

## Datorövning 1

- A) Starta Visual Studio 2005. Det gör du genom att först logga in på datorn med din användare, ta fram startmenyn, och sen välja Microsoft Visual Studio 2005. Om du arbetar hemma, måste du (förmodligen) själv installera Visual Studio 2005.

På kursens hemsida, eller direkt på länken <http://basen.oru.se/kurser/pda/2006-2007-p12/vs2005/c/> finns en instruktion om hur man startar Visual Studio, och sen skapar och provkör ett enkelt C-program.

På kursens hemsida, eller direkt på länken <http://basen.oru.se/kurser/pda/2006-2007-p12/vs2005/installera/> finns en instruktion om hur man laddar hem och installerar Visual Studio.

Följ dessa instruktioner, och skapa och provkör det enkla C-programmet.

Några ytterligare tips:

Kompileringen har gått bra om det står 0 errors och 0 warnings i Output-fönstret nedanför editorfönstret. Får du kompileringsfel kan du dubbelklicka på felet och felets position i koden kommer att markeras.

Får du felmeddelanden vid kompileringen eller länkningen måste du rätta felen och bygga projektet på nytt.

- B) Skriv, spara, kompilera, länka och kör nedanstående program. Börja med att välja **File, New** osv.

```
/* bensin.c version 1.1
 * Ditt namn och din användare
 * Beräkning av bensinpris
 * Anpassat för Microsoft Visual C++
 */

#include <stdio.h>
#include <conio.h>

int main()
{
    float literpris, liter, totalpris;

    /* läs in literpris */
    printf("Literpris = ");
    scanf("%f", &literpris);

    /* läs in antal liter */
    printf("Antal liter = ");
    scanf("%f", &liter);

    /* beräkna och skriv ut totalpris */
    totalpris = literpris * liter;
    printf("Totalpris = %.2f kr\n", totalpris);

    /* vänta på tangent */
    getch();

    return 0;
}
```

Efter att du kört programmet och sett att det fungerar, skriver du ut källkoden genom att välja **File, Print** från huvudmenyn. (Det kan hända att du först måste lägga till den aktuella salens skrivare i Windows.)

- C) Skriv ett program i filen `summa.c` som läser in två reella tal, beräknar och skriver ut talens summa, skillnad, produkt och kvot.
- D) Komplettera programmet `summa` så att kvoten bara beräknas om nämnaren (det andra talet) är skilt ifrån 0. Är nämnaren 0 ska istället ett felmeddelande skrivas ut.
- E) Öppna filen `bensin.c`. Komplettera programmet i filen så att det upprepat frågar efter antalet liter och skriver ut totalpriset. Avslutning av programmet ska ske då 0 liter ges.
- \*F) Sätt på motsvarande sätt in en upprepning i programmet i filen `summa.c` så att den upprepas så länge det första talet ej är 0. Matar man in 0 på det första talet ska programmet avslutas direkt utan att det andra talet efterfrågas.
- \*G) Komplettera `bensin`-programmet så att inläsningen av antal liter alltid sker mitt på en tom skärm. Använd funktionerna **clrscr** och **gotoxy** i `MSViscon.h` och `MSViscon.c`, som du kan hämta från kursidans länk Studprog och placera i baskatalogen **M:\PCSA\Program** (eller var det nu är du lägger dina projektkataloger). Filen `MSViscon.h` inkluderar du med **#include "../MSViscon.h"** och med Add Existing Item på Source Files i Solution Explorer adderar du `MSViscon.c` till projektet.

Här är ett exempelprogram som använder MSViscon:

```
#include <stdio.h>
#include <conio.h>
#include "../MSViscon.h"

int main() {
    gotoxy(5, 5);
    printf("Hej!\n");
    gotoxy(10, 10);
    printf("Och tryck en tangent!\n");
    getch();
    return 0;
}
```

