Inlämningsuppgift 3 – Caesar-chiffer

Kurs : DV1550

December 2016

1 Introduktion

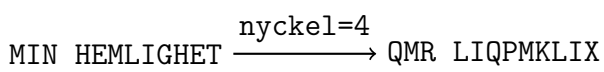
Syftet med den här inlämningsuppgiften är att du ska visa att du kan hantera dynamisk minnesallokering och filer.

2 Uppgift

Du ska skriva ett program som klarar av att kryptera och dekryptera textfiler.

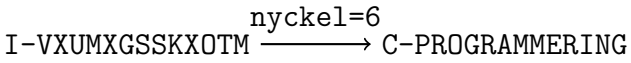
2.1 Caesarchiffer

Caesarchiffer är en krypteringsmetod där klartexten (den läsbara texten) förskjuts n antal steg i alfabetet. Ur Figur 1 ser vi hur krypteringen går till. I exemplet förskjuts alla bokstäver 4 steg i alfabetet, t.ex bokstaven `M` blir `Q` (som ligger 4 steg framåt i alfabetet). Förskjutningen ska ske cirkulärt. Som exempel blir då koden för `Z` med nyckeln 4 istället `D`.



Figur 1: Kryptering klartexten `MIN HEMLIGHET` med nyckeln 4

För dekryptering gäller det omvända (se Figur 2). Texten `C-PROGRAMMERING` har krypterats med nyckeln 6. När den kodade texten dekrypteras med nyckeln 6 blir det som figuren visar.



Figur 2: Dekryptering med nyckeln 6

I den här uppgiften ska stora och små bokstäver behandlas som samma bokstav. Det betyder att resultatet bara behöver innehålla stora boktäver, och att alla bokstäver initialt kan omvandlas till stora vid kryptering/dekryptering. Alfabetet som används är det engelska dvs: `A,B,C, ..., X,Y,Z`. Alla andra tecken (även specialtecken såsom mellanslag, komma, punkt, utropstecken osv.) ska ignoreras (dvs lämnas oförändrade) vid kryptering/dekryptering. Giltiga nycklar är heltal 1 - 25.

2.2 Funktioner att implementera

Två funktioner för kryptering och dekryptering ska implementeras. De ska följa de deklarationer som ges nedan. Båda funktionerna ska returnera strängen "-1" om något gick fel, dvs om krypteringsnyckeln inte ligger i intervallet $[1; 25]$ och ingen kryptering/dekryptering därför skedde. Funktionerna ska ansvara för att allokera plats för den resulterande `char`-arrayen i minnet och returnera en pekare till denna. Funktionerna ska följa nedanstående deklarationer:

```
char* encryptText(char *plainText, int arrLength, int key);

char* decryptText(char *cipherText, int arrLength, int key);
```

Dessutom ska två funktioner för filhanteringen implementeras,

- en funktion som läser in en text från en textfil och placerar den i en dynamiskt allokerad `char`-array, vars adress returneras. Funktionen ska följa prototypen:

  ```
  char* readFromFile(char fileName[]);
  ```

- en funktion som skriver innehållet från arrayen `text`, som skickas in i funktionen, till en textfil med det filnamn som också skickas in som parameter. Funktionen ska följa prototypen:

  ```
  void writeToFile(char fileName[], char text[]);
  ```

3 Frågor

1. Vad är det för skillnad mellan statisk och dynamisk länkning?
2. Nämn någon fördel och någon nackdel med dynamisk länkning jämfört med statisk länkning.

4 Redovisning

Du ska lämna in individuellt. Redovisningen sker genom inlämning via It's Learning för den aktuella labben (Inlämningsuppgift 3). När du är färdig med uppgiften, dubbelkolla så att du har uppfyllt alla funktionalitetskrav och kodlayoutskrav. Du ska lämna in följande:

- Källkoden för programmeringsuppgiften under 2.1 och 2.2 i det här dokumentet. Koden ska vara skriven enligt riktlinjerna i dokumentet "Kodstandard" som finns på kurssidan på It's.
  Döp källkodsfilerna till:
  - `crypto.c.txt` - för filen med alla funktionsdefinitioner
  - `crypto.h.txt` - för filen med alla funktionsdeklarationer
  - `main.c.txt` - för din testfil som visar att dina funktioner fungerar
- Besvara frågorna under avsnitt 3 och lägg svaren i en textfil `svar3.txt`.

Tips! Ett bra sätt att testa är att kryptera en fil och låta en kompis dekryptera den med sitt program och vice versa. Då upptäcker man lättare eventuella tankefel. Det finns en textfil i den här mappen som ni kan testa med.

Det är viktigt att du följer namngivningskonventionen för dina inlämnade filer. **Tänk på att allt du lämnar in ska vara skrivet på engelska.** Tänk också på att inlämnad kod ska vara fri från kompileringsfel och kompileringsvarningar och att ditt program inte får ha några minnesläckor.

VIKTIGT!!
Det kommer *inte* att vara möjligt att lämna in efter deadline. Om du missat den får du ingen ny chans förrän i tentamensperioden, och då får du en annan uppgift.

⚠ Observera att det inte är tillåtet att lämna in kod eller svar som någon annan än du själv har skrivit. Inlämningen kommer att plagiatkontrolleras.