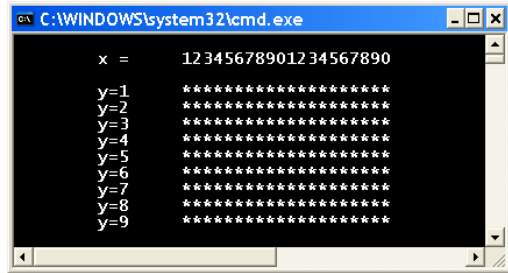


Övningar till kapitel 2

- 2.1 Skriv ett program som med hjälp av en nästlad **for**-sats skriver ut en rektangel fylld med stjärnor (*) till konsolen, bestående av 9 rader och 20 kolumner.

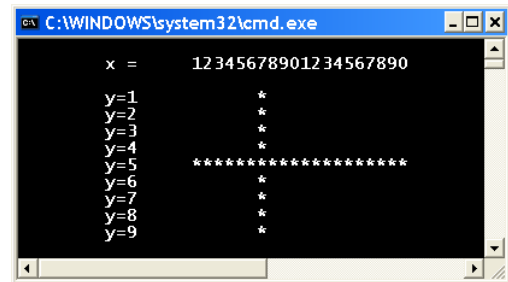
Numrera raderna och kolumnerna utan att förstöra helhetsbilden. Denna uppgift är knappast någon övning i logik, utan snarare i nästlade loopar. Men den förbereder de följande två övningar i sammansatta villkor och logiska operatörer.



```
C:\WINDOWS\system32\cmd.exe

x =      12 3456789012 34567890
y=1      *****
y=2      *****
y=3      *****
y=4      *****
y=5      *****
y=6      *****
y=7      *****
y=8      *****
y=9      *****
```

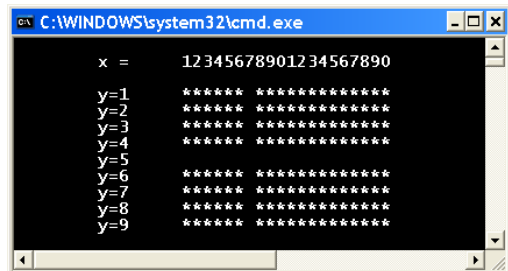
- 2.2 Selekttera (skriv ut) från den stjärnfyllda rektangeln från övn 2.1 endast den 5:e raden och den 7:e kolumnen så att det visas ett kors. Lägg in i den inre **for**-slingan som skriver ut en rad, en **if-else**-sats som i varje varv skriver ut en stjärna om ett sammansatt villkor med ELLER är uppfyllt, annars ett mellanslag. Hur blir det om du byter ut ELLER mot OCH?



```
C:\WINDOWS\system32\cmd.exe

x =      12 3456789012 34567890
y=1      *
y=2      *
y=3      *
y=4      *
y=5      *****
y=6      *
y=7      *
y=8      *
y=9      *
```

- 2.3 Omvandla korset från övn 2.2 till dess *negativ*, dvs skriv ut alla stjärnor från övn 2.1 *utom* den 5:e raden och den 7:e kolumnen. Använd den logiska operatören NEGATION. Negera en gång hela det sammansatta ELLER-villkoret från övn 2.2 och en gång det sammansatta villkorets delvillkor. I båda fall borde du få samma resultat.



```
C:\WINDOWS\system32\cmd.exe

x =      12 3456789012 34567890
y=1      *****
y=2      *****
y=3      *****
y=4      *****
y=5      *****
y=6      *****
y=7      *****
y=8      *****
y=9      *****
```

- 2.4 Skriv ett program som läser in tre tal, hittar och skriver ut det största av dem. Lös problemet genom att använda tre *enkla if*-satser (utan **else**) med sammansatta villkor och den logiska operatoren **&&**. På så sätt kan du i varje *if*-sats jämföra ett tal med de två andra. Varför måste variabeln som lagrar det största talet, initieras vid definitionen?
- 2.5 Skriv ett program som skriver ut sanningsvärdet till det enkla villkoret **a < 10** där **a** är en heltalsvariabel vars värde läses in. Testa ditt program genom att mata in t.ex. 9, 10 resp. 11.
- 2.6 Bestäm sanningsvärden hos de följande logiska uttrycken, först med papper och penna, sedan i ett C#-program:
- a) `(8 < 7) && (true || false)`
 - b) `!(3 < 3.01) || !(0==0) && true`
 - c) `(true || !false) && !(4*5==1) && false`
- 2.7 Följande enkel version av Gissa tal-spelet tillåter endast *en* spelomgång (utan loop). För att koda ett trevägsval nästlar programmet en *if-else*-sats i en annan *if-else*-sats:

```
// GuessIfElse.cs
// Flervägsval med nästlad if-else-sats
using System;

class GuessIfElse
{
    static void Main()
    {
        Console.WriteLine("\n\tGissa ett tal mellan 1 och 20:\t");
        int guessedNo = int.Parse(Console.ReadLine());

        if (guessedNo <= 17)
        {
            if (guessedNo == 17)
                Console.WriteLine("\n\tGrattis, du har " +
                                   "gissat rätt!\n");
            else
                Console.WriteLine("\n\tFör litet!\n");
        }
        else
            Console.WriteLine("\n\tFör stort!\n");
    }
}
```

Modifiera programmet ovan genom att använda logiska operatörer och sammansatta villkor i syftet att förenkla nästlingen. Det nya programmet ska göra samma sak som **GuessIfElse**. Bedöm i slutet själv om det har blivit mer förståelig kod.

- 2.8 Modifiera programmet **PasswdCapslock** (sid 62) genom att lägga in kod som begränsar antalet inloggningsförsök till t.ex. 3. Överskrider man denna

gräns ska programmet avslutas efter att ha skrivit ut ett meddelande av typ *Du har försökt 3 gånger. Nu avslutas programmet!*

Tips: Använd en **if**-sats som avslutar programmet genom att bryta loopen med **break**.