## Datorövning 2

- A) Skriv ett program som läser in massa och höjd för en kropp och beräknar och skriver ut dess potentiella energi enligt  $g \cdot \text{massa} \cdot \text{höjd}$ . Definiera konstanten  $g = 9.81$  först med `define` och sedan med `const`.
- B) Skriv ett program som läser in antalet timmar, minuter och sekunder samt beräknar och skriver ut motsvarande tid i sekunder.
- C) Skriv ett program som läser in två sidor och mellanliggande vinkel för en triangel samt beräknar och skriver ut triangelns area enligt  $0.5 \cdot \text{sida1} \cdot \text{sida2} \cdot \sin(v)$ . Vinkeln  $v$  ska läsas in i grader.
- D) Skriv ett program som läser in en tid i hela sekunder och skriver ut den i timmar, minuter och sekunder.
- E) Skriv ett program som läser in ett datum på formen `yymmdd` (6-siffrigt heltal exempelvis `020124` och på skärmen skriver ut :
- ```
År = 2002
Mån = 01
Dag = 24
```
- F) Låt oss kalla ett tresiffrigt positivt heltal för ett spegeltal om entalssiffran är lika med hundratalssiffran. Talet `474` är exempelvis ett spegeltal. Skriv ett program som läser in ett 3-siffrigt tal och kontrollerar om det är ett spegeltal.
- G) Skriv ett program som läser in en stor bokstav mellan A och Z och skriver ut nästa stora bokstav i alfabetet. Bokstaven Z måste du hantera separat och skriva ut texten 'Detta är den sista stora bokstaven!'.
- \*H) Komplettera programmet i D) så att det upprepat frågar efter tid. Programmet ska avslutas då tiden 0 matas in.
- \*I) Skriv ett program som läser in en entals- och en tiotalssiffra som tecken och sedan beräknar motsvarande heltal som multipliceras med 25 och skrivs ut.
- \*\*J) Skriv ett program som inkluderar headerfilerna `limits.h` och `float.h`, som innehåller information om ditt aktuella C-system. Skriv sedan ut största och minsta tal som kan sparas i en variabel av typerna `char`, `int`, `long`, `float`, `double` och `long double`.
- \*\*K) Komplettera uppgift G) ovan så att även Å, Ä och Ö behandlas.

### Datorövning 3

- A) Skriv ett program som frågar efter ett heltal och skriver ut om talet är udda eller jämnt.
- B) Skriv ett program som frågar efter en positiv vinkel i hela grader och skriver ut om vinkeln är spetsig (mindre än  $90^\circ$ ), trubbig (större än  $90^\circ$ ) eller rät.
- C) Skriv ett program som frågar efter ett reellt tal och skriver ut talets kvadratroten med 3 decimaler. Om det inlästa talet är negativt ska ett felmeddelande skrivas ut.

- D) Skriv ett program som frågar efter en siffra mellan 0 och 5 och skriver ut motsvarande morsekod. Använd switch-sats. Morsekoderna för 0 till 5 är:

|   |       |   |        |   |        |
|---|-------|---|--------|---|--------|
| 0 | ----- | 1 | .----- | 2 | ..---- |
| 3 | ...-- | 4 | .....- | 5 | .....  |

- E) Under vissa förutsättningar gäller att bromssträckan för en bil med hastigheten  $v$  i km/h ges av formeln :

$$s = 0.015v^2 \text{ (meter)}$$

Skriv ett program som skriver ut en tabell som visar bromsträckorna för hastigheterna 30, 40, 50, ....., 130 km/h.

- F) Gör ett C-program som räknar ut talet en miljon genom att en miljard gånger addera en tusendel. Vad blir resultatet? Blir det verkligen en miljon? Om inte – varför? Prova med både float och double!
- G) Lägg till en upprepning i uppgift C) så att programmet upprepat frågar efter ett reellt tal och skriver ut kvadratroten. Upprepningen avslutas om talet 0 inmatas.
- H) Skriv ett program som beräknar antalet termer ( $n$ ) som behövs för att summan av den harmoniska serien  $1 + 1/2 + 1/3 + \dots + 1/n$  ska överstiga 10.
- \*I) Skriv ett program som skriver ut delsumman  $1 - 1/2 + 1/3 - 1/4 + \dots + 1/n$  för ett inläst värde på  $n$ . Inläsningen ska upprepas och avslutas då 0 inläses.
- \*J) Skriv ett program som slumpar ett tal mellan 0 och 99 och frågar efter en gissning av talet. Beroende på storleken av det gissade talet skall texten RÄTT, FÖR STORT eller FÖR LITET skrivas ut på skärmen. Programmet ska upprepa så länge man gissar fel och vid avslut ska antalet gissningar skrivas ut. För att slumpa tal ska du använda funktionerna `srand` som startar slumpgeneratorn slumpmässigt och `rand` som returnerar slumptalet. Använd hjälpen med F1 för dessa funktioner.

