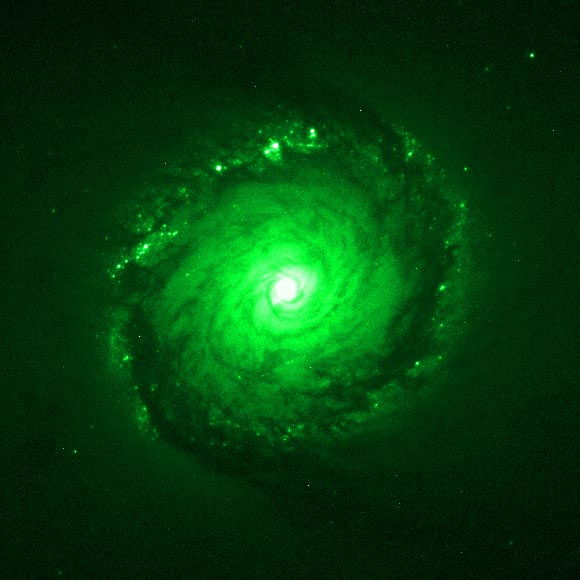
Skapa egna makron i Excel

Grundläggande programmering



© 2007 Lennart Pettersson

Innehåll

[1 Inledning 1](#_Toc155086507)

[1.1 Vad är ett makro? 1](#_Toc155086508)

[1.1.1 Procedurmakron 1](#_Toc155086509)

[1.1.2 Funktionsmakron 2](#_Toc155086510)

[1.2 Vad kan du använda makron till? 2](#_Toc155086511)

[1.3 Spela in makro 3](#_Toc155086512)

[1.4 Att köra en makroprocedur 4](#_Toc155086513)

[2 Visual Basic Editor 5](#_Toc155086514)

[2.1 Miljön i Visual Basic Editor 5](#_Toc155086515)

[2.1.1 Vad är en modul? 5](#_Toc155086516)

[2.1.2 Projektutforskaren 6](#_Toc155086517)

[2.1.3 Egenskapsfönstret 6](#_Toc155086518)

[2.2 Redigera ett inspelat makro 6](#_Toc155086519)

[3 Användarvänlighet 8](#_Toc155086520)

[3.1 Skapa ett verktygsfält med en verktygsknapp 8](#_Toc155086521)

[3.2 Skapa ett eget menyval 10](#_Toc155086522)

[3.3 Skapa en formulärknapp 11](#_Toc155086523)

[4 Att skriva egen kod 13](#_Toc155086524)

[4.1 I huvudet på en programmerare? 13](#_Toc155086525)

[4.2 Beståndsdelar i programmeringsspråket Visal Basic 14](#_Toc155086526)

[4.2.1 Procedurer och funktioner 14](#_Toc155086527)

[4.2.2 Variabler och konstanter 14](#_Toc155086528)

[4.2.3 Inbyggda funktioner i språket 16](#_Toc155086529)

[4.2.4 Villkorssatser 18](#_Toc155086530)

[4.2.5 Vårt exempel med skatteberäkning skrivet i kod 18](#_Toc155086531)

[4.2.6 Repeterande satser (Loopar) 19](#_Toc155086532)

[4.2.7 Funtionsanrop från procedurer 19](#_Toc155086533)

[4.2.8 Strukturerad programmering 19](#_Toc155086534)

[4.3 Skapa makron i Excel i praktiken 20](#_Toc155086535)

[5 Övningsuppgifter 21](#_Toc155086536)

[Uppgift 1: Spela in Makro och koppla till egen knapp 21](#_Toc155086537)

[Uppgift 2: Funktion som beräknar skatten 21](#_Toc155086538)

[Uppgift 3: Funktion som beräknar restid till stjärnor 22](#_Toc155086539)

[Uppgift 4: Funktion som bestämmer kön med hjälp av personnumret 22](#_Toc155086540)

[Uppgift 5: Procedur som beräknar bolån 22](#_Toc155086541)

[Uppgift 6: Procedur som skapar multiplikationstabell 23](#_Toc155086542)

[Uppgift 7: Procedur som räknar antal kvinnor och män i en lista 23](#_Toc155086543)

[Uppgift 8: Procedur som beräknar ålder på givna personnummer 23](#_Toc155086544)

[Uppgift 9: Räkna antal värden i en lista 24](#_Toc155086545)

[Uppgift 10: Funktionsanrop från en Procedur 24](#_Toc155086546)

[Uppgift 11: Procedur som beräknar ordervärdet för angivet land i en lista 24](#_Toc155086547)

[6 Lösningsförslag övningsuppgifter 25](#_Toc155086548)

[Uppgift 1: Spela in Makro och koppla till egen knapp 25](#_Toc155086549)

[Uppgift 2: Funktion som beräknar skatten 25](#_Toc155086550)

[Uppgift 3: Funktion som beräknar restid till stjärnor 25](#_Toc155086551)

[Uppgift 4: Funktion som bestämmer kön med hjälp av personnumret 26](#_Toc155086552)

[Uppgift 5: Procedur som beräknar bolån 26](#_Toc155086553)

[Uppgift 6: Procedur som skapar multiplikationstabell 27](#_Toc155086554)

[Uppgift 7: Procedur som räknar antal kvinnor och män i en lista 27](#_Toc155086555)

[Uppgift 8: Procedur som beräknar ålder på givna personnummer 27](#_Toc155086556)

[Uppgift 9: Räkna antal värden i en lista 28](#_Toc155086557)

[Uppgift 10: Funktionsanrop från en Procedur 28](#_Toc155086558)

[Uppgift 11: Procedur som beräknar ordervärdet för angivet land i en lista 29](#_Toc155086559)

# Inledning

Microsoft har ett stort antal inbyggda verktyg för att underlätta informationshantering. Det mest flexibla och kraftfulla verktyget är möjligheten att spela in eller skriva egna makron. Möjligheterna i detta verktyg är i stort sätt obegränsade eftersom det rör sig om ett fullfjädrat programmeringsspråk.

## Vad är ett makro?

Ett makro är ett inspelat eller skrivet kodavsnitt med en början och ett slut skapat i programmeringsspråket Visual Basic. Detta programmeringsspråk finns som standard i alla ingående program i Office­paketet. Grunderna i språket Visual Basic är gemensam men de olika Office­programmen använder olika standardbibliotek som sin standard för programmeringskoden.

Det finns alltså lokala skillnader i koden vad gäller MS Word, MS Excel, MS Access osv. Detta beror helt enkelt på att du gör olika saker i de olika programmen.

Vad gäller Excel så utgör rutnätet i de olika kalkylbladen en utgångspunkt för data du vill manipulera eller formatera. Visual Basic i MS Excel är alltså anpassat till miljön i MS Excel.

Det finns två typer av makron i MS Excel, Procedurmakron och funktionsmakron. Vad är då skillnaden?

### Procedurmakron

Ett procedurmakro är ett makro som utför en serie åtgärder från början till slut. När du spelar in ett makro skapas alltid ett procedurmakro. Du kan även skriva procedurmakron som hämtar värden från en lista eller enstaka celler, utför åtgärder och/eller beräkningar och presenterar resultatet genom att lägga ut värdena i celler.

|  |  |
| --- | --- |
| Exempel på inspelat makro som ändrar teckensnitt och teckenstorlek på markerad cell.  Blå text är reserverade ord i Visual Basic. Grön text är kommentarer. | |
| Sub Eget\_Format()  ' Makro inspelat av Lennart Pettersson  With Selection.Font  .Name = "Verdana"  .Size = 12  End With  End Sub | Början på makrot (Sub)  Kommentarer i grönt  Ändring av teckensnittet och teckenstorleken  Slut på makrot (End sub) |

|  |  |
| --- | --- |
| Exempel: Ett skrivet procedurmakro som ber användaren ange bruttolön och sedan beräknar bruttolön minus kommunalskatt och därefter visar nettolönen. | |
| Sub Netto()  ' Makro skrivet av Lennart Pettersson  Dim Bruttolön, Skatteprocent, Nettolön    ' Indata  Bruttolön = InputBox("Ange bruttolön")  Skatteprocent = InputBox("Ange skatteprocent")    ' Åtgärder och beräkningar  Nettolön = Bruttolön - (Bruttolön \* Skatteprocent)    ' Utdata  MsgBox "Detta är din nettolön: " & Nettolön  End Sub | Början på makro Variabeldeklaration  Indata  Åtgärder och beräkningar  Utdata  Slut på makrot |

### Funktionsmakron

Ett funktionsmakro är ett makro som tar ett eller flera invärden och returnerar ett värde till den cell där funktionen används. Funktionsmakron måste skrivas för hand. De utgör en möjlighet att skapa sina egna skräddarsydda funktioner. Detta är mycket användbart om du ofta utför komplexa beräkningar och saknar inbyggda funktioner i MS Excel för att utföra dessa.

|  |  |
| --- | --- |
| Exempel: Ett funktionsmakro som beräknar restiden från en given ort till en annan. Funktionen returnerar restid i timmar som ett decimaltal. | |
| Function Restid(Avstånd, Hastighet)  ' Funktion skriven av Lennart Pettersson  Restid = Avstånd / Hastighet  End Function | Början på funktion samt Invärden (parametrar)  Utdata (funktionen returnerar ett värde)  Slut på funktion |

En utförligare redogörelse av beståndsdelarna i Visual Basic samt hur du tänker när du vill programmera följer i *kapitel 4 Att skriva egen kod*.

## Vad kan du använda makron till?

Det finns sannolikt lika många användningsområden för makron som det finns användare. Generellt sätt använder du inspelade makron till att utföra återkommande arbetsmoment som består av flera val.

Det kan till exempel vara att du vill ställa in marginalerna på ett visst sätt, ändra teckenstorlek, teckensnitt och format i cellerna för markerat område. För att utföra detta måste du göra flera val. Här kan du spela in dessa val i en sekvens (ett makro) och koppla makrot till en knapp eller egen meny. Nästa gång du vill utföra ändringen klickar du helt enkelt på knappen eller väljer ditt egna menyalternativ.

Egenhändigt skrivna makron ökar möjligheterna att utföra skräddarsydda åtgärden för en eller flera arbetsböcker. Du kan även använda makron till att kommunicera mellan olika program i MS Office.

