

Tallsystemer, en rask innføring for studenter ved Oslo By Steinerskole.

© Hans Poppe 2001

Det desimale tallsystem:

Det er sannsynlig at du kjenner til det desimale tallsystemet (10-tallsystemet) og derfor skal vi ta utgangspunkt i det. Det er ofte lettere å forstå et nytt konsept, hvis vi tar utgangspunkt i en kjent metode.

Det desimale tallsystemet er det vi bruker hver dag til å telle. Navnet har det fått fra det latinske ordet "decem" som betyr ti. Dette er fornuftig, fordi systemet bruker 10 siffer, fra 0 til 9. (Tell selv fra 0 til 9 og sjekk at det er 10 siffer!). Disse sifrene er det vi kaller *symbolene* i titallsystemet. Ettersom vi har 10 symboler, kan vi telle fra 0 til 9. Legg merke til at selv om 0 ofte representerer "ikke noe", er det et siffer / symbol som gjelder! Vi må jo, tross alt, ha en numerisk måte å skrive "ikke noe" på. Hvis vi ønsker å telle lenger enn til 9 må vi kombinere flere siffer / symboler. (heretter kaller vi symboler for sifre). Vi ser på et eksempel:

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Tabellen har ti tall horisontalt, det samme antall siffer som vi har i titallsystemet. Se nå på rad (rader går horisontalt, kolonner går vertikalt!) to, kolonne 1. Vi har lagt et ettall foran nullen, og får dermed ti i cellen. (En celle er den ruten som er der en rad krysser en kolonne!). I rad tre er 1 byttet ut med 2 foran 0, og vi får tyve i cellen. Nå skal vi forsøke å tenke oss at alle cellene i rad 1 har en null foran seg; det er ikke så vanskelig, 01 er jo samme verdi som 1, og 03 er samme verdi som 3. Husk det er ingen kommaer her. Vi skal nå forsøke å sette disse tallene opp i en litt større tabell:

		<u>Least significant digit</u>									
		<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>
<u>Most significant digit</u>	<u>0</u>	00	01	02	03	04	05	06	07	08	09
	<u>1</u>	10	11	12	13	14	15	16	17	18	19
	<u>2</u>	20	21	22	23	24	25	26	27	28	29
	<u>3</u>	30	31	32	33	34	35	36	37	38	39
	<u>4</u>	40	41	42	43	44	45	46	47	48	49
	<u>5</u>	50	51	52	53	54	55	56	57	58	59
	<u>6</u>	60	61	62	63	64	65	66	67	68	69
	<u>7</u>	70	71	72	73	74	75	76	77	78	79
	<u>8</u>	80	81	82	83	84	85	86	87	88	89
	<u>9</u>	90	91	92	93	94	95	96	97	98	99

Dette introduserer et nytt, viktig konsept. dette er forskjellen på MSD og LSD, altså Most Significant digit og Least Significant Digit. LSD er den som står lengst til høyre (og dermed representerer lavest verdi) mens MSD er lengst til venstre og representerer den høyeste verdien. Etter hvert som vi slipper opp for LSD (jada, det er sikkert noen som får andre assosiasjoner her ☺.....), så øker vi verdien av sifferet til venstre for det med 1. Dette kaller vi, med et fremmedord, å inkrementere (eng. to increment) tallet med en. Der det ikke er noe tall til venstre, setter vi inn en imaginær null, og inkrementerer den. For eksempel, etter 98 inkrementerer vi LSD med 1 og får 99 som er det samme som 099. Ved å legge til 1 vil LSD bli 0 og vi må inkrementere 099 til 100. Nå er MSD = 1 !! (Tallet er 100).

Som en generell regel kan vi si at viktigheten (significance) blir større jo lenger over til venstre vi kommer. Derfor er det slik at når vi teller opp antall siffer i et tall, teller vi ALLTID fra høyre mot venstre. For eksempel er det slik at i tallet 167453, sier vi at LSD (altså 3 tallet) er siffer nr. 0, og 4-tallet er siffer nr. 2. (Ikke glem: 0 er et tall!!)