## Datorövning 4

- A) Skriv ett program som definierar en vektor innehållande 4 heltalselement, läser in värden till alla element och skriver ut elementen i omvänd ordning.
- B) Skriv ett program som läser in 5 reella tal till en vektor och därefter skriver ut vektorns summa, största och minsta element.
- C) Skriv ett program som slumpar 100 tresiffriga heltal till en vektor, skriver ut vektorn, sorterar vektorn samt skriver ut den sorterade vektorn.
- D) Skriv ett program som med funktionen `gets` läser in en sträng innehållande ett antal ord som åtskiljs med blanktecken. Skriv sedan ut strängen tecken för tecken och byt rad efter varje ord alltså vid blanktecken. Strängen kan maximalt innehålla 80 tecken.
- E) Skriv ett program som läser in en sträng med `gets` och skriver ut strängen baklänges. För att bestämma strängens längd kan du använda den färdiga funktionen `strlen` i `string.h`.
- F) Skriv ett program som läser in data till en post som innehåller fälten medlemsnummer, namn och telefonnummer och skriver ut postens fält på skärmen.
- G) Skriv ett program som läser in data till två poster av typen
- ```
struct rtal
{
    int taljare;
    int namnare;
};
```
- som ska avbilda bråktal som exempelvis  $3/5$ . Programmet ska efter inläsning addera ihop bråken till ett summabräk och sedan skriva ut detta summabräk.
- \*H) Skriv ett program som skapar en skruv-vektor innehållande skruvdiametrar med värdena 1.80, 2.15, 2.50, 2.80, 3.15, 3.45, 3.80, 4.15, 4.50, 4.80, 5.45, 6.15, 6.80, 7.50, och 8.15, frågar efter en håldiameter och skriver ut den skruvdiameter som är närmast större.
- \*I) Skriv ett program som läser in flera medlemsposter enligt F) ovan till en vektor av poster innehållande max 10 medlemmar. Inläsningen avslutas då medlemsnummer 0 inläses. Programmet ska avslutningsvis fråga efter ett medlemsnummer och skriva ut namn och telefonnummer för denna medlem.
- \*\*J) Skriv ett program som läser in ett heltal i form av en sträng, omvandlar strängen till ett tal och slutligen skriver ut talet multiplicerat med 2. Exempelvis ska "123" omvandlas till 123 och utskriften ska bli 246. Större än 4-siffriga tal behöver ej hanteras.
- \*\*K) Skriv ett program som slumpar en lottorad bestående av 7 tal mellan 1 och 35. Observera att samma tal ej får återkomma. Lottoraden skrivs slutligen ut sorterad.

## Datorövning 5

- A) Skriv ett program som läser in ett tecken och antalet gånger som tecknet ska skrivas ut samt anropar en funktion för utskriften med huvud enligt :

```
void skriv_tecken(char tecken, int antal)
/* skriver ut tecken antal gånger */
```

Testa att köra programmet i Debuggern. Börja med att sätta en brytpunkt i början av main-funktionen, genom att klicka på den vänstra kanten av kodfönstret. Brytpunkten syns som en stor röd prick. När man sen startar programmet med **Start Degugging** kommer programkörningen att stanna vid varje brytpunkt. Sen kan man stega sig fram rad för rad med F10 eller F11. Kolla variabelvärden genom att ställa dig med muspekaren på variabeln och vänta.

- B) Skriv ett program som läser in ett positivt heltal och kontrollerar om talet är ett primtal dvs. att det ej är jämnt delbart med något annat heltal större än 1 utom sig självt. Primtalstestet ska göras av en funktion som tar talet som parameter och returnerar 1 (sant) om primtal annars 0 (falskt). I funktionen ska du upprepat dividera med först 2 sedan 3 osv. till tal-1 och returnera 0 så fort det går jämnt upp. Går det inte jämnt upp någon gång returneras 1.
- C) Skriv ett program som läser in ett pris i kr och avrundar priset till närmaste 50 öre. Exempelvis ska priset 23.36 avrundas till 23.50 och 23.78 till 24.00. Avrundningen ska ske i en funktion som tar priset som parameter och returnerar det avrundade värdet. Börja med att i funktionen plocka ut hela kr med  $hel = (int)pris$  och sedan decimaldelen med  $dec = pris - hel$ .
- \*D) Skriv ett program som skapar en vektor initierad med 10 tresiffriga vinstnummer i ett lotteri och som sedan läser in ett lottnummer och kontrollerar om lotten har vinst eller är en nitlott. Programmet ska innehålla en funktion som tar vektorn och lottnumret som parametrar och returnerar numret på vinsten (1 till 10) om vinstlott annars 0.
- \*E) Skriv ett program som läser in ett personnummer i form av en sträng med 11 tecken och som kontrollerar och skriver ut om personnumrets alla tecken är ok dvs alla tecken utom det 7:e, som kan vara + eller -, är siffror. Kontrollen ska göras i en funktion som returnerar 1 om allt ok annars 0.
- \*\*F) Komplettera programmet i E) ovan med en funktion som anropas efter teckenkontrollen och som kontrollerar om datum är korrekt angivet med månad mellan 1 och 12 och dag mellan 1 och 31.
- \*\*G) Skriv ett program som i en funktion slumpar värden till en vektor bestående av 10 poster numrerade från 100 till 109 med värden mellan 10.0 och 20.0 av typen:

```
struct matdata
{
    int nr;
    double x;
};
```

Sortera sedan posterna efter värdet på x och skriv ut dessa. Slumpning, sortering och utskrift ska ske i funktioner med lämpliga parametrar.