## Spela in makro

Du spelar in ett makro på följande sätt:

|  |  |
| --- | --- |
| 1. Välj **Makro** (Macro) och **Spela in nytt makro** (Record new macro) i menyn **Verktyg** (Tools).  2. Skriv önskat namn på makrot i fältet **Makronamn** (Macro name). OBS du kan inte använda mellanslag i makronamnet. |  |
| 3. Välj var du vill lagra makrot genom att klicka på listpilen till höger om fältet **Lagra** **makro i** (Store macro in). Här väljer du om makrot skall sparas lokalt i den arbetsbok du befinner dig, i en ny arbetsbok eller i arbetsboken Egna Makron.  Om du väljer **Denna arbetsbok** lagras makrot lokalt i den arbetsbok du befinner dig.  Om du väljer **Ny arbetsbok** lagras makrot lokalt i en ny arbetsbok.  Om du väljer arbetsboken **Egna Makron** skapas en ny arbetsbok med namnet *Egna.xls* i mappen *xlStart*. Makron lagrade i denna arbetsbok är automatiskt tillgängliga från alla Exceldokument på din dator.  4. Du kan även välja om du vill koppla något snabbkommando till makrot. Tänk i så fall på att välja ett snabbkommando som inte redan används. Detta är valfritt.  5. Om du vill kan du även skriva en beskrivning av vad makrot gör i rutan beskrivning vilket också är valfritt.  6. När du angivit Makronamn och valt var makrot skall lagras, och eventuellt skrivit ett snabbkommando och en beskrivning klickar du på **OK**-knappen. Nu är inspelningen igång och verktygsfältet **Stoppa inspelning** visas. |  |

7. Välj i menyer och med hjälp av knappar vad makrot skall utföra. Om det är väldigt många moment kan det vara bra att skriva en körordning eller ett manus med hjälp av papper och penna så att du väljer de olika stegen i makrot i rätt ordning.

8. När du valt färdigt klickar du på knappen **Stoppa inspelning** (Stop recording) på verktygsfältet med samma namn. Om du av misstag har stängt detta verktygsfält kan du i stället välja **Stoppa inspelning** (Stop recording) under **Makro** (Macro) i **Verktygs**-menyn (Tools).

## Att köra en makroprocedur

För att köra ett inspelat makro gör du så här:

|  |  |
| --- | --- |
| 1. Välj **Makro** (Macro) och **Makron** (Macros) i **Verktygs**-menyn (Tools).  2. Markera önskat makro och klicka på knappen **Kör** (Run). |  |

# 

# Visual Basic Editor

När du har spelat in ett makro eller skrivit egna procedurmakron och/eller funktionsmakron kan det vara bra att redigera sina makron, speciellt när du spelar in makron eftersom du då får en massa kod "på köpet" som du kanske inte vill behålla.

Vi skall därför titta närmare på de grundläggande delarna i miljön. Visual Basic Editor är en gemensam kompilator (kodtolkare) för hela Officepaketet.

För att öppna Visual Basic Editor gör du så här:

1. Välj **Visual Basic Editor** under **Makro** (Macro) i **Verktygs**-menyn (Tools)

Du kan även använda snabbkommandot **Alt** + **F11**

## Miljön i Visual Basic Editor

|  |  |
| --- | --- |
| Miljön i Visual Basic Editor består av olika delar med olika syften.  De delar vi kommer att titta lite närmare på i denna bok är **Moduler**, **Projekthanteraren** och **Egenskapsfönstret**. |  |

### Vad är en modul?

|  |  |
| --- | --- |
| En modul är i grund och botten ett textredigeringsfönster där du redigerar inspelad kod eller skriver egen kod.  Du kan ha flera moduler i en arbetsbok och varje modul kan innehålla flera procedurmakron och/eller funktionsmakron.  Om du har många procedurmakron och/eller funktioner i en modul kan du välja att se endast den procedur eller funktion du vill jobba med genom att:  1. Klicka i önskad procedur eller funktion  2. Klicka på knappen i nedre vänstra hörnet av modulen |  |

Om du vill skapa en ny modul väljer du **Modul** (Module) i **Infoga**-menyn (Insert).

### Projektutforskaren

|  |  |
| --- | --- |
| Uppe i vänstra hörnet finns fönstret **Projektutforskaren** som visar alla olika objekt som finns i den aktuella arbetsboken. Detta fönster visar alla kalkylblad som arbetsboken innehåller samt ett objekt för hela arbetsboken. Om det finns makron i arbetsboken ligger dessa vanligtvis i en eller flera Moduler.  Det kan dock finnas kod i varje kalkylblad eller i något som kallas UserForm. Detta behandlas inte i denna bok utan tas upp på kursen Excel VBA som är en mer omfattande kurs i ämnet makroprogrammering i Excel.  Om Projekthanteraren inte syns väljer du **Projektutforskaren** (Project explorer) i **Visa**-menyn (View). |  |

### Egenskapsfönstret

|  |  |
| --- | --- |
| Nere i vänstra hörnet finns ett **Egenskapsfönster**. Detta kan användas för att ändra egenskaper för olika objekt.  Om du till exempel vill byta namn på en modul markerar du modulen och ändrar namnet under egenskapen *Name* i egenskapsfönstret.  Olika objekt (delar) har olika specifika egenskaper.  Om Egenskapsfönstret inte syns väljer du **Egenskapsfönster** (Properties window) i **Visa**-menyn (View). |  |

## Redigera ett inspelat makro

När du spelar in ett makro händer det allt som oftast att du får kod som gäller olika egenskaper som du inte aktivt har valt.

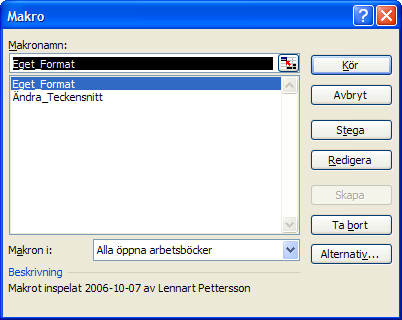
Om du tar bort denna kod kommer makrot att exekveras (köras) snabbare. Detta gäller speciellt när du spelar in långa och omfattande makron.

Du kan komma åt att redigera ett makro på två sätt.

1. Välj **Makro** och **Makron** i **Verktygs**-menyn (Macro-Macros-Tools).

2 Markera det makro du vill ändra i listan.

3. Klicka på knappen **Redigera** (Edit)



eller

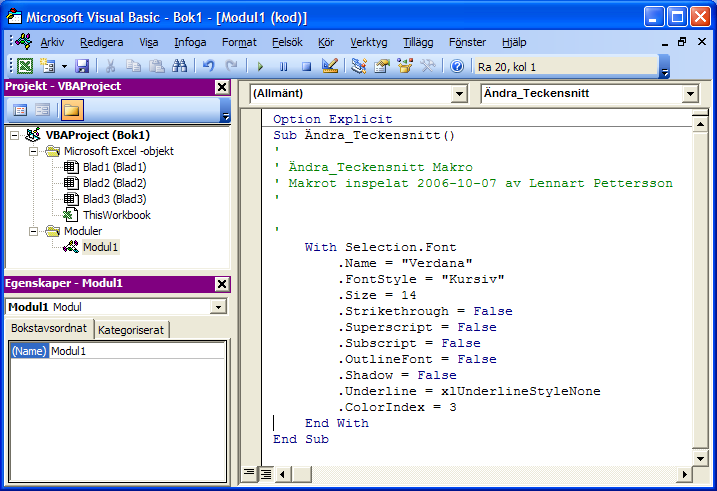
1. Välj **Makro** och **Visual Basic Editor** i **Verktygs**-menyn

(Macro-Macros-Tools).

2. Om du inte ser något makro öppnar du mappen Moduler i Projektutforskaren och dubbelklickar på Modul1 (eller den modul som innehåller makrot).

Nu kan du radera de kodrader du vill ta bort genom att markera raderna och trycka Delete. Hur vet du då vilka radera som du kan radera?

Behåll de rader som innehåller dina val och radera övriga rader. I exemplet nedan har jag spelat in ett makro som ändrar teckensnitt, teckenstorlek, stil och teckenfärg för markerad cell. I exemplet kan du ta bort raderna innanför rutan eftersom de inte är aktivt valda.



# 

# Användarvänlighet

När du spelat in önskat makro eller själv skrivit en egen makroprocedur kan det vara lämpligt att göra makrot tillgängligt genom att koppla det till en verktygsknapp, ett eget menyval eller en formulärknapp. Detta är speciellt viktigt om andra än du själv skall kunna köra makrot.

Om du vill använda verktygsknappar kan det vara lämpligt att skapa ett eget verktygsfält för att kunna placera dina knappar på ett enhetligt ställe.

## Skapa ett verktygsfält med en verktygsknapp

Det kan vara lämpligt att skapa ett eget verktygsfält där du placerar dina verktygsknappar. För att göra det gör du så här:

|  |  |  |
| --- | --- | --- |
| 1. Välj **Anpassa** i **Verktygs**-menyn (Customize – Tools).  2. Markera bladfliken **Verktygsfält** (Toolbars)  3. Klicka på knappen **Nytt** (New)  4. Skriv ett namn för det nya verktygsfältet och klicka **OK**. Ditt nya verktygsfält är skapat. | |  |
|  |  |

Det kan vara lämpligt att dra verktygsfältet uppåt för att placera det i överkant av fönstret.

Nu är det dags att skapa en verktygsknapp som kör ditt nya makro. För att utföra detta gör du så här:

|  |  |
| --- | --- |
| 1. Välj **Anpassa** i **Verktygs**-menyn (Customize – Tools).  2. Markera bladfliken **Kommandon** (Commands).  3. Skrolla ner i rullningslisten vid Kategorier (Categories) tills du kommer till kategorin **Makron** (Macros) och markera denna kategori.  4. Håll ner vänster musknapp på den "gula gubben" under **Kommandon** (Commands) och dra med musknappen nedtryckt till ditt tidigare skapade verktygsfält. När du får ett svart streck på verktygsfältet släpper du musknappen (Se bild till höger). |  |

Nu är det dags att koppla din knapp till önskat makro samt att ändra ikonen antingen genom att välja en befintlig eller genom att skapa en egen.

