

Numeriska metoder och grundläggande programmering – Laboration 3 – Kryptering

Efter den här laborationen ska du kunna hantera text som data och själv kunna skapa enkla grafiska användargränssnitt (GUI).

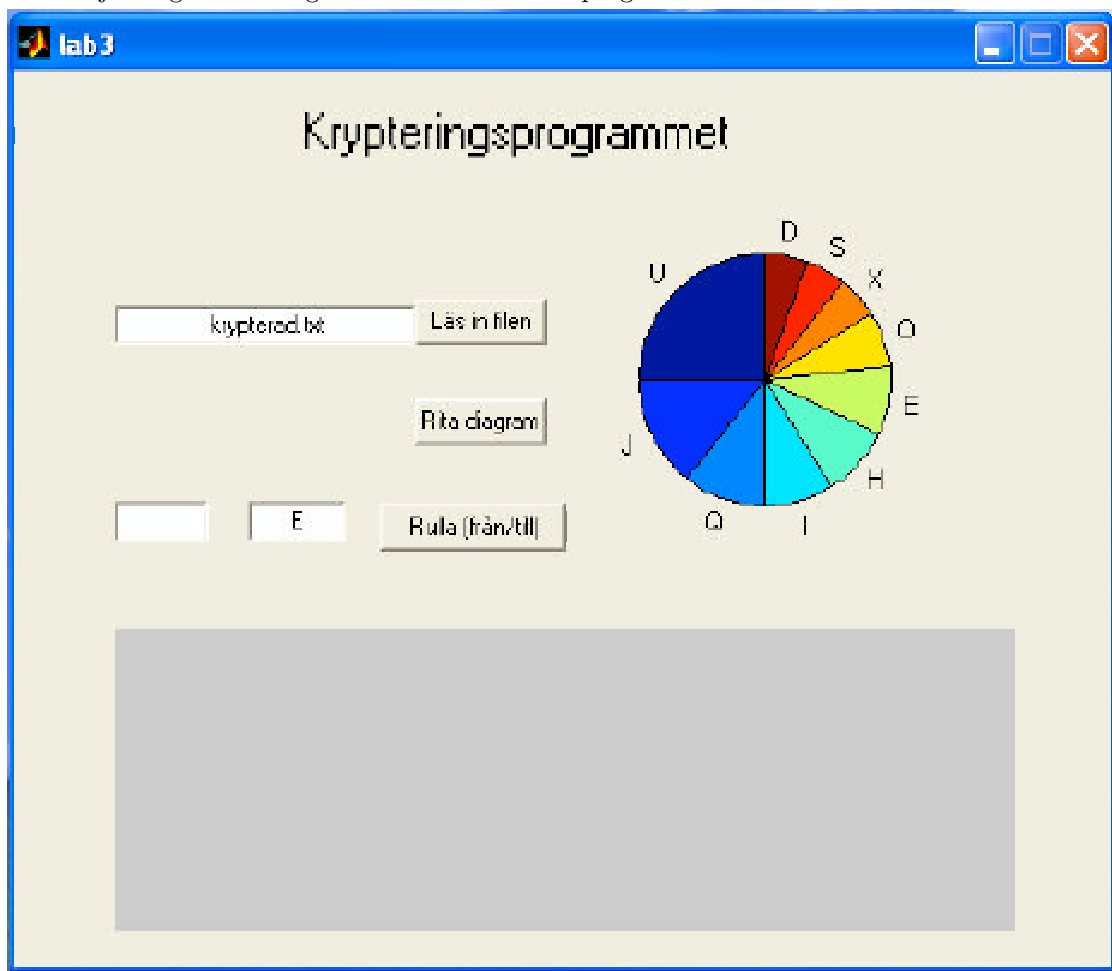
Översikt

Kryptering har blivit en del av vår vardag även om vi inte tänker så mycket på det. När du loggar in på din personliga banksida eller pratar i mobiltelefon sker kommunikationen krypterat.

När man använder *kryptering* konverterar man en *klartext* till en *kryptotext* med hjälp av en krypteringsmetod och vanligen dessutom en *kryptonyckel*. Att konvertera tillbaka kallas att dekryptera. Om man inte känner till krypteringsmetod eller nyckel måste man gissa för att återfinna klartexten. Att göra det systematiskt kallas att forcera kryptotexten.

I denna laboration ska du skriva ett program som forcerar en krypterad text där vi vet krypteringsmetoden men själva måste hitta nyckeln och sedan dekrypterar texten.

Ditt program ska läsa och dekryptera en text på engelska som finns på fil. Det finns många olika algoritmer för kryptering men vi förutsätter att texten är krypterad med »caesarrullning» vilket innebär att varje bokstav har bytts mot en bokstav ett visst antal steg framåt i alfabetet. Med 2 positioners förskjutning blir »The zebra has stripes» krypterat till »Vjg bgdte jcu uvtkrgu». Med vetskap om att »e» är den vanligaste bokstaven i engelsk text kan texten dekrypteras med hjälp av ett frekvensdiagram. Om t.ex. bokstaven »g» är den mest frekventa i kryptotexten kan vi gissa att förskjutningen är 2 steg. Gränssnittet för ditt program bör se ut så här:



Ditt program bör lämna andra tecken än bokstäver oförändrade. Tårtdiagrammet ska innehålla de 10 mest frekventa bokstäverna. Det blir dessutom enklare om du gör om alla bokstäver till versaler. På kursens webbplats finns filer med krypterade texter som du kan använda när du testar.

Tips!

- Tänk igenom programmets struktur innan du skriver kod.
- En oskriven regel är att en funktion gör en sak och gör den bra.
- Funktions- och variabelnamn ska vara beskrivande för att få ett lättläst program.
- Testkör vartefter du blir klar med koden. Det är lättare att hitta fel i lite kod än mycket!
- Läs kryptotexten inklusive radbytestecknet till en vektor. Den kan därefter läggas in i en textruta i gränssnittet (i GUI-versionen).
- Programmera helst beräkningen av bokstavsfrekvenser utan slingor. Matlab anses fungera snabbare så. Även rullningen kan skrivas utan slingor. Försök!
- Skriv gärna programmet utan GUI först, det underlättar vid felsökning. När du gör programmet med GUI kan du kopiera in kod från din första version. Det är inget krav att göra en version utan GUI men däremot med!
- När du börjar med GUI rekommenderar jag varmt Matlabs demo av GUI. Startas t.ex. genom att på kommandoraden skriva:

```
>> demo matlab 'creating graphical user interfaces'
```

- Om du vill kan du bygga ut programmet så att det också kan hantera åäö. Dessa bokstäver har inte teckenkoder intill a-z.