

## F04 - Repetition av F1-F3

Programmeringsteknik med C och Matlab, 7,5 hp

Niclas Börlin

[niclas.borlin@cs.umu.se](mailto:niclas.borlin@cs.umu.se)

Datavetenskap, Umeå universitet

2023-10-03 Tis

- ▶ Vid beräkning av ett uttrycks värde måste vi hålla reda på en del saker
  - ▶ Operatorernas prioritet
  - ▶ Operandernas datatyp (påverkar operatorerna)
  - ▶ osv.
- ▶ Vad har följande uttryck för värde?
  - ▶  $1+2*2/5.0$
  - ▶  $1.0+2*2/5$
  - ▶  $(1+2)*2/5$

## Strukturerad problemlösning

- ▶ När du löser uppgifter under kursen
  - ▶ Förstå och analysera (input?, output?, samband?)
  - ▶ Designa (lösningssidé, algoritm, testa lösningen)
  - ▶ Implementera och testa
- ▶ Ta först fram en lösning
- ▶ Koda sedan
  - ▶ Träna först på olika konstruktioner fristående
- ▶ Om du kör fast i kodningen
  - ▶ Släpp tangentbordet
  - ▶ Försäkra dig om att din lösning fungerar, eller justera lösningen, träna på olika konstruktioner fristående
  - ▶ Greppa tangentbordet igen

## Större exempeluppgift (OU1, HT09) (1)

1. Skriv ett program som konverterar temperaturangivelser mellan Celsius och Fahrenheit
2. Formeln du använder för att omvandla Celsius till Fahrenheit är  $F = (9.0/5.0) * C + 32.0$
3. Programmet ska börja med att fråga vilken omvandling som ska ske
4. Beroende på användarens svar ska sedan ett gradtal läsas in och omvandlas
5. Svaret skall sedan presenteras för användaren
6. Ungefär så här:
  1. Convert from F to C
  2. Convert from C to FYour choice: 2  
Enter temperature in centigrade: 21.3  
21.30 C is 70.34 F

- ▶ Om användaren anger ett felaktigt val i början så ska en varning skrivas ut:
  1. Convert from F to C
  2. Convert from C to FYour choice: 3  
Incorrect choice
- ▶ Tips: Börja med att skriva in koden för att skriva ut "menyn" och läsa in användarens val till ett heltal
- ▶ Skriv därefter
  - ▶ en funktion för att konvertera från Celsius till Fahrenheit och
  - ▶ en för att konvertera från Fahrenheit till Celsius
- ▶ Skriv ert namn och användarnamn som en kommentar i filen

- ▶ I motsats till de flesta C-funktioner kan `printf` ta ett godtyckligt antal parametrar:
  - ▶ Den första är formatsträngen som innehåller text och platshållare
  - ▶ Sedan följer ett värde per platshållare
- ▶ En biblioteksfunktion — inte en del av språket

```
printf("If %f is larger than %d then %s\n", x, n, s);
```

## printf

- ▶ Formatsträngen till `printf` kan innehålla styrkoder (finns fler än dessa):
  - ▶ `%c` ett tecken
  - ▶ `%d` ett heltal
  - ▶ `%f` ett flyttal (decimaltal)
  - ▶ `%e` ett flyttal i exponentialform
  - ▶ `%g` som `%e` eller `%f` beroende på vilken som blir kortast
  - ▶ `%s` textsträng
  - ▶ `%%` tecknet `%`
- ▶ Man kan även styra hur många positioner en variabel skall få vid utskriften (minst)
  - ▶ `%5d` totalt 5 positioner (högerjusterat)
  - ▶ `%6.2f` totalt 6 positioner med 2 decimaler

## Escape sequences

- ▶ Vi har framförallt sett en så kallad *escape sequence*, nämligen `\n` för *newline*
  - ▶ Det finns många andra
  - ▶ Ni behöver kanske känna till dessa två
    - ▶ `\"` för ett citattecken
    - ▶ `\\` för tecknet `\`
- men troligen inte någon mer, t.ex. inte
- ▶ `\t` för tabulator
  - ▶ `\r` för return (tillbaka till början på raden)
  - ▶ `\b` för backspace
  - ▶ `\a` för alert (ljud- eller ljussignal)

- ▶ Precis som printf tar scanf ett godtyckligt antal parametrar:
  - ▶ Först kommer en formatsträng
  - ▶ Sedan en variabeladress per platshållare
- ▶ `scanf("%lf %d", &x, &n);`
- ▶ Platshållarna är desamma som för printf förutom
  - ▶ `%lf` för **double**

- ▶ Om platshållaren är `%c` matchar scanf även blanka tecken
- ▶ Om användaren skriver in mer data än scanf kan matcha buffras denna data och processas av nästa anrop till scanf
  - ▶ Detta kan orsaka problem

## Kontrollstruktur

- ▶ En kombination av individuella instruktioner som ses som en logisk enhet med en väg in och en väg ut
- ▶ Språket C har tre typer
  - ▶ *Sekvens* – Block, sammansatt sats (inom `{}`)
  - ▶ *Selektion* – Välj en av flera vägar
  - ▶ *Repetition* – Upprepa en sats eller ett block

## Short-circuit- eller "lat" evaluering

- ▶ Kodens evaluering stoppas om ett beslut kan tas
- ▶ Om första delen av ett `||`-uttryck evakueras till *sant* evalueras inte andra delen
- ▶ Om första delen av ett `&&`-uttryck evakueras till *falskt* evalueras inte andra delen

```
int a = 1, b = 2, c = a;
if (a == c || b <= a && 2 != b) {
    printf("Is this printed...?\n");
}
if ((a == c || b <= a) && 2 != b) {
    printf("...or this?\n");
}
if ((a == b) && (a == c)) {
    printf("This?\n");
}
```

Förkortning	Uttydning
<code>n +=3</code>	<code>n = n + 3</code>
<code>n -=3</code>	<code>n = n - 3</code>
<code>n *=3</code>	<code>n = n * 3</code>
<code>n /=3</code>	<code>n = n / 3</code>
<code>n++</code>	<code>n = n + 1</code>
<code>n--</code>	<code>n = n - 1</code>

```
for (int i = 0 ; i < 3 ; i++) {  
    for (int j = 0 ; j < 3 ; j++) {  
        printf("<%d,%d> ", i, j);  
    }  
    printf("\n");  
}
```