

## Inledning:

### *Läs igenom hela instruktionen först!*

Tanken är att fräscha upp lite C och börja jobba med bitwise-operationer. För en snabb repetition av det binära talsystemet kan ni kika här:

<http://www.matteboken.se/lektioner/matte-1/tal/talsystem>

Men man kommer ganska långt bara med denna tabell:

128	64	32	16	8	4	2	1
1	0	1	1	0	1	0	1

Så det binära talet: 10110101 =

$$128 + 0 + 32 + 16 + 0 + 4 + 0 + 1 = 181$$

Vi kan kontrollera detta på räknaren i Windows:

The screenshot shows the Windows Calculator application in 'Kalkylatorn' (Calculator) mode, specifically the 'PROGRAMMERARE' (Programmer) tab. The main display shows the decimal value '181'. Below the display, the binary representation '1011 0101' is shown under the 'BIN' label. The calculator is set to 'QWORD' mode. The right-hand pane, titled 'Minne' (Memory), displays the message 'Det finns inget sparad i minnet' (Nothing is saved in memory). The calculator interface includes various buttons for bitwise operations (Lsh, Rsh, Or, Xor, Not, And), arithmetic operations (Mod, CE, C, <math>\times</math>, <math>-</math>, <math>+</math>, <math>=</math>), and standard arithmetic symbols (↑, ↓, A, B, C, D, E, F, (, ), ±, 0, ,).

Dessa Bit-operatorer låter oss bearbeta bitar (dvs 1:or och 0:or) istället för hela tal (Se det utdragna avsnittet ur boken C från början av Jan Skansholm som ligger på moodle)

## Bit-operatorer

Är bara definierade för heltalstyper.  
Utför alla operationerna bit för bit.

~	byter 0 <-> 1
&	och, bit för bit
<	eller, bit för bit
^	exclusive or
<<	vänsterskift
>>	högerskift

## ÖvningsUppgifter för bitoperationer med Facit:

1. [Write a C program to check Least Significant Bit \(LSB\) of a number is set or not.](#)
2. [Write a C program to check Most Significant Bit \(MSB\) of a number is set or not.](#)
3. [Write a C program to get nth bit of a number.](#)
4. [Write a C program to set nth bit of a number.](#)
5. [Write a C program to clear nth bit of a number.](#)
6. [Write a C program to toggle nth bit of a number.](#)
7. [Write a C program to get highest set bit of a number.](#)
8. [Write a C program to get lowest set bit of a number.](#)
9. [Write a C program to count trailing zeros in a binary number.](#)
10. [Write a C program to count leading zeros in a binary number.](#)
11. [Write a C program to flip bits of a binary number using bitwise operator.](#)
12. [Write a C program to total number of zeros and ones in a binary number.](#)
13. [Write a C program to convert decimal to binary number system using bitwise operator.](#)
14. [Write a C program to swap two numbers using bitwise operator.](#)
15. [Write a C program to check whether a number is even or odd using bitwise operator.](#)

## Den faktiskt uppgiften att lämna in (hämtad ur boken):

### Övning 15.5

Skriv ett program som läser in ett hexadecimalt heltal till en variabel av typen **unsigned int**. Undersök om det inlästa talet innehåller en etta i den tredje biten från höger. Ändra i så fall biten till en nolla och sätt den första biten från höger till en etta. Skriv sedan ut talet i hexadecimal form.