

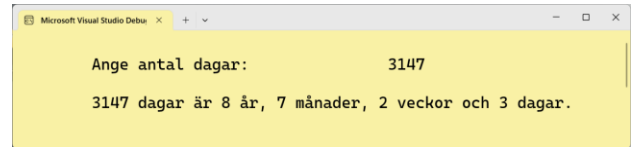
Välkomna alla SUV_x-23-are till **Boiler room fre 6/10, kl 9-12**. Här kommer mina två uppdrag till klassen för dagens Boiler room.

1. Uppdrag Days2Year

Lösningen **Year2Days** till [övn 4.8](#) (kursboken, [sid 110](#)) omvandlar tiden i antal år, månader och veckor till antal dagar. Man behöver inte ha löst denna uppgift, för att genomföra detta uppdrag.

Uppdraget går ut på att vända om problemet ovan, dvs i ett program – kalla det **Days2Year** – omvandla ett givet antal dagar till antal år, månader, veckor och resterande dagar.

Programmet ska ge följande utskrift när man vill omvandla t.ex. 3 147 dagar till antal år, månader, veckor och dagar:



```
Ange antal dagar:          3147
3147 dagar är 8 år, 7 månader, 2 veckor och 3 dagar.
```

Ni kan använda för denna omvandling följande algoritm och pseudokod:

Algoritmen:

1. Kalla den givna tiden i dagar för totaldagar.
2. Dividera totaldagar med 365 och strunta i resten, så får du det sökta antalet år.
3. Ta resten vid divisionen ovan. Dividera denna rest med 30 och strunta i resten så får du det sökta antalet månader.
4. Ta resten vid divisionen i punkt 3. Dividera denna rest med 7 och strunta i resten så får du det sökta antalet veckor.
5. Resten vid divisionen i punkt 4 är det sökta antalet resterande dagar.

Operationen *Dividera och strunta i resten* kallas för *heltalsdivision*.

Operationen *Ta resten vid heltalsdivision* heter *modulo*.

Läs om *modulo* i kursboken på [sid 97](#) eller på webben [här](#).

Pseudokoden:

1. Läs in antalet totaldagar.
2. år = totaldagar heltalsdividerad med 365
3. månader = (totaldagar modulo 365) heltalsdividerad med 30
4. veckor = ((totaldagar modulo 365) modulo 30) heltalsdividerad 7
5. Resterande dagar = ((totaldagar modulo 365) modulo 30) modulo 7

Uppdraget:

Utveckla ett C++ program som implementerar ovanstående algoritm och pseudokod.

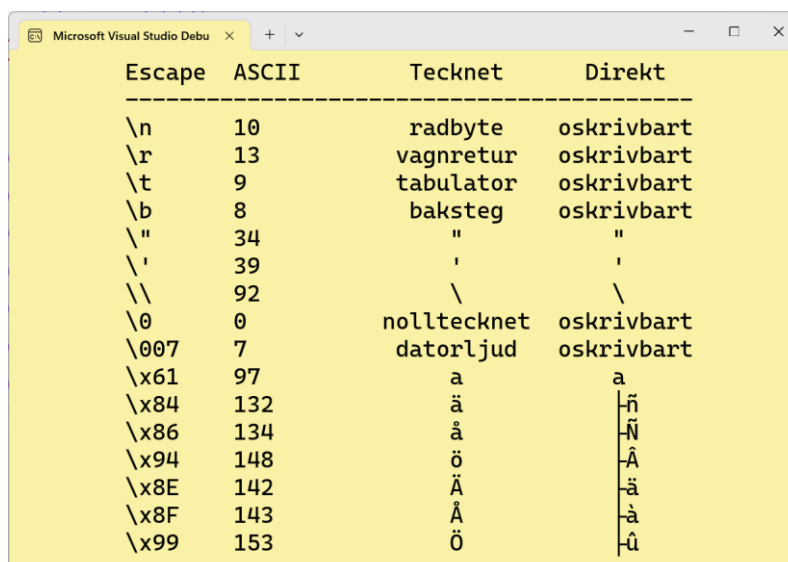
Skriv ut resultatet i användarvänlig from, se exemplet ovan.

Anmärkning:

Se upp med att använda de svenska tecknen ä, å, ö i namngivning av variabler. Testa om det är möjligt i resp. miljö.

2. Uppdrag EscapeTab

Skriv ett C++ program som skriver ut följande tabell över utvalda escapesequenser:



Escape	ASCII	Tecknet	Direkt
<code>\n</code>	10	radbyte	oskrivbart
<code>\r</code>	13	vagnretur	oskrivbart
<code>\t</code>	9	tabulator	oskrivbart
<code>\b</code>	8	baksteg	oskrivbart
<code>\"</code>	34	"	"
<code>\'</code>	39	'	'
<code>\\</code>	92	\	\
<code>\0</code>	0	nolltecknet	oskrivbart
<code>\007</code>	7	datorljud	oskrivbart
<code>\x61</code>	97	a	a
<code>\x84</code>	132	ä	ä
<code>\x86</code>	134	å	å
<code>\x94</code>	148	ö	ö
<code>\x8E</code>	142	Ä	Ä
<code>\x8F</code>	143	Å	Å
<code>\x99</code>	153	Ö	Ö

Kolumnerna innehåller följande:

- Tabellens första kolumn visar escapesekvensen i C++ kod.
- Den andra kolumnen visar ASCII-koden till det tecken som representeras av resp. escapesekvens.
- I den tredje kolumnen står själva tecknet, om det är skrivbart.
Om tecknet är oskrivbart står tecknets beteckning (betydelse) där.
- Den fjärde kolumnen visar hur de skrivbara tecknen skrivs ut i konsolen, om man skriver ut dem *direkt*, dvs utan escapesekvens.

Om de är skrivbara, men skrivs ut felaktigt, står där det felaktiga tecknet.

Om de är oskrivbara står där **oskrivbart**.

Även de svenska tecknen **ä, å, ö, Ä, Å** och **Ö** är med i tabellen samt någon vanlig bokstav, t.ex. **a**.

Den vanliga bokstaven **a** ska visa att *alla* tecken kan representeras med escapesekvenser.

Dvs escapesekvenser är ett parallellt system till den direkta användningen av teckentabellen som kan användas för att i olika miljöer ta reda på den rådande teckentabellens kodning. Det blir relevant när C++ kod byter plattform.

Med hjälp av programmen [Char2int](#) (sid 120), [Int2char](#) (sid 121) och [Escape](#) (sid 125) som vi gått igenom på lektion, kan ni kolla tecknens ASCII-koder resp. escapesekvenser.

Anmärkingar till escapetabellen:

vagnretur (i tabellens tredje kolumn) betyder **Return** eller **Enter** *utan* radbyte, dvs förflyttning av skrivmarkören till början av *samma* rad.

nolltecknet (i tabellens tredje kolumn) har i C++ olika betydelser, t.ex. som strängavslutningstecknet i arrays av **char**. I andra sammanhang har det andra funktionaliteter ([sid 202](#)).

Lycka till!

Hälsningar

Taifun