För att koppla din knapp till ett makro gör du så här:

|  |  |
| --- | --- |
| 1. Se till att din nya knapp är markerad (har en svart ram)  2. Klicka på knappen **Ändra markering** (Modify selection) och välj **Koppla makro till knapp** (Assign macro)  3. Markera önskat makro i dialogrutan och klicka **OK**. |  |

För att välja en befintlig ikon till din knapp gör du så här:

|  |  |
| --- | --- |
| 1. Se till att din nya knapp är markerad (har en svart ram)  2. Klicka på knappen **Ändra markering** (Modify selection) och välj **Ändra knappbild** (Change button image)  3. Markera önskad ikon |  |

För att skapa en egen ikon till din knapp gör du så här:

|  |  |
| --- | --- |
| 1. Se till att din nya knapp är markerad (har en svart ram)  2. Klicka på knappen **Ändra markering** (Modify selection) och välj **Redigera knappbild** (Edit button image)  3. Klicka på knappen **Radera** (Clear) för att radera den befintliga ikonen.  4. Klicka sedan på önskad färg och rita din ikon i rutnätet. Du kan se hur ikonen kommer att se ut under **Förhandsgranska** (Preview)  5. Klicka på **OK**-knappen när du är färdig |  |

För att välja om du vill ha både text och/eller en egen ikon på din knapp gör du så här:

|  |  |
| --- | --- |
| 1. Se till att din nya knapp är markerad (har en svart ram)  2. Klicka på knappen **Ändra markering** (Modify selection) och välj **Standard** (Default style) om du bara vill ha ikon.  Välj **Endast text** (Text only) om du bara vill ha text.  Välj **Bild och text** (Image and Text) om du vill ha både ikon och text. |  |

## Skapa ett eget menyval

Om du föredrar menyval framför verktygsfält kan du skapa egna menyer med egna menyval. För att skapa en egen meny gör du så här:

|  |  |
| --- | --- |
| 1. Välj **Anpassa** i **Verktygs**-menyn (Customize – Tools) eller högerklicka var som helst i verktygsfälten eller i menyn och välj **Anpassa** (Customize) i snabbmenyn.  2. Klicka på fliken **Kommandon** (Commands).  3. Markera **Ny meny** (New menu) längst ner under Kategorier (Categories).  4. Ta tag i **Ny meny** (New menu) under **Kommandon** (Commands) och dra upp till den befintliga menyn och placera på önskat ställe. |  |
| 5. Se till att din nya meny är markerad (svart ram runt den) och klicka på knappen **Ändra markering** (Modify selection) på dialogrutan **Anpassa**. Skriv in önskat namn på din nya meny i rutan till höger om **Namn** (Name).  6. Skrolla i rullningslisten vid Kategorier (Categories) tills du kommer till kategorin **Makron** (Macros) och markera denna kategori.  7. Håll ner vänster musknapp på den "gula gubben" under **Kommandon** (Commands) och dra med musknappen nedtryckt till din nya meny. Åk ner i menyn så visar sig en tom grå ruta under menyn. När du får ett svart streck på den grå rutan släpper du musknappen.  8. Ändra namn på menyvalet på samma sätt som punkt 5 ovan.  9. Ändra ikon på menyn på samma sätt som för en verktygsknapp. |  |

För att välja om du vill ha både text och/eller en egen ikon i din meny gör du så här:

|  |  |
| --- | --- |
| 1. Se till att din nya knapp är markerad (har en svart ram)  2. Klicka på knappen **Ändra markering** (Modify selection) och välj **Standardinställningar** (Default style) om du bara vill ha både ikon och text.  Välj **Endast text** (Text only) om du bara vill ha text. |  |

## Skapa en formulärknapp

Ett alternativ till att skapa egna verktygsknappar och menyval är att skapa en formulärknapp som du placerar i kalkylbladet. En sådan knapp ligger lokalt i kalkylbladet vilket innebär att den följer med om du flyttar eller kopierar arbetsboken till annan plats. För att skapa en formulärknapp gör du så här:

|  |  |
| --- | --- |
| 1. Högerklicka var som helst bland verktygsknapparna och välj **Formulär** (Forms) i snabbmenyn.  2. Klicka på **Knapp** (Button) på verktygsfältet **Formulär** (Forms). OBS! Håll inte ner musknappen och dra.  3. Placera pekaren där du vill placera din knapp, håll ner vänster musknapp och dra ut en rektangel till önskad storlek. När du släpper musknappen får du en knapp och dialogrutan **Koppla makro** (Assign macro) visas automatiskt.  4. Markera det makro du vill koppla till knappen och klicka OK.  5. Markera texten på knappen och skriv in önskad text. Klicka vid sidan av knappen för att aktivera den.  Om du vill ändra text på en knapp gör du detta enklast genom att högerklicka på knappen, klicka på den streckade kanten på knappen och skriva om texten.  Vill du koppla ett annat makro till en befintlig knapp gör du detta genom att högerklicka på knappen och välja **Koppla makro till knapp** (Assign macro). Markera sedan önskat makro och klicka **OK**. |  |

# 

# Att skriva egen kod

Nu har vi talat om vad ett makro är, hur du spelar in ett makro och kopplar det till en knapp eller meny samt hur miljön i Visual Basic ser ut. Nu är det dags att "get your hands dirty", det vill säga gå in på hur programmeringsspråket Visual Basic är uppbyggt och hur du tänker när du skall skapa dina program. Detta kapitel är inte en uttömmande beskrivning av språket utan tar endast upp grundläggande begrepp som du behöver för att skriva dina första egna makron.

## I huvudet på en programmerare?

När du skall skriva egna makron är det viktigt att du tänker på ett strukturerat sätt. Du har säkert en idé om vad ditt makro skall åstadkomma. Försök att definiera i sekventiella steg vad som skall ske. Skriv gärna upp dessa steg med papper och penna.

Låt oss ta ett exempel.

Du har bestämt dig för att skapa ett makro som beräknar din slutskatt och nettolön samt din totala skatteprocent. Makrot skall fungera oavsett hur stor eller liten bruttolönen är. Det är lämpligt att börja med att definiera *indata*, *åtgärder och beräkningar* samt *utdata* för makrot i just den ordningen.

Du måste börja med *indata* för att ha värden att utföra beräkningar med. Du måste ha gjort *beräkningarna* för att ha någon *utdata* att visa.

*Vad behöver du för att kunna göra beräkningarna (Indata)?*

Du behöver veta Bruttolön, Procentvärden för kommunalskatt och statlig skatt samt värnskatt. Du behöver också veta var brytpunkterna ligger för den statliga skatten och värnskatten.

*Hur skall makrot hämta dessa data?*

Från förbestämda celler i MS Excel

*Vad skatt makrot göra och beräkna(Åtgärder och beräkningar)?*

Makrot skall kontrollera storleken på bruttolönen och beroende på bruttolönens storlek räkna ut skatten. Därefter skall nettolön och total skatteprocent beräknas.

*Vilka beräkningsregler gäller för skatt?*

Om bruttolönen är 25000 kr eller mindre skall endast kommunalskatt på hela bruttolönen beräknas.

Om bruttolönen är större än 25000 kr och mindre än 35000 kronor skall kommunalskatt på hela bruttolönen samt statlig skatt med 20 % på den del av bruttolönen som överstiger 25000 kr beräknas.

Om bruttolönen är större än 35000 kr skall kommunalskatt på hela bruttolönen, statlig skatt med 20 % på den del av bruttolönen som överstiger 25000 kr samt värnskatt med 5 % på den del av bruttolönen som överstiger 35000 kr beräknas.

*Hur skall beräknade värden visas (Utdata)?*

Värdena skall läggas ut i bestämda celler i MS Excel.

Nu har du definierat vad makrot skall göra samt i vilken ordning det skall ske (indata – Åtgärder och beräkningar – Utdata) är det dags att fundera på hur du skall skriva koden.

## Beståndsdelar i programmeringsspråket Visual Basic

För att kunna omvandla exemplet ovan till programmeringskod måste vi först titta närmare på beståndsdelarna i programmeringsspråket Visual Basic. Det finns vissa grundläggande element som är gemensamma i alla programmeringsspråk, t ex användandet av *variabler* och *konstanter* för att lagra värden under körningen, villkorssatser som *IF-Than-Else* och *Select – Case* samt repeterande satser, så kallade loopar såsom *For-Next*, *Do-While*, *Do-Until*, som i MS Excel bland annat kan användas till att stega igenom en lista med data.

### Procedurer och funktioner

Som vi tidigare har konstaterat finns det dessutom två grundläggande strukturer, *Procedurer* och *Funktioner*.

Om du vill köra ett Makro med hjälp av en knapp måste du använda en procedur. Du bör använda procedurer till att stega igenom en lista eller till att lägga ut flera värden i olika celler. En procedur har följande struktur:

Sub Procedurnamn()

End Sub

Funktioner är användbara när du vill utföra beräkningar där endast ett värde skall returneras. En Funktion tar ett eller flera invärden (parametrar) och returnerar ett värde. Funktioner kan även användas till att kontrollera om ett villkor är sant eller falskt, det vill säga utföra logiska kontroller. Funktioner kan anropas från procedurer eller användas direkt i Excel. En funktion har följande grundstruktur:

Function Funktionsnamn(parameter1, parameter2 ...)

Funktionsnamn = "värde som returneras"

End Function

### Variabler och konstanter

Variabler är oumbärliga för en programmerare eftersom de används för att lagra värden som till exempel hämtas från ett kalkylblad. En variabel är en behållare som du använder till att lagra och ändra värden.

Du kan ange vilken typ av värde du vill lagra i en viss variabel (ange datatyp). Detta innebär att du kan begränsa vad en viss variabel skall användas till. Variabler deklareras i regel i början av en procedur eller funktion och tilldelas ett värde längre ner i koden på önskat ställe.

Variabler kan deklareras och tilldelas ett värde på följande sätt:

Dim MinVariabel1 En variabel som kan lagra alla typer av värden, d v s både tal och text. Denna datatyp kallas *Variant* och

Dim MinVariabel1 As Variant kan deklareras på de två sätten du ser till vänster.

MinVariabel1 = "En textsträng" Du ger en variabel ett värde genom att skriva

MinVariabel1 = 42 variabelnamnet följt av ett likhetstecken och ett

värde (Tilldelning)

Dim MinVariabel2 As **Byte** Deklaration av en variabel som kan lagra positiva heltal mellan 0 och 255.

MinVariabel2 = 150 Tilldelning av ett tillåtet värde.

MinVariabel2 = -25 Otillåten tilldelning som ger ett körningsfel.

MinVariabel2 = "Bosse" Otillåten tilldelning som ger ett körningsfel.

Dim MinVariabel3 As Integer Deklaration av en variabel som kan lagra positiva och negativa heltal mellan -32768 och 32768.

