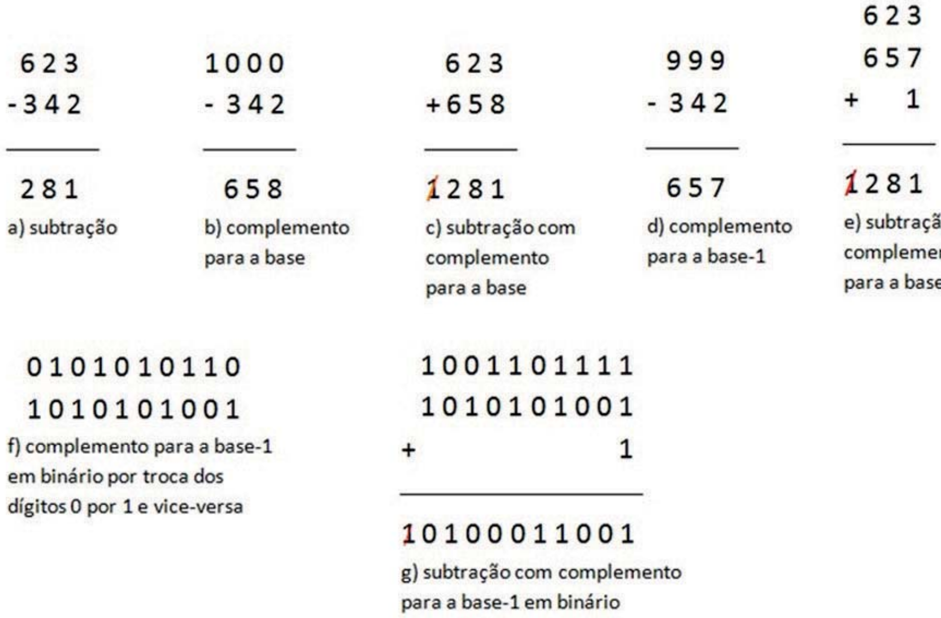
Subtraindo com somas!

Por: Maria do Carmo Martins
Professora do Departamento de Matemática
da Universidade dos Açores
mika@uac.pt



O título é um tanto ou quanto paradoxal. Como é que é possível subtrair usando apenas somas? Na realidade é um método simples de executar e (talvez) de explicar. Das quatro operações básicas da Aritmética aprendidas na escola primária: adição, subtração, multiplicação e divisão, simpatizamos mais com a adição e a multiplicação. As outras duas são um verdadeiro tormento. Hoje, vou falar-vos de um método que permite contornar subtrações... Se o leitor quiser seguir todos os detalhes, aconselho-o a ter à mão papel e lápis e a deixar-se levar nesta aventura numérica. Suponhamos, por exemplo, que se pretende calcular 623 - 342. Ora, recorrendo ao algoritmo que aprendemos em tenra idade, executamos com destreza: 2 para 3 é 1; 4 para 12 são 8 e vai 1; 3 mais 1 é igual a 4, que para 6 são 2. Ou seja, 623 - 342 = 281, ver figura a). O passo que nos leva a tirar 4 de 12 é seguramente o mais complicado e o que maior angústia nos causou quando percebíamos que tínhamos falhado a conta.

Como efetuar então subtrações recorrendo somente a somas? O algoritmo é simples: tome-se o número 342 (o subtrativo da operação acima) e calculemos **o seu complemento para a base**, ou seja, vamos determinar o número que adicionado a 342 é igual a 1000. Obtemos o número 658, ver figura b). Agora procedamos à adição 623 + 658, ou seja o aditivo adicionando ao complemento para a base que calculámos. O resultado é 1281, ver figura c). Se ignorarmos o algarismo da esquerda (o algarismo das unidades de milhar), obtemos 281 que é a diferença correta. Surpreendidos? Será um truque? Funciona sempre? E porquê?

