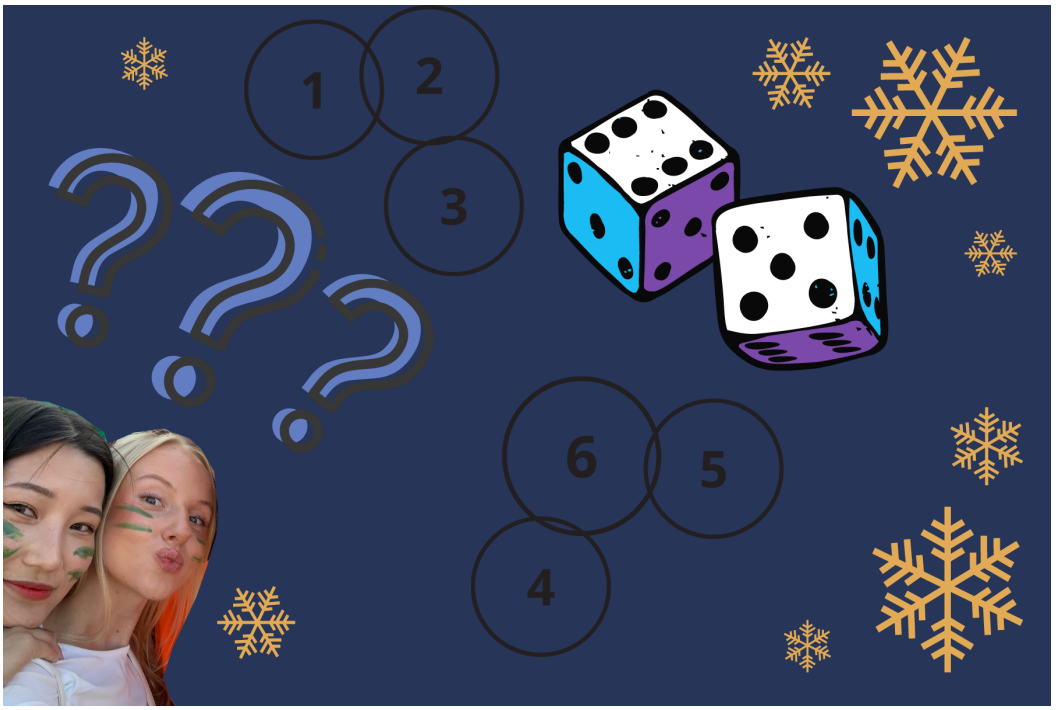
Laboration 4 – Gissa talet

Laboranter:

Namn : Anna Selstam
Datorid : am3963

Namn : Malin Ramkull
Datorid : am5914

Datum då laborationen genomfördes: 2021-12-09



(Anna Selstam 2021)

Genom att skicka in labrapporten 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

Namn: Anna Selstam
Dator-id: am3963

Namn: Malin Ramkull
Dator-id: am5914

Resultat

Uppgift 3.2.4 (redovisas i rapport)
R24 används som ”in-parameter”. Men vilka övriga register kan användas för samma syfte? Varför kan man i detta fall inte använda något annat register istället för R24?
Svar: Funktions-argument är allokerade från vänster-höger. De tilldelas respektive register R25-R8, som används för att skicka parametrar till funktioner. I vårt fall använder våra anropande rutiner R24, därav kan vi inte använda något annat register.

Uppgift 5.2.2 (redovisas i rapport)
Det finns även en annan funktion som påverkar markören på displayen. Med funktionen lcd_set_cursor_pos() anger man var markören ska placeras. Med en kombination av bitvisa operationer skapar man instruktionen som skickas till displayen. Förklara hur de bitvisa operationerna i funktionen lcd_set_cursor_pos() fungerar. Ge gärna exempel på resultatet av bitoperationerna för en valfri rad och kolumn, exempelvis rad 1 (andra raden) och kolumn 2 (tredje kolumnen).
Svar: uint8_t är en osignerad integer typ som är garanterad att vara 8 bitar dvs 1 byte. I vår funktion lcd_set_cursor_pos() tar vi in två uint8_t - en för row och en för col. Vi deklarerar även en lokal variabel cursor_pos som initieras till det binära talet 1000 0000. Vidare i instruktionen skiftar vi in det binära värdet av row 6 steg till vänster, samt det binära talet från col. Vår cursor har nu en position som vi sedan kan använda när vi vill skriva ut något på LCD’n med hjälp av lcd_write.