MinVariabel3 = 150 Tilldelning av ett tillåtet värde.

MinVariabel3 = -25000 Tilldelning av ett tillåtet värde.

MinVariabel3 = 50000 Otillåten tilldelning som ger ett körningsfel.

MinVariabel3 = "Bosse" Otillåten tilldelning som ger ett körningsfel.

Dim MinVariabel4 As Long En variabel som kan lagra positiva och negativa heltal mellan -2 147 483 648 och 2 147 483 648.

MinVariabel4 = 1 500 000 Tilldelning av ett tillåtet värde.

MinVariabel4 = -10 250 000 Tilldelning av ett tillåtet värde.

MinVariabel4 = "Bosse" Otillåten tilldelning som ger ett körningsfel.

Dim MinVariabel5 As Single En variabel som kan lagra positiva och negativa decimaltal (flyttal med enkel precision) från 3.402823E38 till -1.401298E-45 för negativa tal och 1.401298E-45 till 3.402823E38 för positiva tal. Mycket stora tal kan lagras med denna datatyp.

MinVariabel5 = 1 500 000,0002 Tilldelning av ett tillåtet värde.

MinVariabel5 = -10 250,329002 Tilldelning av ett tillåtet värde.

MinVariabel5 = "Bosse" Otillåten tilldelning som ger ett körningsfel.

Dim MinVariabel6 As Double En variabel som kan lagra positiva och negativa decimaltal (flyttal med dubbel precision) mellan -1.79769313486231E308 till

-4.94065645841247E-324 för negativa tal och 4.94065645841247E-324 till 1.79769313486231E308 för positiva tal. Mycket stora tal kan lagrar med denna datatyp.

MinVariabel6 = 1 500 000,0002 Tilldelning av ett tillåtet värde.

MinVariabel6 = -10 250,329002 Tilldelning av ett tillåtet värde.

MinVariabel6 = "Bosse" Otillåten tilldelning som ger ett körningsfel.

Dim MinVariabel7 As Date Deklaration av en variabel som kan lagra tal i datumformat

MinVariabel7 = "2006-02-01" Tilldelning av ett tillåtet värde.

MinVariabel7 = -25 Tilldelning av ett tillåtet värde. Ger datumet

1899-12-05. (Heltalet 1 ger 1900-01-01)

MinVariabel7 = "Bosse" Otillåten tilldelning som ger ett körningsfel.

Dim MinVariabel8 As String Deklaration av en variabel som kan lagra alla tecken som en textsträng (alfanumeriskt). Strängvariabler är lämpliga vid manipulering av enskilda tecken i ett ord. Det går att utföra beräkningar på variabler av typen String som innehåller tal. Du kan däremot inte jämföra variabler av typen String med andra typer av variabler såsom till exempel integer.

MinVariabel8 = "2006-02-01" Tilldelning av ett tillåtet värde.

MinVariabel8 = -25 Tilldelning av ett tillåtet värde.

MinVariabel8 = "Bosse" Tilldelning av ett tillåtet värde.

Konstanter tilldelas ett värde direkt vid deklarationen. De kan ha samma datatyper som variabler. Tanken med att använda konstanter är att om du behöver kunna ändra ett visst värde som används på många ställen i koden görs ändringen endast på ett ställe, nämligen i deklarationen.

Konstanter kan deklareras och tilldelas ett värde på följande sätt:

Const Ljusår = 300000

Const Ljusår2 As Long = 300000

Const Pi As Single = 3.1415

### Inbyggda funktioner i språket

Det finns en stor mängd inbyggda funktioner för olika ändamål i Visual Basic. Några är generella medan andra är specifika för MS Excel. Nedan följer några användbara exempel.

|  |  |
| --- | --- |
| InputBox()  Variabel = InputBox("Ange ett värde") | Funktion för indata som genererar en dialogruta. Värdet som användaren skriver i fältet sparas i en variabel. |
| MsgBox  MsgBox"Detta är ditt värde: " & Variabel  Val = MsgBox("Vill du avsluta?",vbOKCancel) | Funktion för utdata som visar ett textmeddelande eller ett värde, eller som i exemplet till vänster både och.    Funktionen kan även användas till att ta emot ett inputvärde, t ex ett klick på OK eller Avbryt. Du använder parenteser i funktionen. |
| isNumeric(Variabel)  Not isNumeric(Variabel) | En logisk funktion som kan användas för att kontrollera om en variabel innehåller ett numeriskt värde eller inte. Funktionen returnerar Sant om värdet i variabel är numeriskt, annars returneras Falskt. Kan även föregås av *Not* som vänder på logiken. |
| Range()  Värde = Range("A1") 'Hämta värde  Range("B2") = Värde 'Lägg ut värde  Range("B1:A5").Select 'Markera område | Mycket användbar Excel-specifik funktion som kan användas till att hämta ett värde ur en cell, lägga ut ett värd till en cell, eller markera en cell eller ett område. Funktionen kan även användas i en så kallad For-Each loop. Detta kommer vi till senare |
| CInt(Variabel)  Int(Variabel)  CLng(Variabel)  Cdate(Variabel)  CDbl(Variabel) | Det finns en mängd funktioner för att omvandla (konvertera) ett värde till en annan datatyp.  Konverterar och avrundar värde till heltal.  Konverterar värde till heltal och kapar decimalerna.  Konverterar och avrundar värde till långt heltal.  Konverterar värde till datumformat.  Konverterar värde till stort decimaltal. |
| ActiveCell.Offset(Rad,Kolumn)  ActiveCell.Offset(0,1).Select  ActiveCell.Offset(1,0) = värde  ActiveCell.Offset(-1,-2) = värde \* värde | En funktion som används för att markera eller hämta resp. lägga ut ett värde i en cell vid sidan av den aktiva cellen.  Markerar en cell till höger om den aktiva cellen (som är markerad)  Lägger ut ett värde i den cell som ligger under den aktiva cellen.  Lägger ut ett värde i den cell som ligger en cell ovanför och två till vänster om den aktiva cellen |

### Villkorssatser

Villkorssatser är en oumbärlig del av programmeringsvärlden eftersom du ofta vill göra något med ett värde beroende på olika faktorer.

De finns två grundläggande villkorssatser: *If-Then-Else* och *Select-Case*. Vad du väljer är en fråga om tycke och smak eftersom de är likvärdiga.

Om du har väldigt många olika villkor som kan vara uppfyllda för en variabel kan *Select-Case* vara enklare att använda. Å andra sidan, om du är van att använda *Om*-funktionen i MS Excel kanske *If-Then-Else* känns enklare.

Några exempel på hur *If-Then-Else* och *Select-Case* kan skrivas. Case-satsen i högra kolumnen gör samma sak som If-satsen i vänstra kolumnen.

|  |  |
| --- | --- |
| If-sats | Case-sats |
| If KollVärde > 0 Then Värde2 = 100 | Select Case KollVärde  Case Is > 0  värde2 = 100  End Select |
| If KollVärde > 0 Then  Värde2 = 100  Else  Värde2 = 50  End if | Select Case KollVärde  Case Is > 0  värde2 = 100  Case Is <= 0  värde2 = 50  End Select |
| If KollVärde > 10000 Then  Värde2 = 100  ElseIf KollVärde > 5000 Then  Värde2 = 50  Else  Värde2 = 0  End if | Select Case KollVärde  Case Is > 10000  värde2 = 100  Case 5000 To 10000  värde2 = 50  Case Else  värde2 = 0  End Select |

### Vårt exempel med skatteberäkning skrivet i kod

I avsnittet "I huvudet på en programmerare" formulerade vi stegvis hur vi skulle kunna skapa ett program som beräknar skatt, nettolön och totalskatteprocent utifrån angiven bruttolön och kommunalskatteprocent. Nu har du gått igenom nödvändiga kodstrukturer för att kunna skapa detta program utifrån tidigare angiven problemdefinition.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C |  | Själva kalkylen i Excel skulle kunna se ut som till vänster.  Värdena på bruttolön och kommunal­skatt anges av användaren i cellerna C1 och C2.  Makrot beräknar sedan Skatt(%), Skatt och Nettolön och lägger ut dessa i C3, C4 och C5 i kalkylbladet. |
| 1 | Indata | Bruttolön | 33 000 kr |  |
| 2 | Indata | Kommunalskatt | 32 % |
| 3 | Utdata | Skatt (%) | 37 % |
| 4 | Utdata | Skatt | 12 160 kr |
| 5 | Utdata | Nettolön | 20 840 kr |

Sub Löneberäkning()

' Deklarationer av konstanter för statlig skatt och värnskatt

Const statSkatt = 0.2

Const vSkatt = 0.05

' Deklarationer av variabler för att hämta kommunalskatt och bruttolön

' från Kalkylbladet samt en variabel för att uträknad totalskatt

Dim Brutto As Long, skatt

Dim KomSkatt As Single

' Hämta värden från celler i kalkylbladet

Brutto = Range("C2")

KomSkatt = Range("C3")

' Beräkna totalskatt beroende på bruttolönens storlek

If Brutto > 35000 Then

skatt = (Brutto \* KomSkatt) + \_

((Brutto - 25000) \* statSkatt) + \_

((Brutto - 35000) \* vSkatt)

ElseIf Brutto <= 35000 And Brutto > 25000 Then

skatt = (Brutto \* KomSkatt) + \_

((Brutto - 25000) \* statSkatt)

Else

skatt = (Brutto \* KomSkatt)

End If

' Lägg ut skatteprocent, skatt och nettolön på utvald plats i kalkylbladet

Range("C4") = skatt / Brutto

Range("C5") = skatt

Range("C6") = Brutto - skatt

End Sub

### Repeterande satser (Loopar)

Nästa oumbärliga programstruktur är repeterande satser eller loopar som det även kallas. En loop är som namnet antyder en sats som repeterar en åtgärd ett visst antal gånger. Antingen ett angivet antal repetitioner eller tills ett visst villkor är uppfyllt.

Loopar är extremt användbara när du vill stega igenom flera rader i en lista för att där göra beräkningar eller utföra åtgärder. Det finns två grundläggande typer av loopar i Visual Basic. Do-loopar som repeterar utifrån ett angivet villkor, och For-loopar som repeterar ett givet antal gånger.

Nedan följer några exempel på de loopar som finns tillgängliga.

