

Programmering 1

Laboration 8: Strängar

Denna laboration är tänkt att både utföras och **redovisas** under de schemalagda timmar som finns anslagna för ändamålet. Ifall du inte har möjlighet att delta i laborationstillfällena, eller inte hinner slutföra laborationen under den schemalagda tiden måste du redovisa dina resultat i en **skriftlig laborationsrapport** som skickas in via Moodle. Ett obligatoriskt delkrav för att kunna bli godkänd i kursen är att **alla laborationer är godkända**.

Obs: Vissa uppgifter (markerade med en asterisk) kommer att gås igenom gemensamt under laborationstillfällena. Dessa uppgifter godkänns alltså automatiskt om du deltar i laborationstillfället.

1. Teckenräkning

Headerfilen **cctype.h** innehåller funktioner som arbetar med enskilda bokstäver. Utnyttja dessa funktioner för att lösa följande uppgift.

Att göra: Skriv en funktion **analyzeString** som räknar antalet versaler ("stora bokstäver", *upper case*) och gemener ("små bokstäver", *lower case*) i en sträng. Tecken som inte är bokstäver får bilda en tredje kategori. Definiera en struktur **stats** som innehåller dessa tre heltal och låt funktionen returnera en sådan struktur till **main**.

Exempel på användning av funktionen:

```
char test[] = "Tjolahopp Tjolahej";
stats result = analyzeString(test);

// Should result in 2 upper, 15 lower and 1 other char
printf("Upper: %d, lower: %d, other: %d\n", result.upper,
                                             result.lower,
                                             result.other);
```

2. Modifikation av strängar 1

Att göra: Skriv en funktion **trimRight** som raderar alla mellanslag från slutet av en sträng.

Exempel på användning av funktionen:

```
char test[] = "Hello World          ";
trimRight(test);
printf("After trim: '%s'\n", test); => 'Hello World'
```

3. (*) *Modifikation av strängar 2*

Att göra: Skriv en funktion **trimLeft** som raderar alla mellanslag från början av en sträng.

Exempel på användning av funktionen:

```
char test[] = "      Hello World  ";
trimLeft(test);
printf("After trim: '%s'\n", test); => 'Hello World  '
```

4. (*) *Strukturer och strängar*

Att göra: Skapa en datatyp (dvs. deklarera en **struct**) för studerande vid Höskolan. Namn, inskrivningsår och antalet avklarade studiepoäng skall ingå.

Skapa en funktion **initStudent** som skapar en strukturvariabel och låter användaren mata in värden till denna. Använd **fgets** för att läsa in strängar.

Skapa därefter en funktion **printStudent** som skriver ut information om en student:

Skriv ett program som använder sig av **initStudent** och **printStudent** för att läsa in information om ett antal studerande och lagra dessa i en räkka.

Obs 1: När du kör detta program kommer du att upptäcka ett problem:

fgets() läser också in och lagrar radbytestecknet **'\n'** i slutet av strängen!

Hur kan detta problem korrigeras?

Obs 2: fgets() läser endast in så många tecken som anges i funktionsanropet. Vad händer med eventuella överlopps tecken? Hur kan vi beakta detta så att programmet fungerar korrekt ifall användaren matar in för mycket data?

5. (*) *Headerfiler*

Att göra: Modifiera uppgift 4 så programkoden delas upp i tre filer:

- En headerfil som innehåller **struct**-deklarationen samt funktionsdeklarationerna för **initStudent()** och **printStudent()**
- En källkodsfil som innehåller funktionsimplementationerna för **initStudent()** och **printStudent()**
- En källkodsfil som innehåller funktionsimplementationen för **main()**