struct`) fungerar i stort sett likadant som andra pekare.
 - Referera till ett element i en `struct`-pekare med `"->"`
 - t.ex. `struct t_adress *padr;`

```
padr->fornamn //ger fältet fornamn i adresstrukturen som  
//padr pekar på
```

Uppgift 3:

Listfunktionsprototyper

exempel

- //Listfunktioner
- struct t_nod *infoga_adress (struct t_nod *head);
- void skriv_ut_lista (struct t_nod *head);
- struct t_nod *tom_lista (struct t_nod *head);