Vi skal nå se raskt på de matematiske operasjonene vi kan utføre med decimale tall. (Vi skal gjøre dette veldig enkelt og ikke bruke negative tall og heller ikke verken brøk eller desimaler. Kun hele tall fra 0 og oppover. (I matematikken kalles slike tall for naturlige tall. Datafolk kaller dem "unsigned integers")

Addisjon: Prosessen med å øke en verdi med en annen verdi (addere). Dette gjøres ganske enkelt ved å øke verdien av et tall med verdien av et annet tall. $2+3=5$ fordi du teller tre verdier opp fra 2 og havner på 5. Likeledes: $9+6=15$ fordi du teller opp 6 verdier fra 9 og vil havne på 15.

Subtraksjon: Prosessen med å trekke et tall fra et annet. Dette gjøres ved å minke (telle nedover), akkurat som vi telte oppover når vi adderer. Eksempel: $15-6=9$ fordi du havner på 9 hvis du teller 6 trinn nedover fra 15. Subtraksjon er altså det motsatte av addisjon. Den eneste forskjellen er hvilken retning vi teller.

Multiplikasjon: Prosessen med å addere et tall til seg selv en rekke ganger. $3*3=9$ fordi $3+3+3=9$ og $8*6=48$ fordi $8+8+8+8+8+8=48$. Enkelt, ikke sant? $5*0=0$ fordi du adderer 5 til seg selv INGEN ganger! $5*1=5$ fordi du tar 5 én gang, og fem er ganske enkelt fem!

Divisjon: Den inverse (altså den motsatte) prosessen av multiplikasjon. I prinsippet gjøres dette ved å subtrahere (trekke fra) det samme tallet en rekke ganger fra det opprinnelige tallet. Eksempel: $6/2=3$ fordi $6-2-2-2=0$. Du trekker altså fra 2 tre ganger fra 6 for at svaret skal bli 0. Likeledes: $48/8=6$ fordi $48-8-8-8-8-8-8=0$. Se på dette, det er litt vanskelig å se systemet med en gang, men forsøk med flere tall til du plutselig "ser lyset..."

I midlertid er det dessverre slik at dette systemet ikke bestandig virker. Se på eksempelet: $20/8=$. $20-8-8=4$, og dette går altså ikke "opp". Vi skal bare konsentrere oss om hele, naturlige tall, så i dette tilfellet sier vi at $20/8=2$ "med 4 til rest". Denne "resten" kalles på matematikk og dataspråk for en "MODULO". Dermed forstår vi (jada, vi gjør det!) at $20/1=20$ fordi du bare trenger 1 stykk "20" for å få 20, og moduloen vil alltid bli 0.

OBS!! Det går IKKE an å dividere på 0. Du kan IKKE dele et tall på 0. Det går ikke an! Svaret blir ALDRI 0, og heller ALDRI et uendelig tall, eller noe annet tall. Det går ikke an! Det er en vanlig feiloppfatning at $5/0=0$, men dette er totalt galt. Dette regnestykket er en umulighet. Legg også merke til at det går den andre veien: $0/7=0$, og her er svaret 0, og regnestykket er riktig, men å dele på 0 går ikke an. Du får ikke noe svar, uansett hvordan du vrir og vender på det. Altså: **Å DIVIDERE MED 0 GÅR IKKE AN, DET HAR IKKE NOE SVAR.**