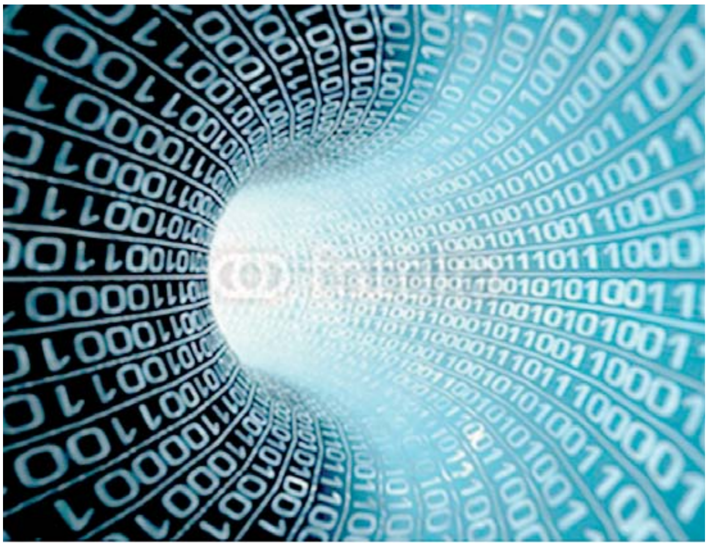
Vamos explicar "o porquê" e assim tentar aclarar algumas das interrogações do leitor. Iremos focar a nossa atenção no cálculo do complemento para a base. Como vimos, o complemento para a base de 342 é 658, pois 342 + 658 = 1000. Generalizando, se tivermos um numeral representativo do número *x* com *n* dígitos, o complemento para a base de *x* resulta de efetuarmos 10^*n* - *x*. No caso de 342, temos 10^3 - 342 = 1000 - 342 = 658. Outro exemplo: qual o complemento para a base de 3267? Como este numeral tem 4 dígitos fazemos 10^4 - 3267 = 10000 - 3267 = 6733. Se quisermos calcular 8231 - 3267, fazemos 8231 + 6733 = 14964 e desprezando o dígito da esquerda, obtemos 4964 que o leitor pode confirmar facilmente ser o resultado de 8231 - 3267. Resultou pela segunda vez! Porquê? Falta-nos esmiuçar mais um ingrediente: o que significa "desprezar o dígito da esquerda"? Quando desprezámos o dígito da esquerda em 1281 obtivemos 281. Mas isto significa 1281 - 1000 = 1281 - 10^3 = 281 (em que 3 é o número de dígitos de 342). Desprezar o dígito da esquerda de 14964 corresponde a efetuar 14964 - 10000 = 14964 - 10^4 = 4964 (sendo 4 o número de dígitos de 3267).

Podemos agora explicar porque é que o algoritmo funciona: se tivermos dois numerais representativos dos números *x* e y com *n* algarismos, o complemento para a base de *y* é 10^*n* - *y*. Calcular *x* - *y* é então calcular *x* + *complemento para a base de y* e *desprezar o dígito da esquerda*, ou seja, *x* + (10^*n* - *y*) - 10^*n* = *x* + (10^*n* - 10^*n*) - *y* = *x* + 0 - *y* = *x* - *y*. Eis a razão pela qual o algoritmo funciona sempre!

O leitor irá argumentar que se trata de um embuste. Trocamos uma subtração por outra, a do cálculo do complemento, e nem nos livramos do "e vai 1". Então, qual o interesse deste método? Vamos por partes. Primeiro vamos livrar-nos do "e vai 1". Em vez de calcularmos o complemento para a base, vamos calcular **o complemente para a base-1**. Tome-se novamente o primeiro exemplo: 623 - 342. Agora calculemos o *complemento para a base-1* de 342. Para tal, efetuamos (1000 - 1) - 342 = 999 - 342 = 657, ver figura d). Agora o cálculo é muito mais simples e livramo-nos do "e vai 1", pois basta substituir cada dígito de 342 pelas unidades que lhe falta para chegar a 9. Para que o resultado da subtração seja correto há que compensar adicionando mais 1 no fim. Vejamos: 623 - 342 = 623 + 657 + 1 - 1000 = 281 (em que o -1000 significa "desprezar o dígito mais à esquerda"). Embora a subtração seja mais simples, há que fazer ainda uma subtração! Como nos livramos dela? E para que serve afinal este método que curiosamente era um dos que os nossos pais e avós utilizavam antigamente?