*For-loopar (Används vanligtvis när man vet hur många gånger loopen skall repetera):*

For i = 1 to 100

'Repeterar 100 gånger, variabeln i ökas med 1 för varje repetition

*'Programkod*

Next i

For i = 10 to 1 Step -1

'Repeterar 10 gånger, variabeln i minskas med 1 för varje repetition

*'Programkod*

Next i

*Nedanstående loop är extremt praktisk när man stega igenom en Excellista.*

For Each Värde In Range(”A:A”)

'Stegar igenom A-kolumnen cell för cell, variabeln Värde används för att

'komma åt cellinnehållet

'*Programkod*

Next Värde

*Do-loopar (Lopar med en villkorsdel. Om villkoret aldrig blir uppfyllt får man en evighetsloop):*

Do Until Värde > 100

'Repeterar tills variabeln Värde är större än 100

'*Programkod*

Loop

Do While Värde <= 100

'Repeterar tills variabeln Värde är större än 100

'*Programkod*

Loop

### Funktionsanrop från procedurer

Om du vill använda en egen funktion genom att trycka på en knapp måste du först skapa en procedur som anropar funktionen. Detta eftersom knappar bara kan köra procedurer.

Exempel: Låt säga att du sedan tidigare har en funktion som räknar om Euro, US Dollar eller Brittiska Pund till kronor. Du vill skapa en knapp för att använda funktionen. Funktionen kan se ut som nedan.

Function ValutaOmvandling(ValutaBelopp, Kurs)

ValutaOmvandling = ValutaBelopp / Kurs

End Function

För att kunna köra den med en knapp skapar du en procedur som anropar funktionen.

Sub KollaValuta()

Dim Belopp As Double, ValutaSlag As String, ValutaKurs As Single, Anrop

' Be användarer ange Belopp Valuta och kurs

Belopp = InputBox("Ange belopp i svenska kronor")

ValutaSlag = InputBox("Ange önskat valutaslag: Euro, USD eller GBP")

ValutaKurs = InputBox("Ange valutakurs som ett decimaltal")

'Kolla att en giltig valuta angivits

If ValutaSlag <> "Euro" And ValutaSlag <> "USD" And ValutaSlag <> "GBP" Then

MsgBox "Du måste ange Euro, USD eller GBP som valuta", vbExclamation

Exit Sub

Else

'Anropa funktionen och visa resultatet.

'Valiabeln Anrop används för funktionsanropet.

'Den tar även emot värdet som funktionen returnerar

Anrop = ValutaOmvandling(Belopp, ValutaKurs) 'Funktionsanrop

MsgBox Belopp & " SEK blir " & Round(Anrop, 2) & " " & ValutaSlag

End If

End Sub

### Strukturerad programmering

Om du du vill skapa mer omfattande applikationer i Excel är det bra att dela in koden i funktionella enheter. Vanligtvis skapar man en huvudprocedur som anropar funktionella underprocedurer. Detta gör att du får en överblick över ditt program samt det blir lättare att testa, felsöka och underhålla koden. Ett enkelt exempel på detta sätt att dela in ett program i funktionella enheter kan ses nedan.

Sub Huvudprogram()

HanteraIndata

GörBeräkningar

HanteraUtdata

End Sub

Sub HanteraIndata()

'*Programkod*

End Sub

Sub GörBeräkningar()

'*Programkod*

End Sub

Sub HanteraUtdata()

'*Programkod*

End sub

## Skapa makron i Excel i praktiken

När du utvecklar makron i Excel är det praktiskt att kombinera inspelad kod med egen skriven kod. Detta eftersom det går mycket fortare att spela in till exempel olika formateringar av celler, eller objekt som diagram, än att försöka leta reda på hur man skriver denna kod manuellt. Du kommer sannolikt att vilja ha full kraft i dina makron och det enda sättet att få detta är att lära sig en smula om hur man skriver egen programmeringskod.

När du kombinerar inspelad kod med egenhändigt skriven kod får du snabbt ett färdigt program som dessutom har maximal funktionalitet.

I den sista övningen under övningsuppgifter finns ett lite längre exempel på hur ett sådant program skulle kunna byggas upp och se ut.

# 

# Övningsuppgifter

Försök att lösa nedanstående uppgifter utan att titta på lösningsförslagen. Om du tycker att det är svårt att komma igång kan du titta på lösningarna till de första exemplen och sedan försöka lösa de övriga uppgifterna. Alla uppgifter finns i arbetsboken *Uppgifter.xls*. Det finns lösningsförslag i arbetsboken *Lösningsförslag uppgifter.xls*

Du kan även skriva in lösningsförslagen och testa att ändra i koden. Det kan vara ett bra sätt att förstå vad koden faktiskt gör. *Kom ihåg att envishet är en programmerares bästa vän!*

## Uppgift 1: Spela in Makro och koppla till egen knapp

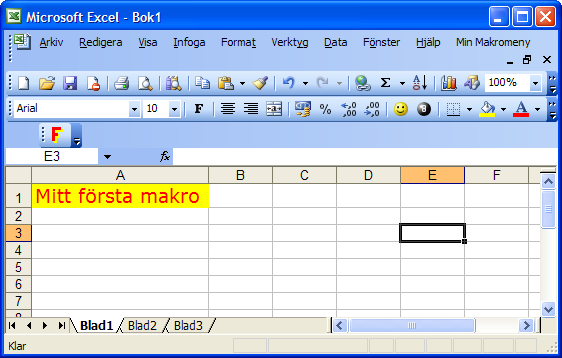
Spela in ett makro som ändrar format på markerad cell/markerade celler enligt följande:

Teckensnitt: *Verdana*, Teckenstorlek: *14*, Teckenfärg: *Röd*, Fyllningsfärg: *Gul*

Makrot skall även autopassa *kolumnbredden* för markerad cell/markerade celler.

Skapa ett nytt verktygsfält med en egen knapp och koppla makrot till knappen. Skapa en egen ikon för knappen.

Ta bort all ”extra” kod i makrot som inte gäller det du valt när du spelade in makrot. Testa knappen.



## Uppgift 2: Funktion som beräknar skatten

Skapa en ny funktion i Visual Basic Editor som tar bruttolön och kommunalskatt som inparametrar och returnerar skatt som skall betalas. Ge funktionen namnet Skatt.

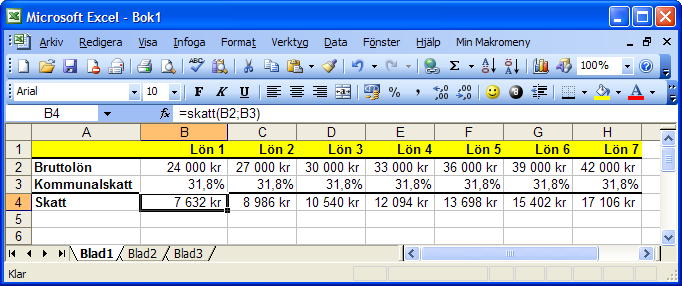
Själva skatteberäkningen skall ske enligt följande:

Kommunalskatt på hela bruttolönen

Statlig skatt med 20 % på den del av bruttolönen som överstiger 35000 Kr

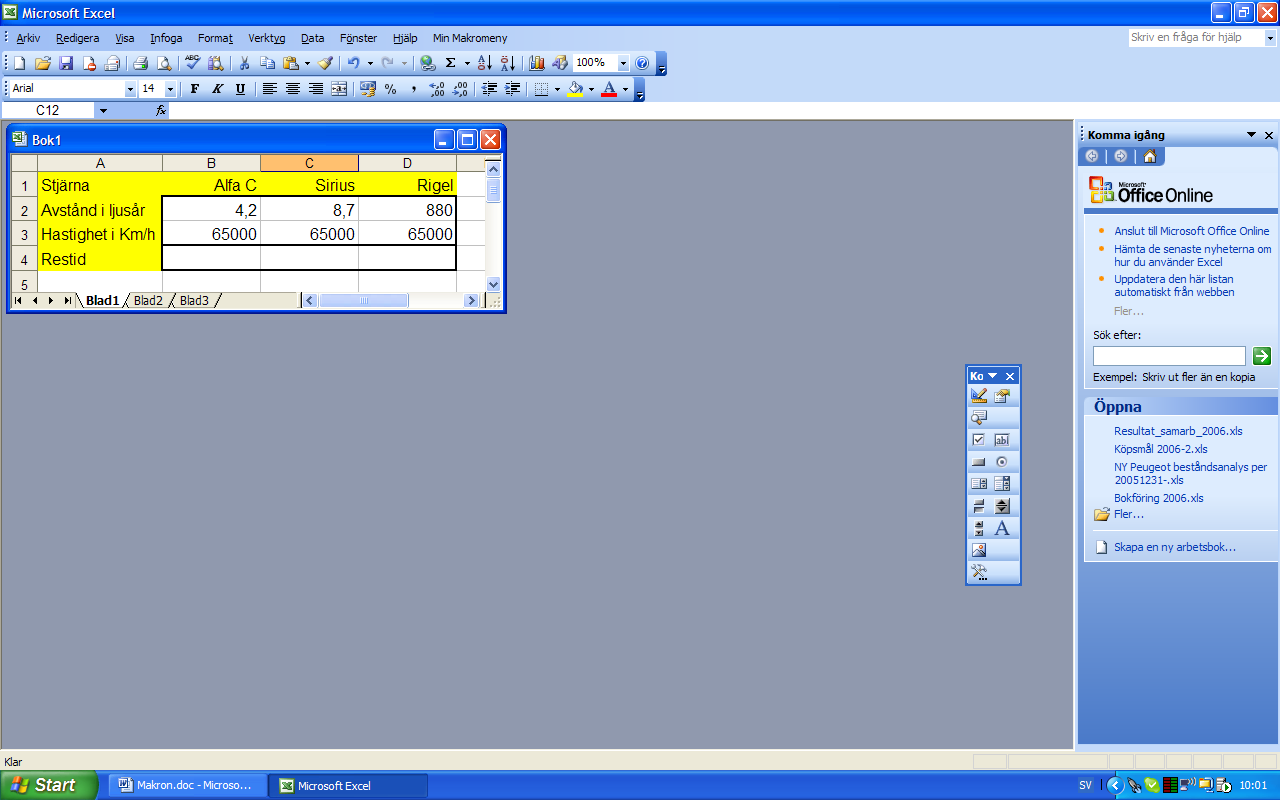
Värnskatt med 5 % på den del av bruttolönen som överstiger 45000 Kr

Skapa en kalkyl enligt bilden nedan och infoga funktionen i cellen B4. Autofyll sedan åt höger för att få skatten för alla löner.