Uppgift 5.2.4 (redovisas i rapport)
Beskriv hur funktionen lcd_write_str() fungerar, genom att referera till hur man använder pekaren för att stega igenom teckensträngen.
Svar: Funktionen tar emot en sträng och sätter en pekare på denna som sedan skickas vidare till lcd_write. Lcd_write tar in denna som en parameter tillsammans med den andra parametern “DATA” (lcd_register) som instruerar data i form av characters att skrivas på LCDn. I funktionen skrivs en while loop med villkoret att så länge pekaren inte pekar på tecknet ‘\0’, som markerar slutet på en sträng, så skickas tecknet vidare till lcd_write. Sedan ökas pekaren med 1.

Uppgift 7.2.3 (redovisas i rapport)
I början av funktionen deklareras en tecken-array (numbers), som används för att skapa en sträng av siffertecken. Funktionen tillåter endast att maximalt tre siffror kan matas in. Som ni ser så dimensioneras arrayen till att rymma fyra tecken. Vad används den sista positionen till?
Svar: För att kunna utnyttja de inbyggda strängmanipuleringsfunktionerna i C krävs det att den sista positionen reserveras för tecknet ‘\0’ som indikerar slutet på en teckensträng. ‘\0’ betyder, enkelt översatt från Engelska, “noll tecken”, och är alltså vad den sista platsen i arrayen är reserverad för.

Namn: Anna Selstam
Dator-id: am3963

Namn: Malin Ramkull
Dator-id: am5914

Uppgift 8.2.5 (redovisas i rapport)
Redogör för era erfarenheter och kunskaper från denna laboration (minst en halv A4-sida):
• Vad har ni lärt er?
• Om ni får välja en sak, upplevde ni något som var intressant/givande?
• Fanns det något som upplevdes som svårt?
• 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?

Malin: Det var intressant att ta sig an C-kod efter att ha jobbat med assembler-kod. C-kodning påminner mer om den typen av språk man tidigare arbetat med, därmed upplevdes denna laborationen inte lika utmanande som de tidigare. Detta gjorde en mer bekväm och laborationen kändes genomgående avslappnad, vilket var en skön omväxling i förhållande till förra veckans laboration.

Denna laborationens uppbyggnad var annorlunda, då de olika delarna av programmet var färdigkodade för oss och uppgiften var att binda samman de olika delarna, inkluderat ett fåtal justeringar/tillägg i koden. Det var ett nytt och roligt sätt att arbeta på.

I största allmänhet flöt allt på bra. De olika problem vi stötte på lyckades vi till större del felsöka och lösa själva eller efter diskussion med andra labgrupper, vilket kändes givande. Det ska bli kul att fortsätta kommande laborationer i C-kod.

Anna: I denna laboration var koden lättare att förstå då vi, efter kursen Objektorienterad programmering, besitter mer kunskap och förståelse inom denna typ av kodstruktur. Samtidigt var den väldigt utmanande då uppgiften i det stora hela var att komplettera C koden så allt fungerade enligt ändamålet. Detta sätt att jobba på var nytt och ledde till att många frågor uppkom där det krävdes att hela strukturen studerades noga. Efter ett tag och lite guidning kändes det varmare i skorna och sättet vi skulle binda ihop funktionerna med, började lossna. Det mest utmanande men även mest intressanta var att interpretera hur kodningen i C skulle göras, efter tidigare intensiva labbar i Assembly samt Java i vår andra delkurs.

Det var roligt att få testa på “arbetslivet” och hur liknande arbetsuppgifter (i de flesta fall) är presenterade, och att få uppleva och jobba med en annan typ av frustration gentemot uppgiften. Det upplevdes även laborera i ett lugnare tempo denna gång vilket var lite skönt efter 3 extremt krävande labbar.

Namn: Anna Selstam
Dator-id: am3963

Namn: Malin Ramkull
Dator-id: am5914