Para respondermos às duas questões, temos que falar em cálculo mecanizado, isto é, vamos analisar, ainda que de forma abstrata, como um computador faz esta conta. Em primeiro lugar, os computadores não representam os números em base decimal como nós o fazemos. Na base decimal utilizamos dez símbolos distintos (de 0 a 9) para representar os números. Contudo, nem sempre se utilizou a base decimal. Por exemplo, a civilização Suméria usava a base sexagesimal (60 símbolos distintos para representar os números). Este sistema foi depois passado para os Babilónios e, em algumas coisas, chegou aos nossos dias. Por exemplo, ainda dividimos uma hora em sessenta minutos e um minuto em sessenta segundos.

Os computadores usam a base binária para representar os números: recorrem somente aos símbolos 0 e 1, o que simplifica bastante a forma como construir os circuitos eletrónicos. Listo os números binários entre 0 e 15: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111. Note o leitor que em base binária a seguir a 1 vem 10, tal como acontece em base decimal quando usamos todos os símbolos e a seguir a 9 vem 10. Tal como em base decimal 342 significa 3 x 100 + 4 x 10 + 2 = 3 x 10^2 + 4 x 10^1 + 2 x 10^0, em base binária, por exemplo, 1100 = 1 x 2^3 + 1 x 2^2 + 0 x 2^1 + 0 x 2^0 = 1 x 8 + 1 x 4 + 0 x 2 + 0 x 1 = 12 (em base decimal). A realçar que na base binária só há a tabuada do 0 e do 1 (pelo que não temos que saber a malfadada tabuada do 7 ou do 8!). As coisas tornam-se bastante simples: 0 x 0 = 0, 0 x 1 = 0, 1 x 0 = 0 e 1 x 1 = 1. A adição é igualmente simples: 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1 e 1 + 1 = 0 (**e vai 1**). A escolha da base binária para ser usada nos computadores é clara. Então porque não a usamos no dia-a-dia? A base binária tem a desvantagem de representar os números com mais algarismos do que a base decimal. Por exemplo, 623 em binário é 1001101111 e 342 é 0101010110.



Tome-se o número 0110 e calculemos o seu complemento para a base-1. Neste caso, teremos de calcular **1111** - 0110. Note-se que na base binária o dígito de maior valor é o 1 e por isso usamos 1111. Calculemos então 1111 - 0110: 0 para 1, 1; 1 para 1, 0; 1 para 1, 0; 0 para 1, 1; ou seja, 1111 - 0110 = 1001. Simples, não? Mas, pode ser ainda mais simples. Calcular o complemento para a base-1 em binário resume-se a trocar os dígitos 0 por 1 e 1 por 0. Portanto, o complemento para a base-1 de 1001 é 0110, o de 1100 é 0011 e a "conta" é feita sem recorrer a qualquer subtração. Retomemos a nossa conta original, 623 - 342, mas agora em binário. Ora 342 é 0101010110, cujo seu complemento para a base-1 é 1010101001, ver figura f). Assim, 1001101111 + 1010101001 + 1 = 10100011001, ver figura g), e desprezando o dígito da esquerda obtemos 0100011001, que corresponde a 281 em base decimal, como não poderia deixar de ser.

Para terminar, vamos resolver "o problema" de desprezar o dígito da esquerda (a última subtração que nos resta). Os computadores trabalham com um número fixo de dígitos (*bits*). Quando o leitor ouve dizer que o seu computador é de 32 *bits* ou de 64 *bits*, significa que o tamanho dos números que este representa tem 32 *bits* ou 64 *bits*, respetivamente. Ao pôr em prática o algoritmo que indicámos, surge sempre um dígito extra (como já explicámos antes). Mas, este dígito extra seria o 33.º ou o 65.º que não se consegue representar numa máquina com 32 ou 64 *bits*, por isso dá imenso jeito que seja desprezado, não sendo necessário fazer qualquer subtração. E assim, todo este método faz sentido, digo eu! Bom Natal!