## Datorövning 6

- A) Skriv ett program som från tangentbordet läser in ett antal reella tal (avslutas med 0.0) och skriver in dessa tal i en textfil rta1.txt, ett tal per rad. Kontrollera filen i en texteditor efter det att du kört programmet.
- B) Skriv ett program som läser filen rta1.txt ovan och beräknar talens medelvärde.
- C) Skriv ett program som räknar antalet semikolon i en textfil vars namn inläses. Testa på något av dina C-program.
- D) Skriv med editorn en textfil innehållande ett antal personers namn, ett per rad. Skriv sedan ett program som läser filen och skriver ut namnen numrerade på skärmen. Numret (börja med 1) ska stå framför namnet på samma rad.
- E) Skriv ett program som slumpar 500 tärningskast och skriver in dessa i en binärfil tarning.dat.
- F) Skriv ett program som läser binärfilen tarning.dat ovan och skriver ut tärningsutfallens medelvärde som ska bli nära 3.5 om slumpningen är tillfredsställande.
- \*G) Skriv ett program som slumpar temperaturer mellan 20 och 30 grader till posterna med nummer 100 till och med 199 av typen :
- ```
struct matpost
{
    int nr;
    double temp;
};
```
- och skriver in posterna i en binärfil temp.dat.
- \*H) Skriv ett program som läser filen temp.dat ovan och skriver ut posterna med lägsta resp högsta temperatur på skärmen.
- \*\*I) Skriv med editorn en textfil bil.txt, som innehåller ett antal bilar med registreringsnummer, ägare och bilmärke, på var sin rad. Skriv sedan ett program som läser värden från bil.txt till poster av lämplig typ och skriver in posterna i en binärfil bil.dat.
- \*\*J) Skriv ett program som läser in ett registreringsnummer, söker efter denna bil i bil.dat ovan och uppdaterar bilens ägare. Den nya ägarens namn läses in.



## Datorövning 7

- \*A) I filen slump.c (finns i din katalog om du kopierat alla filer från kurssidans Studprog) finns funktionen slump som slumpar en vektor med heltal mellan ett minsta och ett största värde och i filen search.c finns funktionen linsearch som söker linjärt efter en nyckel i en vektor och om nyckeln finns returnerar nyckelns platsnummer. Skriv ett program som slumpar en vektor innehållande ett inläst antal(max 100) tresiffriga heltal, skriver ut vektorn och sedan frågar efter ett heltal och söker efter dess plats i vektorn. Kopiera in de funktioner som du behöver från slump.c och search.c till ditt program.
- \*B) I filen slump.c i din katalog finns funktionen slumpsort som slumpar en sorterad vektor med heltal mellan ett minsta och ett största värde och i filen search.c finns funktionen binsearch som söker binärt efter en nyckel i en vektor och om nyckeln finns returnerar nyckelns platsnummer. Skriv ett program som slumpar en sorterad vektor med ett inläst antal (max 100) firsiffriga heltal, skriver ut vektorn och sedan frågar efter ett heltal och söker efter dess plats i vektorn.
- \*C) I filen sort.c i din katalog finns sorteringsrutinen ursort som sorterar vektorer i stigande ordning. Skriv ett program, som använder slump-funktionen i slump.c för att slumpa ett inläst antal (max 100) tre-siffriga heltal till en vektor, skriver ut vektorn på skärmen, sorterar vektorn med ursort samt skriver ut den sorterade vektorn.
- \*\*D) Använd funktionen clock i time.h för att bestämma hur lång tid i sekunder det tar för ursort att sortera en slumpad vektor med 1000, 2000 resp. 3000 slumpade heltal. Hur ändras sorteringstiden med antalet element i vektorn. Använd hjälpen med F1 för att få reda på hur clock-funktionen används.
- \*\*E) Jämför sorteringstiderna för att sortera en slumpad vektor med 2000 element för de tre sorteringsalgoritmerna ursort, bubbsort och insort som alla finns i sort.c.
- \*\*F) Skriv om sök-funktionerna i search.c så att de söker efter en nyckelsträng i en vektor av strängar. Spara funktionerna i stsearch.c och skriv ett huvudprogram där du initierar en vektor med 10 strängar och sedan läser in en sträng som man söker efter i vektorn och om den finns skriver ut platsnummer för.
- \*\*G) Skriv om sorterings-funktionerna i sort.c så att de sorterar strängar. Spara funktionerna i strsort.c och skriv sedan ett program som initierar en vektor av strängar och skriver ut strängarna sorterade på skärmen.
- \*\*H) Skriv en textfil med editorn innehållande ett antal namn, en per rad. Skriv sedan ett program som läser in namnen från filen till en vektor, sorterar vektorn och skriver ut namnen sorterade på samma fil.