

Labbrapport 3 – Tärningsspelet

Laboranter:

Namn1: Datorid: Malek Abdul Sater

Namn2: Datorid: Abdirahman Omar Ali

Datum då laborationen genomfördes: 19/11 – 19

Genom att skicka in labbrapporten intygar du/ni att följande regler har följts:

1. Laborationsuppgifter skall lösas självständigt av varje laborationsgrupp. Det är tillåtet att diskutera lösningar, men INTE att kopiera lösningar! Det är alltså INTE tillåtet att ge laborationsresultat eller färdiga lösningar till en annan grupp.
2. Bägge gruppmedlemmarna förväntas ta aktiv del i genomförandet av laborationen och skrivandet av rapporten. Detta inkluderar att bygga, programmera, dokumentera, testa och felsöka. Bägge gruppmedlemmarna skall kunna svara på frågor om hur laborationen genomförts och vilka resultat som erhållits.
3. Examination baseras alltid på individuella resultat

4 Moment 2: Implementering av mappningstabell

4.2 Mappningstabell i programminnet

Uppgift 4.2.2

Förklara vad den rad vi lade till i koden i uppgift 4.2.1 innebär och varför innehållet mellan "" är organiserat som det är ovan.

Svar:

Map table innehåller värden där man senare i uppgift 4.2.3 lägger till en rutin som omvandlar map_table med Z pekaren vid inläsning av tangentbord.

Uppgift 4.2.4

Förklara vad kod-raderna i uppgift 4.2.3 gör. INTE vad de olika instruktionerna betyder, utan **varför** de behövs!

Svar:

Inläsningen av tangentbordet har värdet 0-11 som kommer räkna upp i ordning och efter man ska mappa ASCII värden, adderar man RVAL till ZL pekaren . Efter detta kan man addera carry RVAL till ZH och kunna läsa motsvarande tecken i map_table.

5. Moment 3: Användning av strängar i programminnet

Uppgift 5.2.4

Förklara vad de rader vi lade till i koden i uppgift 5.2.1 och 5.2.2 gör. Vad innebär ""+"" efter Z?

Rutinen kommer läsa av hela strängen från programmet där den har begränsat värde på 8 tecken och kommer sedan med hjälp av **PRINTSTRING** macron som sedan skriver ut på displayen. Z+ innebär att efter läsning ökar pointer med 1.

Uppgift 5.2.5

I exemplet använder vi en extra konstant som talar om hur många tecken som ingår i strängen. Det finns ett annat sätt att hantera strängens slut. Googla på tex: "AVR

assembler strings in program memory” och beskriv kortfattat (kod behövs inte) hur man skulle kunna göra istället. (tips: C hanterar sina strängar så...) Även exempel i föreläsningbilder kan vara intressanta.

Svar:

Genom att lägga till en nolla efter strängen. Strängen måste ha ett jämnt antal tecken så måste antingen 1 eller 2 nollor läggas till efter.

8 Moment 6 : Test av komplett system, redovisning och reflektion

Uppgift 8.1.1

Testkör programmet! Om det fungerar som det ska är det dags att undersöka om tärningen ger rättvisa värden:

Svar:

| Antal Kast: | Total kast (0x9E) | Procent: |
|-------------|-------------------|----------|
| '1' | 14 | 12.0% |
| '2' | 20 | 17.2% |
| '3' | 17 | 14.6% |
| '4' | 27 | 23.2% |
| '5' | 12 | 10.3% |
| '6' | 26 | 22.4% |

Uppgift 8.1.3

Redogör för era erfarenheter från denna laboration. Vad har ni lärt er? Gick allting bra eller stötte ni på problem? Om allting gick bra, vad var i så fall anledningen detta? Om ni stötte på problem, hur löste ni i så fall dem?

Abdirahman:

Dagens laboration har hittills varit den mest utmanade samt mest lärofylld av labbar vi har utfört. Nu har vi fått kunskapen till att kunna bygga en Dice-spel dock gick inte det så smidigt som man ville. I uppgift 5.2.1 hade man i uppgift att kunna få ”Welcome” skrivas ut på displayen men det ledde till en hel del felsökningar och tid. Dagar av felsökningar där vi fick kolla igenom kretsen om kablar var felkopplad eller det var någon glapp i kretsen(samt skylla på komponenter för att bli mindre frustrerande) men vi löste det till slut med att använda oss av rätt pekaren och fick det att synas på displayen! Dessutom hade vi problem med register som krockade med varandra från gamla subrutiner och vi fastnade med loopar med tur så löste vi det genom att följa aktivitetsdiagrammet som var angiven i laborations handledningar.

Malek:

Laborationen var relativt svår, då det var mycket nytt och labb handledningen svår att tolka. Bortsett från det var de olika momenten hyfsat straightforward. Vi lärde oss att använda x och z pekare, hantera strängar och använda oss av mapping tables för att bland annat förenkla vår kod. Ett problem vi stötte på till en början var just användningen av z och x pekare och vad de egentligen innebar, och bara efter noggranna google-sökningar nådde vi önskade resultat.

Sedan stannade utvecklingen även upp vid hantering av strängar, då det tog lång tid innan vi insåg att man endast skulle tillsätta en nolla på slutet av en mapping table. Labben var väldigt lärorik.