Eksponenter: Prosessen med å multiplisere et tall med seg selv, en eller flere ganger. $2^3=8$ (nei! ikke 9!), fordi $2*2*2=8$, fordi $2*(2+2)=8$, fordi $(2+2)+2+2=8$. På denne måten blir $3^4=81$ fordi $3*3*3*3=81$, fordi $3*3*(3+3+3)=81$ fordi $3*((3+3+3)+(3+3+3)+(3+3+3))=81$ fordi $(3+3+3)+(3+3+3)+(3+3+3)+((3+3+3)+(3+3+3)+(3+3+3))+((3+3+3)+(3+3+3)+(3+3+3))=81$. Det er et unntak her også, det er dersom eksponenten er 0. For eksempel 3^0 . Dette kan heller ikke egentlig regnes ut, det er en definert verdi og den er definert til 1. Altså $3^0=1$ og $67^0=1$. Dette er en meget viktig regel, hvorfor vil du forstå om en liten stund. Akkurat som multiplikasjon er en slags hurtigmetode (stenografisk metode) for å addere et nummer med seg selv, er eksponentregning en hurtigmetode for å multiplisere et tall med seg selv. Vi skal se på de symbolene / tegnene vi bruker for å skrive de forskjellige matematiske operatorene:

Operator	Symbol
addisjon	+
subtraksjon	-
multiplikasjon	*
divisjon	/
modulo (rest)	%
eksponent	^

Dette er kanskje ikke den måten du er vant til å skrive disse tingene på, men vi skal likevel bruke dem, fordi de er standardmåten å skrive dette på når vi bruker en PC (i hvertfall en windowmaskin☺) og brukes i programmeringsspråk (C++ og Java, for å nevne to) så vel som i regneark og internasjonal matematisk litteratur. Dersom vi kombinerer disse operatorene med tall får det som kalles et uttrykk (eller en "expression" på engelsk og "data"). Uttrykk beregnes på en forhåndsdefinert måte. Det er ikke tilfeldig hva som regnes ut først i et regnestykke med flere ledd. rekkefølgen er denne:

1. eksponenter
2. multiplikasjoner
3. divisjoner og moduloer (rester)
4. addisjoner og subtraksjoner

Se på punkt 4, hva kommer først? Det er rekkefølgen fra venstre mot høyre i uttrykket som bestemmer. Altså $2+7-1=8$. $4+2*3=10$ fordi $2*3=6+4=10$ (multiplikasjon før addisjon). $4+3*2^2=16$ fordi $2^2=4$ og $3*4=12$ og $12+4=16$. Derfor blir også $2^2*3+4=16$. Sjekk selv! Hvis du nå ønsker å først addere $4+3$, deretter multiplisere med 2 og så beregne potensen må du eksplisitt bruke parenteser. I eksempelet over ville du måtte skrive: $((4+3)*2)^2=196$. Dette fordi det blir regnet som følger: $4+3=7$ og $7*2=14$ og $14^2=196$. dette er jo et helt annet svar..... Vi skal droppe desimalmatematikken her....

En matematisk titt på desimale numre:

En matematisk måte å se på tallet 7 kunne være $7*1$, fordi 7 er syv enere. Ser vi på tallet 36 kan vi gjøre det samme, det er to siffer, det vil si 3 tiere og 6 enere. 3 er MSD og 6 er LSD. MSD er i posisjon 1 og LSD er i posisjon 0. (NB! dette er viktig).

2487 blir gjort om på samme måte; vi ser på verdiene 1000, 100, 10 og 1. Det er $2*1000$, $4*100$, $8*10$ og $7*1$.

Hvis du ser på verdiene 1000, 100, 10 og 1, som vi multipliserer med 2,4,8 og 7 henholdsvis, og samtidig ser på posisjonen det enkelte tall har (3, 2, 1, 0) og kombinerer dette med at du er sikker på at det er desimale tall vi regner med så kan vi skrive det slik: $2*10^3 + 4*10^2 + 8*10^1 + 7*10^0$.

Likeledes:: $36 = 3*10^1 + 6*10^0$ og $7 = 7*10^0$. Dette må du klare å forstå, det er ikke noe poeng i å gå videre før dette sitter. Ellers kan jeg garantere at du vil falle fra når dette blir hexadesimale eller binære tall.

Det hexadesimale system:

I prinsippet er det hexadesimale tallsystemet akkurat som det desimale, men med flere siffer å velge mellom. Slik som det var 10 siffer i det desimale systemet (0-9) er det 16 siffer i det hexadesimale system. Sifrene er 0 – F. Altså: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

Hexadecimal kommer fra latin hvor decem betyr 10 og hexa betyr 6 ($6+10=16$).