## Uppgift 3: Funktion som beräknar restid till stjärnor

Skapa en ny funktion i Visual Basic Editor som beräknar restiden för olika stjärnor. Du kan börja med att skapa en ny arbetsbok enligt nedan.

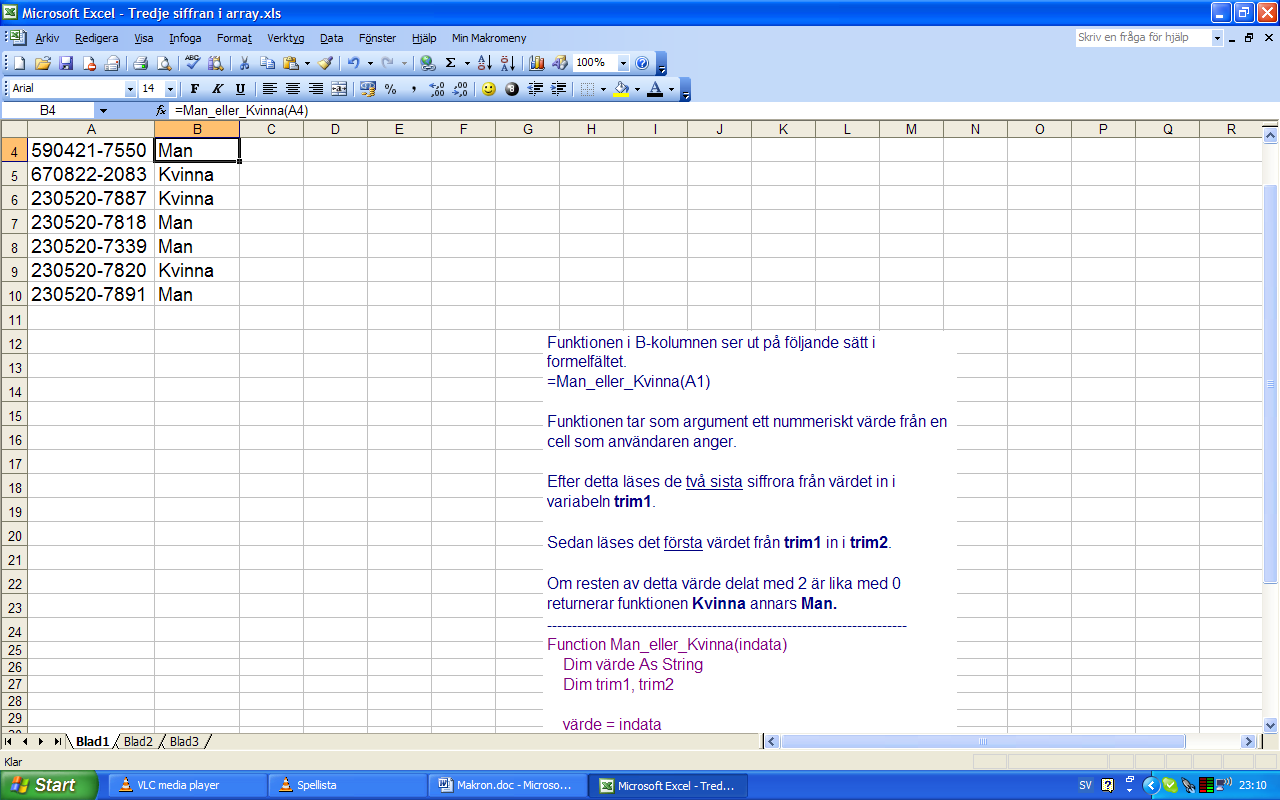


Skapa en funktion som returnerar restiden i år. Formeln för att beräkna restiden ser ut så här:

(Avstånd i ljusår \* ljushastighet \* 60 \* 60) / Hastighet i Km/h

## Uppgift 4: Funktion som bestämmer kön med hjälp av personnumret

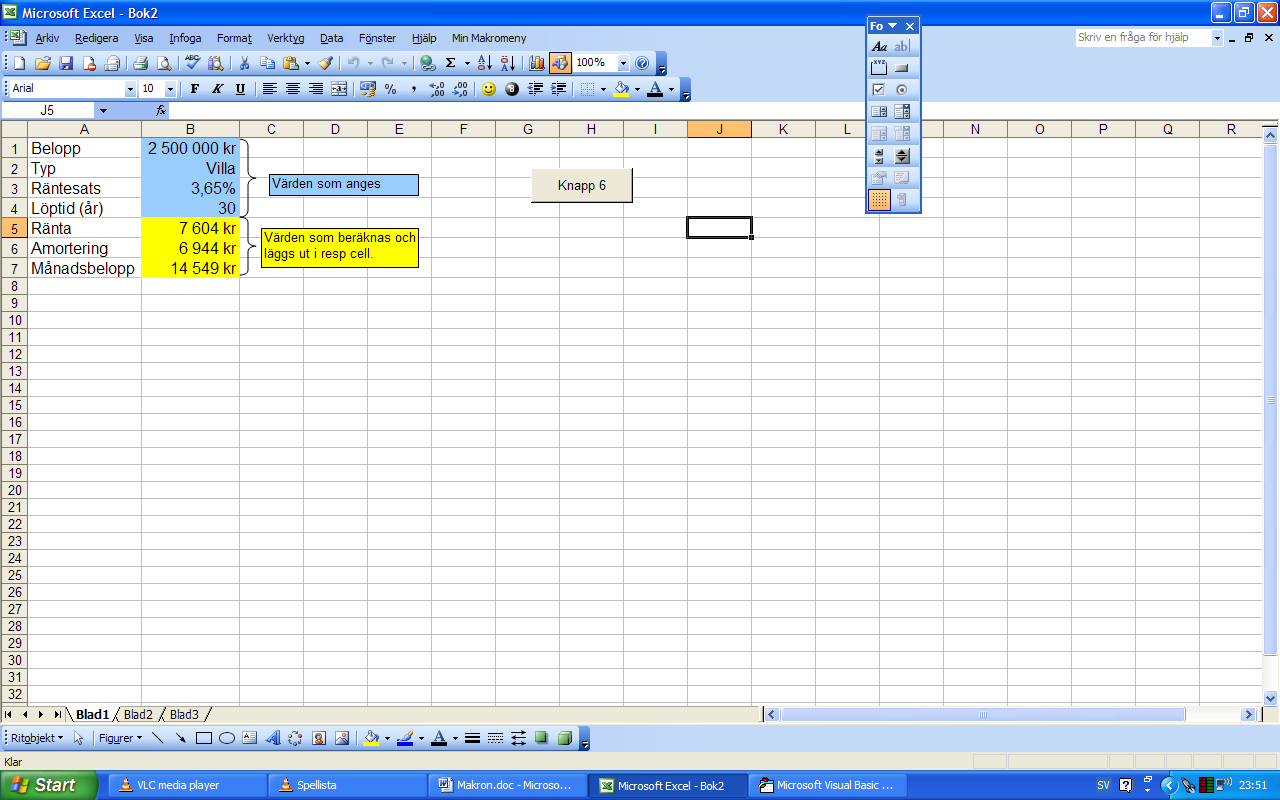
Skapa en funktion som tar ett personnummer som indata. Funktionen skall returnera ordet Man om näst sista siffran i personnumret är udda och ordet Kvinna om näst sista siffran i personnumret är jämn. Funktionen skall fungera oavsett hur personnumret är inskrivet.



## Uppgift 5: Procedur som beräknar bolån

Skriv ett eget procedurmakro som beräknar *Ränta*, *Amortering*, och *Månadsbelopp* på ett bolån. Du skall ange *Belopp*, *Typ av bostad*, *Räntesats* samt *Löptid* i år. Excelbladet kan se ut som nedan.

Endast *Villa* eller *Bostadsrätt* skall vara tillåtet för *Typ*.

