

## Tentamen i Programmering i R, 7.5 hp

---

Skrivtid: 8.00-12.00  
Hjälpmedel: Inget tryckt material, dock finns "R reference card v.2" och några andra referenskort tillgängliga elektroniskt.  
Betygsgränser: Tentamen omfattar totalt 20 poäng. 12 poäng ger Godkänt, 16 poäng ger Väl godkänt.  
Tänk på följande:  
Skriv dina lösningar i **fullständig och läsbar kod**.  
Lösningen skrivs i en körbar R-fil med namnet **Main.R**.  
Se filen **DocStudent.pdf** för hur tentan ska lämnas in.  
**Kommentera direkt i Main.R** filen när något behöver förklaras eller diskuteras.  
Eventuella grafer som skapas under tentans gång behöver **INTE** skickas in för rättning, det räcker med att **skicka in den kod som producerar figurerna**.

---

**OBS: Glöm inte att spara din fil ofta! Om R krashar kan kod förloras.**

### 1. Datastrukturer (4p)

- (a) Skapa en `data.frame` med objektnamnet `studenter` med följande fyra variabler: **1.5p**
- `name` (character) med elementen Kalle, Jane, Andrej, Lisa, Kim
  - `analysis` (numeric) med värdena 9, 19, 6, 8, 11
  - `sociology` (numeric) med värdena 5, 13, 22, 27, 13
  - `sex` (factor) med värdena Male, Female, Male, Female, Female
- (b) För att bli godkänd på analystentan och sociologitentan krävs minst 10 poäng. Skriv kod som väljer ut namnen på de som blev godkända på analystentan. **1p**
- (c) Läs in det inbyggda datamaterialet `ChickWeight` i R. Använd sedan funktioner ur `dplyr` för att göra följande: Välj ut de observationer där variabeln `Time` har värdet 21, spara som `time_21`. Räkna ut medelvärden grupperat på variabeln `Diet` för `time_21`. Byt sedan namn på variablerna `time_21` till "W", "T", "C" och "D" (i den givna ordningen). **1.5p**

### 2. Kontrollstrukturer (4p)

- (a) Skapa en `for` - loop som itererar över heltalen -150 till 300 och skriver ut "X can be divided by 127" för de tal som är delbara med 127 och "X can be divided by 61" för de tal som är jämt delbara med 61. **2p**

- (b) Skapa en `while` - loop som genomför följande beräkning till dess att  $x > 1000$ . Starta med  $x_1 = 0, x_2 = 1, x_3 = 2$ . Vad är  $x$  när loopen avbryts? **2p**

$$x_i \leftarrow 1 - x_{i-1} + x_{i-2}^2$$

### 3. Strängar och datum (4p)

- (a) Läs in paketen `lubridate` och `stringr` i R. **0.5p**
- (b) Jeanne d'Arc är Frankrikes nationalhelgon. Hon föddes den 6 januari 1412 i Domrémy i Lorraine som en bondflicka och lärde sig aldrig att läsa eller skriva. Trots detta blev hon en av Frankrikes mest kända härförare i kriget mot England. Den 23 maj 1430 togs hon tillfånga och lämnades till Engelsmännen som dömde henne till döden genom att brännas på bål. Jeanne d'Arc brändes till döds den 30 maj 1431 för kätteri. **1.5p**
- Vilken veckodag togs Jean d'Arc tillfånga?
  - Hur många dagar levde Jeanne d'Arc i fångenskap?
  - Hur många månader levde Jeanne d'Arc?
- (c) Läs in textfilen `brb3kap.txt` i som innehåller tredje kapitlet i brottsbalken. Döp textvektorn till `brb3kap`. Räkna ut hur många gånger ordet `lag` förekommer i texten, oavsett stora eller små bokstäver, dock ej lagman eller andra ord sammansatta med lag. Använd funktioner ur `stringr`-paketet. **2 p**

### 4. Funktioner: Statistik och grafik (4p)

- (a) Vi ska skapa en egen funktion `my_rexp(n, rate)` för att simulera värden från en exponentialfördelad variabel i följande steg. Vi använder det som brukar kallas "inverse transform sampling". **2p**
- Börja med att dra `n` slumpstal från en uniform slumpvariabel  $U(0,1)$
  - Transformera de uniforma slumpdragningarna till en exponentialfördelning på följande sätt

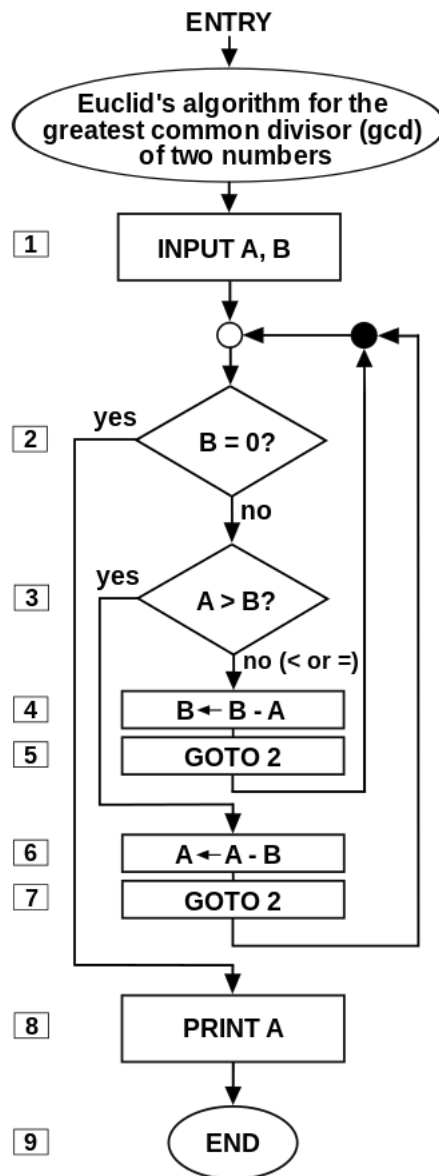
$$T = \frac{-\ln(U)}{\lambda}$$

där  $T$  blir en exponentialfördelad variabel med `rate` =  $\lambda$ .  $\ln()$  är den naturliga logaritmen.

- (b) Visualisera dragningar från `my_rexp(1000, 0.1)` med ett histogram med basgrafiken och ett histogram i `ggplot2`. Histogrammet från basgrafiken ska ha relativ skala på y-axeln. I histogrammet från `ggplot2` ska staplarna vara fyllda med röd färg. Om du inte lyckats implementera `my_rexp()` kan du använda R:s inbyggda `rexp()` som har samma parametrar. **2p**

## 5. Funktioner: Euclides algoritm (4p)

Implementera Euclides funktion för att hitta största gemensamma delare i enlighet med följande flödeschema (taget från Wikipedia). Döp funktionen till `gcd(A, B)`. Från Wikipedia: "Inom matematiken är den största gemensamma delaren (förkortat SGD) av två eller flera heltal vilka alla inte är noll det "största" heltal som delar alla talen. Största gemensamma delaren av heltalen a och b skrivs ofta  $SGD(a, b)$  eller i talteoretisk litteratur endast  $(a, b)$ "  
Ex:  $SGD(48, 180) = 12$



*Exempel:*

```
gcd(152, 72)
```

```
[1] 8
```

```
gcd(37, 13)
```

```
[1] 1
```

Kom ihåg: Lämna in din lösning i en körbar **R-fil** med namnet **Main.R**  
*Lycka till!*