Vi ser på tabellen fra tidligere, men utvidet til hexadesimalt system:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F

Fra og med nå slutter vi å kalle dette hexadesimale tall, vi kaller dem hex. Dette fordi det er svært vanlig og dessuten blir det litt mindre for meg å skrive.

Bortsett fra åpenbare forskjeller i antall og utseende på de 16 symbolene er det slik at "MATH RULES!!" Matematikken er akkurat den samme: $3+3=6$, $7+2=9$ og $9+4=D$. Når du inkrementerer er det alltid til venstre for den posisjonen du er i. Du legger altså til en MSD til venstre her også, dersom du har sluppet opp for symboler. For eksempel: $C+5=11$, og Hex $11+3C=4D$. Dette er jo enkelt og logisk, ikke sant?? Vel, det er nok slik at mange syns dette er vanskelig, ettersom vi er trent så nøye i desimaltall, men ikke i hex tall. Det er her det er greit med kalkulator....

Følgende utsagn er riktig i hex-matematikk. Forsøk å se om du greier å "se" dette:

$$D1C5 = D*10^3 + 1*10^2 + C*10^1 + 5*10^0$$

Regn om svaret til desimaltall.

(Fikk du treogfemtitusensyvhundredeogen?)

Den ”store tabellen” for hextall vil naturligvis bli mye større enn desimaltabellen. Desimaltabellen hadde 100 ruter ($10 \cdot 10 = 100$) mens hextabellen blir $16 \cdot 16 = 256$. Her er den:

		Least significant digit															
Most significant Digit		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	1	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
	2	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
	3	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
	4	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
	5	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
	6	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
	7	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
	8	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
	9	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
	A	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	B	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
	C	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
	D	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
	E	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
	F	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Det er en spesiell måte å skrive hextall på. Det er jo egentlig ikke mulig å se om 63 er sekstittre eller om det er et hexadesimalt 63 som tilsvarer 99 desimalt (prøv med kalkulatoren). Måten å skrive dette på er tofoldig: enten med en h etter tallet (63h) eller med 0x foran tallet (0x63), som er standardmetoden i C++ programmering. Disse metodene både kan brukes og blir brukt om hverandre.

Den desimale matematikken virker fremdeles. Det er for så vidt ingen forskjeller her. For eksempel: $10 + 10 = 20$, og $10h + 10h = 20h$. Legg i midlertid merke til at selv om $50 + 50 = 100$ så er ikke $50h + 50h = 100h$, neida det blir A0h. Sjekk selv på kalkulatoren. OBS, ikke la deg lure, dersom det står 50h er det et hextall, står det ikke noe er det desimalt!!! Husk også at kalkulatoren ikke skriver verken h eller 0x. Du må selv se etter hvilket modus den står i.

Det binære tallsystemet.

Akkurat som i det hexadesimale tallsystemet er det slik at: "MATH RULES!!" Matematikken virker og den er stort sett den samme. (Det finnes en spesiell type matematikk som brukes på binære tall. Dette kalles logisk AND, logisk OR, logisk XOR + noen flere spesielle saker, men dette kommer vi tilbake til neste VK2.)

Bi er det latinske ordet for to, og dermed forstår vi alle alt sammen... Det er to siffer / symboler å velge mellom. 0 og 1 heter de, men det gjetter du vel for en god stund siden... Som du sikkert også forstår (husk: matematikken er den samme) så er talltabellen mye mindre her enn i desimale eller hexadesimale tall.

0	1
10	11
100	101

Vi tar et par eksempler på regnestykker:

$$1b+1b=10b$$

$$110b+101b=1011b$$

Den matematiske representasjonen blir også det samme i binært som det var i desimale og hexadesimale tall: $10 = 1 \cdot 10^1 + 0 \cdot 10^0$. Ser du mønsteret?