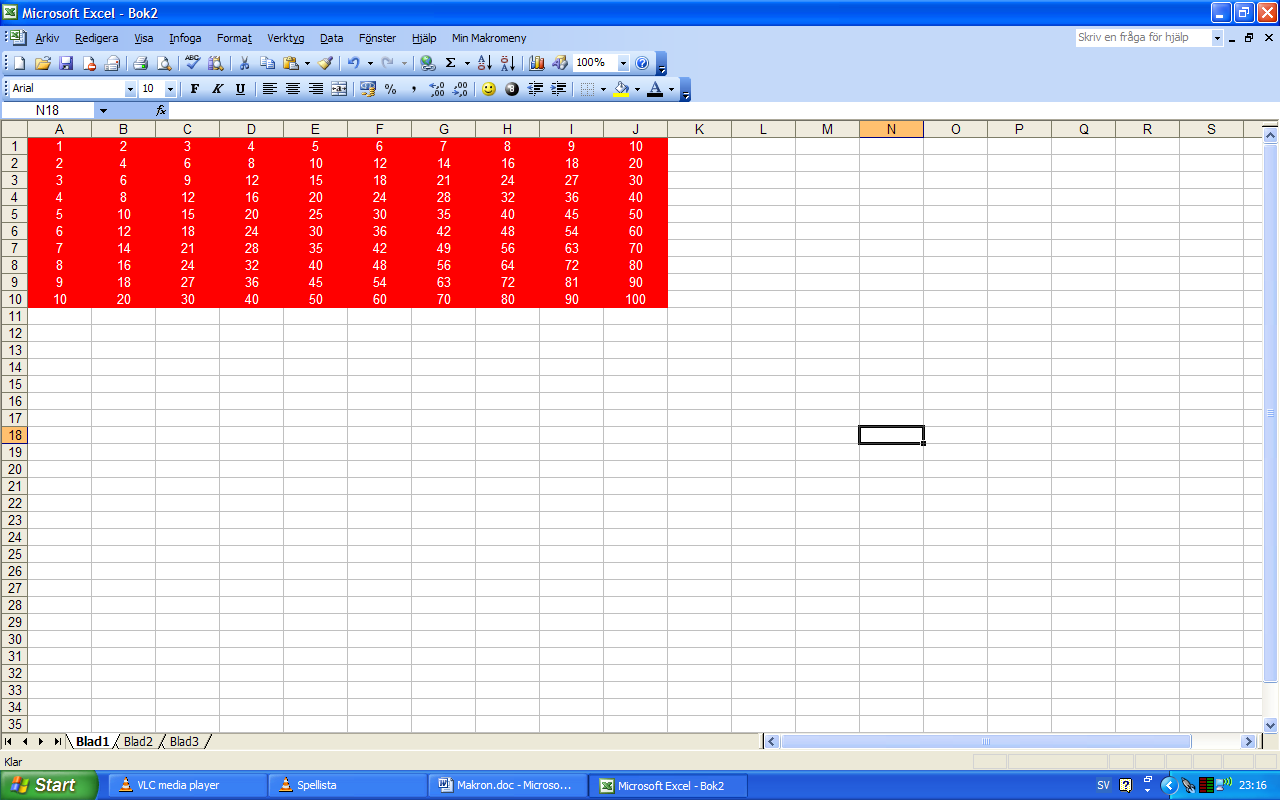


Formel för att beräkna månadsränta: Belopp \* Räntesats / 12

Formel för att beräkna månadsamortering: Belopp / Löptid / 12

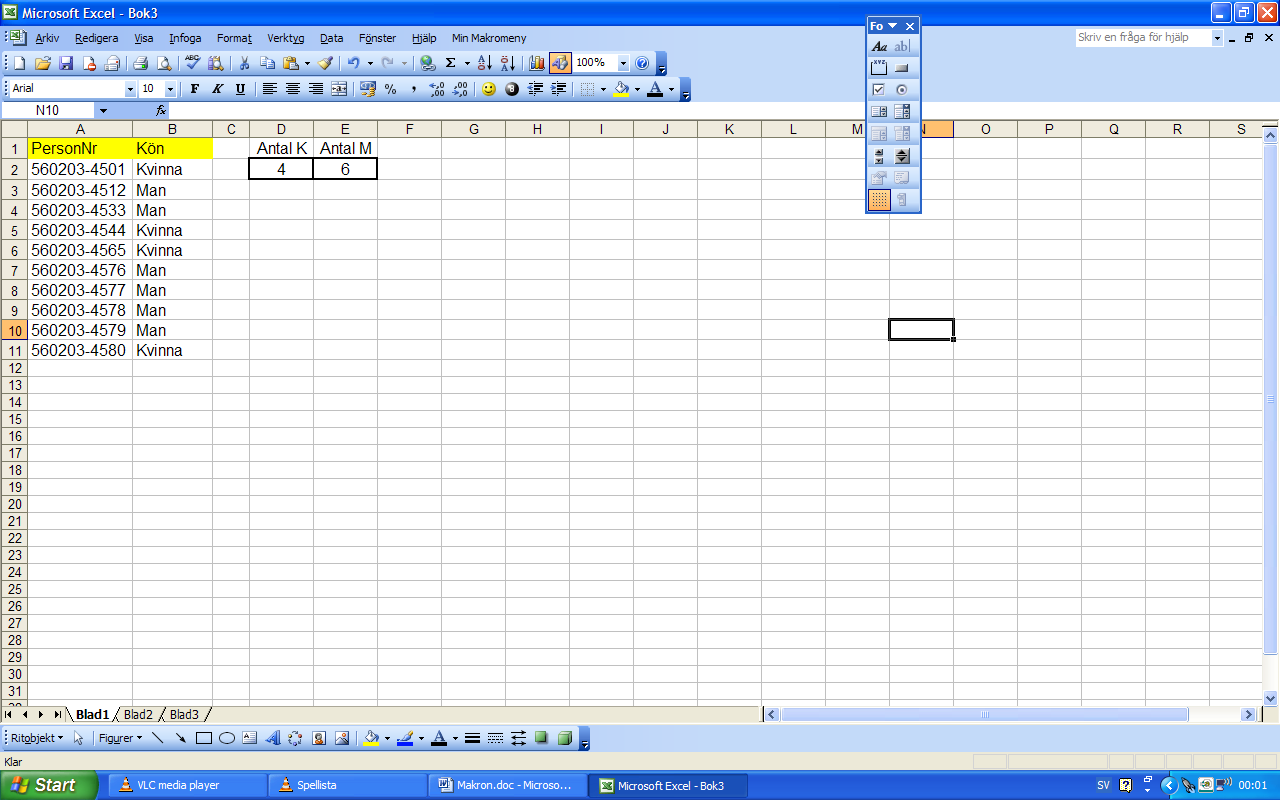
## Uppgift 6: Procedur som skapar multiplikationstabell

Skriv ett eget *procedurmakro* som skapar en multiplikationstabell enligt bilden nedan. Här behöver du använda loopar.



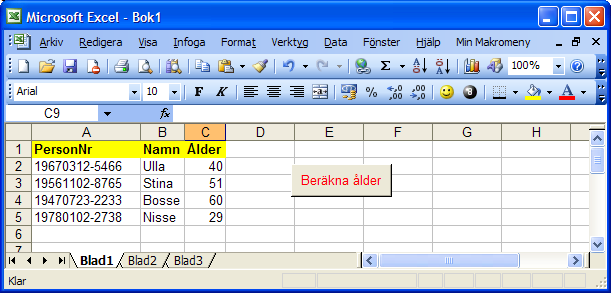
## Uppgift 7: Procedur som räknar antal kvinnor och män i en lista

Skapa nedanstående kalkyl om du inte har gjort uppgift 4. Skapa sedan en procedur som stegar igenom listan i B-kolumnen och räknar antalet Män och antalet Kvinnor. Dessa två värden skall sedan läggas ut i cellerna D2 och E2.



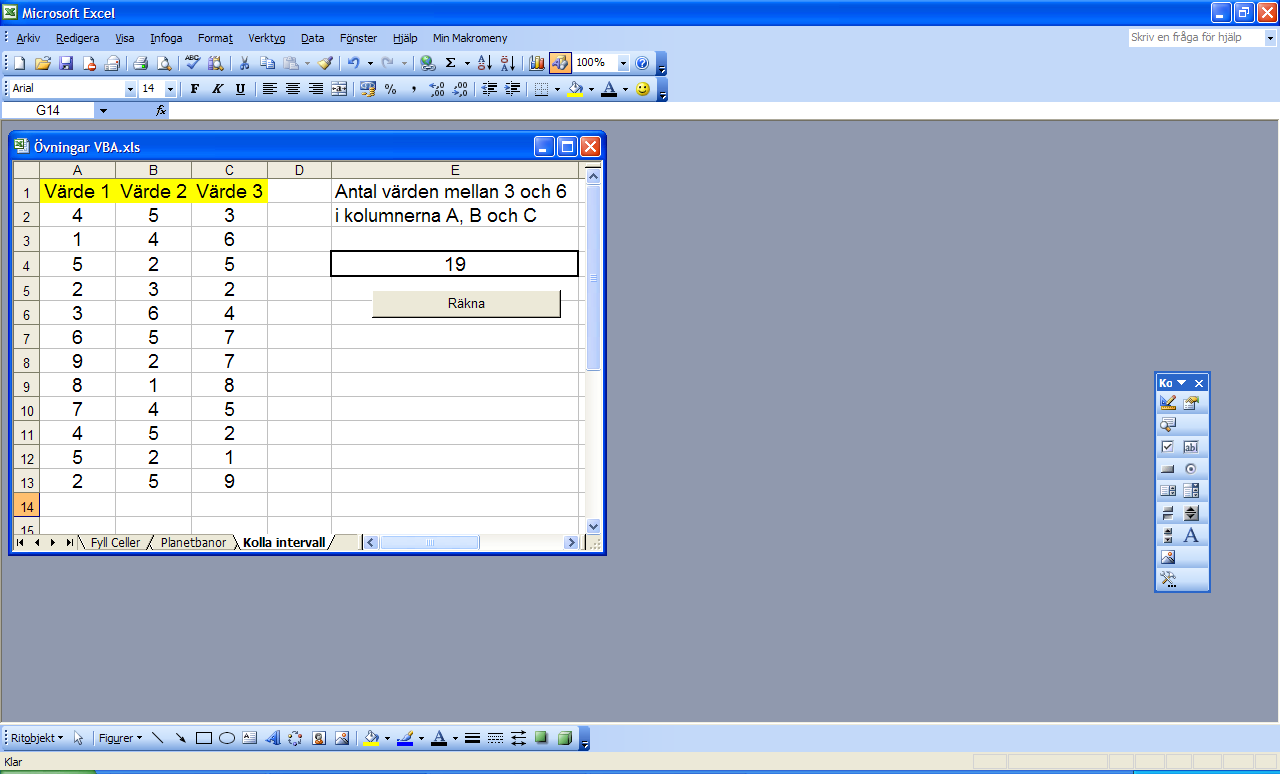
## Uppgift 8: Procedur som beräknar ålder på givna personnummer

I denna uppgift skall du skapa en procedur som stegar igenom en lista och beräknar åldern på personernas personnummer. Tänk på att du måste läsa in de fyra första tecknen från personnumret i en variabel och sedan konvertera värdet till ett tal för att kunna beräkna åldern. Det kan också vara lämpligt att bryta loopen när den träffar på en tom cell. Vi antar att det inte finns tomma celler i listan.



## Uppgift 9: Räkna antal värden i en lista

Skapa en kalkyl enligt bilden nedan. Skapa sedan en procedur som loopar igenom ett avgränsat område (A2 till C13) enligt bilden nedan. Proceduren skall räkna antalet värden mellan 3 och 6 och lägga ut resultatet i cellen E4.

