

Inlämningsuppgift 2 – Tic-Tac-Toe

Kurs: DV1550, DV1552

Deadline 8 december 2016

1 Introduktion

I denna inlämningsuppgift ska du skriva egna funktioner i C, och lägga dem som en egen .c-fil med tillhörande deklarationsfil (.h-fil). Den är också en övning i att hantera arrayer.

2 Uppgift

Du ska skriva en version av spelet "Tic-Tac-Toe" ("tre i rad"). Du ska placera dina funktioner i en särskild fil `functions.c` och skriva en tillhörande funktionsdeklarationsfil `functions.h`. Koden för alla funktioner som hör till uppgiften ska ligga i samma fil. Du ska även skicka med en fil med ditt huvudprogram, som implementerar det färdiga spelet med hjälp av funktionerna.

2.1 Spelregler

"Tic-Tac-Toe" spelas på en kvadratisk spelplan med nio rutor. Två spelare spelar mot varandra. Ett drag görs genom att den aktuella spelaren markerar en av de lediga rutorna med sitt märke (kryss respektive ring). Den spelare som lyckas markera tre rutor i rad (d.v.s. på samma rad, i samma kolumn eller i samma diagonal). Om rutorna tar slut innan någon fått tre i rad blir det oavgjort.

2.2 Implementation

Spelplanen ska representeras av en tvådimensionell array av typen `char`. Du ska implementera minst två stycken funktioner som ditt program *ska* anropa. Observera att dessa två funktioner *måste* följa de givna prototyperna. Den ena funktionen `printGame` ska rita ut spelplanen med det aktuella spelläget och följa följande prototyp:

```
void printGame(char board[][3]);
```

(Observera att radlängden i matrisen måste anges för att funktionen ska veta hur många värden som tillhör en rad och när nästa rad tar vid. I minnet lagras ju värdena bara efter varandra i en följd.)

Den andra funktionen `isWinner` ska utifrån gällande spelläge avgöra om den aktive spelaren vunnit efter sitt drag. Funktionen ska returnera sant (1) om spelet är avgjort och falskt (0) annars. Prototypen för funktionen `isWinner` ser ut så här:

```
int isWinner(char board[][3]);
```

Om en spelare matar in en ruta som inte finns eller en som inte är ledig ska spelaren få försöka igen. När spelet är slut ska ett meddelande skrivas ut, som visar vem som vann eller om det blev oavgjort. På nästa sida följer ett exempel på hur en lyckad programkörning kan se ut, kursiv stil betyder att användaren matar in.

(Tips! Om man vill rensa fönstret mellan presentationerna kan funktionsanropet `system("cls");` användas. Kom bara ihåg att inkludera `stdlib.h` i så fall.)

Extrauppgift: Om du vill ha en extra utmaning kan du implementera en "datorspelare" också. Vid spelets uppstart låter du användaren välja om spelet ska köras mot "datorn" eller mot en annan mänsklig spelare. "Dator"-spelaren kan vara mer eller mindre "intelligent".

Körexempel

Welcome to "Tic-Tac-Toe"!

Who is going to play?

Player 1, state your name: *Batman*

Player 2, state your name: *Robin*

```
| 1 | 2 | 3 |
|---|---|---|
| 4 | 5 | 6 |
|---|---|---|
| 7 | 8 | 9 |
```

Batman, in what square do you choose to put your X? *1*

```
| X | 2 | 3 |
|---|---|---|
| 4 | 5 | 6 |
|---|---|---|
| 7 | 8 | 9 |
```

Robin, in what square do you choose to put your O? *7*

```
| X | 2 | 3 |
|---|---|---|
| 4 | 5 | 6 |
|---|---|---|
| 0 | 8 | 9 |
```

Batman, in what square do you choose to put your X? *2*

```
| X | X | 3 |
|---|---|---|
| 4 | 5 | 6 |
|---|---|---|
| 0 | 8 | 9 |
```

Robin, in what square do you choose to put your O? *7*

Robin, in what square do you choose to put your O? *4*

```
| X | X | 3 |
|---|---|---|
| 0 | 5 | 6 |
|---|---|---|
| 0 | 8 | 9 |
```

Batman, in what square do you choose to put your X? *3*

```
| X | X | X |
|---|---|---|
| 0 | 5 | 6 |
|---|---|---|
| 0 | 8 | 9 |
```

Game winner is Batman!

2.3 Frågor

1. Var är skillnaden mellan call-by-reference, call-by-value.
Ge gärna exempel.
2. Vad är en funktionsdeklaration? Ge gärna exempel.
3. Vad är en funktionsdefinition? Ge gärna exempel.
4. Vad är syftet med att använda pre-processordirektivet `#ifndef` i headerfiler? Ge gärna exempel.

3 Redovisning

Du ska lämna in individuellt. Redovisningen sker genom inlämning via It's Learning för den aktuella labben (Inlämningsuppgift 2). 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. Du lämnar in en fil med dina funktionsdeklarationer och en fil med dina funktionsdefinitioner. Du ska dessutom lämna in din `main`-funktion. Överst i varje kodfil ska du skriva en kommentarrad med ditt namn. Döp källkodsfilerna:
 - `functions.h.txt`
 - `functions.c.txt`
 - `main.c.txt`
- Besvara alla frågorna (1-4) under 2.3 i det här dokumentet. Lägg dina svar i en textfil som du döper till `svar2.txt`.

När du är färdig med uppgiften, dubbelkolla så att du har uppfyllt alla funktionalitetskrav och kodlayoutskrav. 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.

VIKTIGT!!

Det kommer *inte* att vara möjligt att lämna in efter deadline. Om du missat den får du ingen ny chans förrän i tentamensperioden vid terminskiftet, 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.