Når vi skriver binære tall får vi samme problemstilling som med de hexadesimale tallene. Du kan ikke se forskjell på 100 og 100 og 100. De binære tallene får bokstaven b etter seg, slik som de hexadesimale får en h. Det er ingen tilsvarende 0x variant (for programmering) rett og slett fordi det høyst uvanlig å kode i binære (maskinsprog) eller desimale (menneskesprog) tall.

Desimale, hexadesimale og binære tall, sett samlet:

Hva skal vi med dette? En PC er en maskin som KUN regner med binære tall, dypest nede, innerst inne. Den bruker det binære systemet til å utføre alle regneoperasjoner. Når PCen din viser deg desimale tall på skjermen er dette bare en oversettelse som foretas for at ditt liv skal være litt lettere å leve. Binære tall (av en viss størrelse) er laaaaange og meget vanskelige å forholde seg til. Vi foretrekker derfor at PCen oversetter for oss. Ettersom datamaskinen bruker bits i grupper på 8 og 8, som kalles byte (fra engelsk by eight), er det lurt å finne en metode for å uttrykke bytes på en måte som er noe lettere for mennesker, samtidig som det er lett for datamaskinen å arbeide med det.. Det er her de hexadesimale tallene kommer inn i billedet. Et hextall kan alltid omgjøres til et binært tall med fire bits. 2^4 bits er en byte og derfor kan, naturligvis, en byte alltid skrives som 2 hextall. (4 bits kalles "a nibble" på engelsk. Sannsynligvis fordi "byte" høres ut som "bite" som betyr "å bite" på norsk. "To nibble" på engelsk betyr å spise / bite i små biter. (See the light, Lothar?))

Vi setter raskt opp en tabell for å konvertere mellom binær, desimal og hextall:

Bin	Hex	Dec
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	6
111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Ser du det? Alle hextall kan uttrykkes med 4 bits, eller en halv byte (også kjent som nibble). Dette betyr altså at det høyeste tallet en PC kan behandle på en eneste enkelt regneoperasjon er 1111111b. (åtte entall i binær form).Det er ikke så vanskelig å konvertere disse tallene til desimale, vi vet jo at i binærform blir det:

$11111111 = 1 \cdot 10^7 + 1 \cdot 10^6 + 1 \cdot 10^5 + 1 \cdot 10^4 + 1 \cdot 10^3 + 1 \cdot 10^2 + 1 \cdot 10^1 + 1 \cdot 10^0$.
Nå konverterer vi etter tabellen over til desimaltall, men ta bare den delen som er til ”høyre” i regneuttrykket: (se nøye på dette en gang til før du spør...)

$$1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

Altså: det høyeste tallet en PC kan regne med i en enkel regneoperasjon (1111111b) er 255 i desimaltall. Det gir oss 256 verdier, husk 0 er en verdi den også.

Det samme gjelder for hextall også:

$$1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 80 + 40 + 20 + 10 + 8 + 4 + 2 + 1 = FF$$

Du ser at $1111111b = 255 = FFh$.

Med den kule, lille tabellen over ser du raskt at det ville ha vært lurt å dele byten (1111111b) opp i to nibbles. (Du ser det instinktivt, ikke sant?? ☺) Da ville vi fått at $1111111b = 1111b \& 1111b$, ikke sant? (Her bruker jeg ampersand istedenfor pluss, rett og slett fordi det ikke er noe matematisk her, vi bare deler det i to med en strek, det er ingen divisjon som utføres).

Dermed er jo veien kort til å forstå at $1111b \& 1111b = Fh \& Fh$ eller: $1111111b = FFh$.

Dette virker også den andre veien: $E7h = 1110 \ 0111b$ fordi $Eh = 1110b$ og $7h = 0111b$.

For å konvertere til og fra desimale tall må du fremdeles bruke kalkulatoren, eller en større konverteringstabell. Du kan jo lage en?

Du bør nå være i stand til å se hvilken type tall du arbeider med i alle tre tallsystemene. Du bør også være i stand til å konvertere mellom de tre systemene enten ved hjelp av kalkulator eller med hoderegning. Alternativt lager du deg din egen tabell.