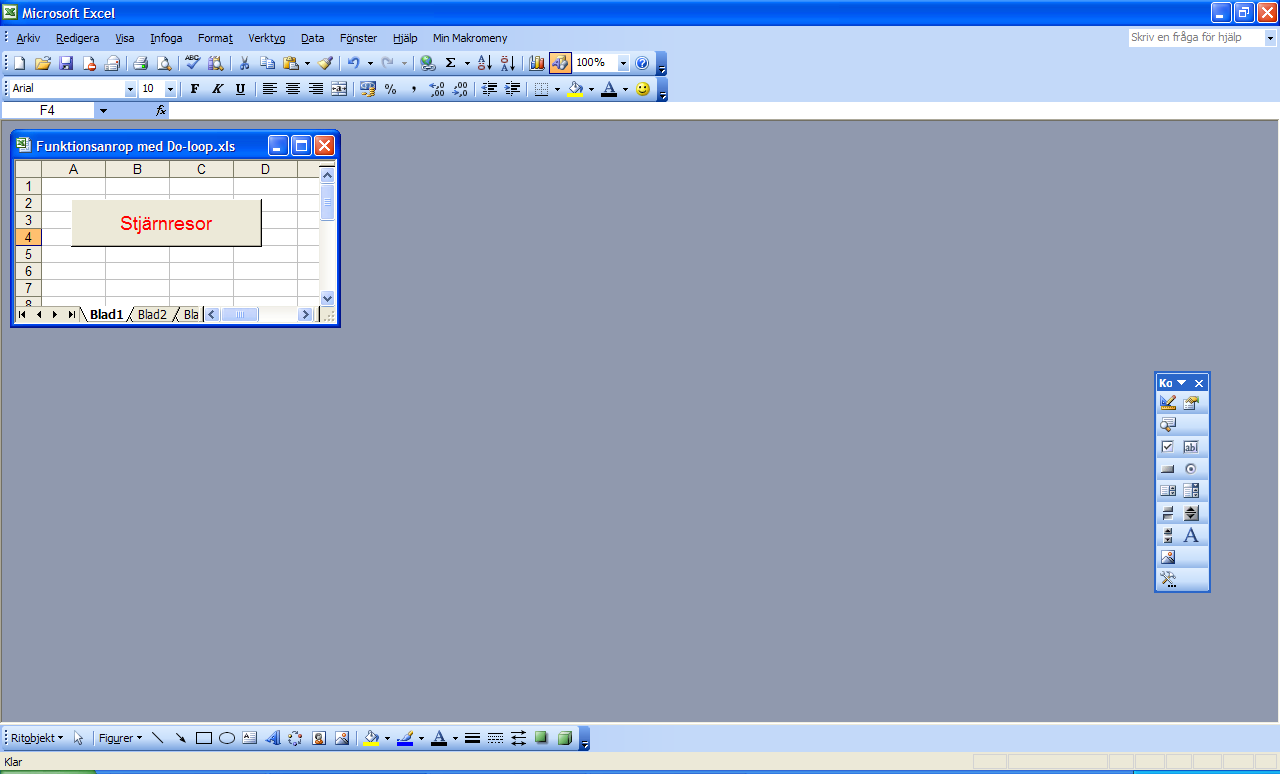


## Uppgift 10: Funktionsanrop från en Procedur

I denna uppgift skall du köra en funktion genom att klicka på en Formulärknapp. För att kunna göra detta måste du först skapa en procedur som är kopplad till knappen. Proceduren skall anropa funktionen stjärnresor (uppgift 4).

Användaren skall ange Stjärna, Avstånd i ljusår samt Hastighet i km/h. Använd indatarutor (InputBox) för att hantera användarens inmatning. Visa resultatet med en meddelanderuta (MsgBox)

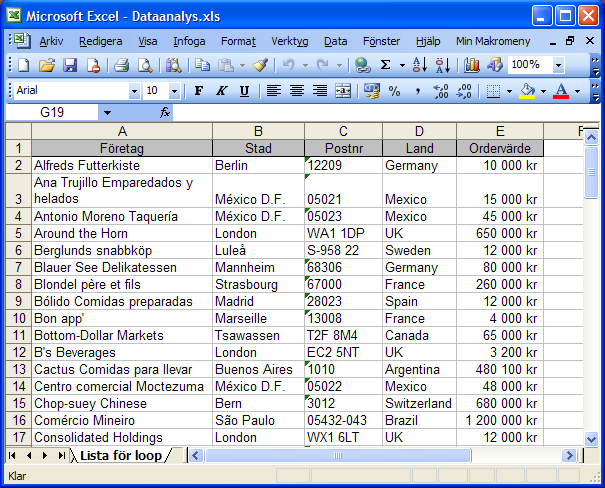
Skapa en formulärknapp som användaren klickar på för att köra.



## Uppgift 11: Procedur som beräknar ordervärdet för angivet land i en lista

Skapa en procedur där användaren anger ett land i en indataruta. Koden skall sedan stega igenom listan i D-kolumnen, hämta värden från E-kolumnen där landet i listan är lika med landet som användaren angivit och beräkna det totala ackumulerade värdet för angivet land.

Lägg ut Landet och totalvärdet i lämplig cell vid sidan av listan.



# 

# Lösningsförslag övningsuppgifter

## Uppgift 1: Spela in Makro och koppla till egen knapp

Sub Eget\_Format()

With Selection.Font

.Name = "Verdana"

.FontStyle = "Normal"

.Size = 14

.ColorIndex = 3

End With

With Selection.Interior

.ColorIndex = 6

.Pattern = xlSolid

.PatternColorIndex = xlAutomatic

End With

Columns("A:Z").EntireColumn.AutoFit

End Sub

## Uppgift 2: Funktion som beräknar skatten

Function skatt(Brutto, KomSkatt)

Const StatSkatt = 0.2

Const VSkatt = 0.05

' Beräkna totalskatt beroende på bruttolönens storlek

If Brutto > 35000 Then

skatt = (Brutto \* KomSkatt) + \_

((Brutto - 25000) \* StatSkatt) + \_

((Brutto - 35000) \* VSkatt)

ElseIf Brutto <= 35000 And Brutto > 25000 Then

skatt = (Brutto \* KomSkatt) + \_

((Brutto - 25000) \* StatSkatt)

Else

skatt = (Brutto \* KomSkatt)

End If

End Function

## Uppgift 3: Funktion som beräknar restid till stjärnor

Function Restid(Avstånd, Hastighet)

Dim AvstKM

Const Ljushast = 300000 'Km/s

AvstKM = Avstånd \* Ljushast \* 60 \* 60

Restid = AvstKM / Hastighet

End Function

## Uppgift 4: Funktion som bestämmer kön med hjälp av personnumret

Function Man\_eller\_Kvinna(indata)

Dim värde As String

Dim trim1, trim2

värde = indata

trim1 = Right(värde, 2)

trim2 = Left(trim1, 1)

If trim2 Mod 2 = 0 Then

Man\_eller\_Kvinna = "Kvinna"

Else

Man\_eller\_Kvinna = "Man"

End If

End Function

## Uppgift 5: Procedur som beräknar bolån

Sub Bolån()

'Variabeldeklarationer ------------------------------------

Dim Belopp, Typ, Löptid

Dim Ränta, Amort

'Indata ---------------------------------------------------

Belopp = Range("B1")

Typ = Range("B2")

Ränta = Range("B3")

Löptid = Range("B4")

'Beräkningar och åtgärder ---------------------------------

If Typ = "Villa" Then

Ränta = Belopp \* Vränta / 12

Amort = Belopp / Löptid / 12

ElseIf Typ = "Bostadsrätt" Then

Ränta = Belopp \* Bränta / 12

Amort = Belopp / Löptid / 12

Else

MsgBox "Du kan bara ange Villa eller Bostadsrätt", vbExclamation

Exit Sub

End If

'Utdata --------------------------------------------------

Range("B5") = Ränta

Range("B6") = Amort

Range("B7") = Ränta + Amort

End Sub

## Uppgift 6: Procedur som skapar multiplikationstabell

Sub Fyllceller()

Dim AntalCeller, räknare1, räknare2

AntalCeller = InputBox("Hur många celler skall fyllas?")

For räknare1 = 1 To AntalCeller

For räknare2 = 1 To AntalCeller

ActiveCell.Value = räknare1 \* räknare2

With Selection.Interior

.ColorIndex = 3

.Pattern = xlSolid

End With

Selection.Font.ColorIndex = 2

ActiveCell.Offset(0, 1).Select

Next räknare2

ActiveCell.Offset(1, -AntalCeller).Select

Next räknare1

End Sub

## Uppgift 7: Procedur som räknar antal kvinnor och män i en lista

Sub RäknaAntal()

Dim cellVärde, antalK, antalM

For Each cellVärde In Range("B:B")

If cellVärde.Value <> "" Then

If cellVärde.Value = "Kvinna" Then

antalK = antalK + 1

End If

If cellVärde.Value = "Man" Then

antalM = antalM + 1

End If

Else

Exit For

End If

Next cellVärde

Range("D2") = antalK

Range("E2") = antalM

End Sub

## Uppgift 8: Procedur som beräknar ålder på givna personnummer

Sub ålder()

Dim persNr As String, cellVärde, år

For Each cellVärde In Range("A2:A65000")

If cellVärde.Value <> "" Then

persNr = cellVärde.Value

år = CInt(Left(persNr, 4))

cellVärde.Offset(0, 2) = Year(Date) - år

Else

Exit Sub

End If

Next cellVärde

End Sub

## Uppgift 9: Räkna antal värden i en lista

Sub Kolla\_Värden()

Dim x, räknare

räknare = 0

For Each x In Range("A:C")

If x.Value <> "" Then

If IsNumeric(x.Value) Then

If x.Value <= 6 And x.Value >= 3 Then

räknare = räknare + 1

End If

End If

Else

MsgBox "Du är det klart", vbInformation

Range("E4") = räknare

Exit Sub

End If

Next x

End Sub

## Uppgift 10: Funktionsanrop från en Procedur

Sub Funktionsanrop()

Dim Anrop 'Variabel för anrop av funktion

Dim Stjärna, Avst, Hast 'Indatavariabler

Dim val

Do Until val = 2

'Indata

Stjärna = InputBox("Ange önskad stjärna")

Avst = CDbl(InputBox("Ange avståndet i ljusår"))

Hast = CLng(InputBox("Ange hastighet i km/h"))

'Funktionsanrop med beräkning

Anrop = Stjärnresor(Avst, Hast)

'Utdata

MsgBox "Restid till " & Stjärna & " blev " & Anrop & " år"

val = MsgBox("Vill du köra igen", vbOKCancel)

Loop

End Sub

Function Stjärnresor(Avstånd, Hastighet)

Dim AvstKM

Const Ljushast = 300000 'Km/s

AvstKM = Avstånd \* Ljushast \* 60 \* 60

Stjärnresor = AvstKM / Hastighet

End Function

## Uppgift 11: Procedur som beräknar ordervärdet för angivet land i en lista

Sub Ordervärden()

Dim CellVärde, Totalt, Land

' Indata

Land = InputBox("Ange önskat land")

Totalt = 0

For Each CellVärde In Range("D:D")

If CellVärde.Value <> "" Then

If Land = CellVärde.Value Then

Totalt = Totalt + CellVärde.Offset(0, 1)

End If

Else

Exit For

End If

Next CellVärde

' Utdata

Range("G5") = "Totalt för " & Land

Range("G6") = Totalt

End Sub