

Olimpíadas Nacionais de Informática
14 de Abril de 2000
Lisboa, Escola Secundária Professor Herculano de Carvalho

Primeiro Problema: Números perfeitos, defeituosos e excessivos.

Um número *perfeito* é um número igual à soma dos seus factores próprios (Um *factor próprio* de um número x é outro número y tal que x é divisível por y , e x é diferente de y). Por exemplo, 6 é um número perfeito, pois $6 = 1 + 2 + 3$, e 1, 2, e 3 são os únicos factores próprios de 6 (o outro factor é 6, mas não é um factor próprio). Outro número perfeito é 28, pois $28 = 1 + 2 + 4 + 7 + 14$. Um número *defeituoso* é um número cuja soma dos factores próprios é menor que ele, e um número *excessivo* é um número cuja soma dos factores próprios é maior que ele. (Todos os número de que falamos são inteiros positivos, claro).

Por definição, o *defeito* de um número é a diferença entre ele e a soma dos seus factores próprios. Logo, um número perfeito tem defeito zero, um número defeituoso tem defeito positivo e um número excessivo tem defeito negativo.

Dizemos que um número é menos defeituoso que outro se o seu defeito for menor que o do outro, ou em caso de empate (isto é, se os defeitos forem iguais) se for menor que o outro. Esta definição aplica-se a qualquer par de números, claro, perfeitos, defeituosos ou excessivos. Por exemplo, um número excessivo é sempre menos defeituoso que um número defeituoso.

Tarefa:

A sua tarefa neste problema é escrever um programa para calcular e apresentar a listas dos números perfeitos, a lista dos números defeituosos e a lista dos números excessivos existentes num dado intervalo. As listas virão apresentadas lado a lado, num ficheiro texto, ordenadas, a dos perfeitos por ordem numérica crescente, a

dos defeituosos por ordem de “defeituosidade” crescente, e a dos excessivos (que têm defeito negativo) por ordem de “defeituosidade” decrescente.

Dados:

Os dados são os limites do intervalo e vêm no ficheiro P1.INP, o qual tem uma única linha com dois números. O primeiro número é o limite esquerdo do intervalo e o segundo o limite direito. Ambos os números são inteiros positivos inferiores a 4096. O limite esquerdo é menor ou igual ao limite direito.

Resultado:

O resultado vem no ficheiro P1.OUT, o qual contém três colunas: a coluna da esquerda contém os números perfeitos do intervalo dado, a coluna do meio contém os números defeituosos, e a coluna da direita os excessivos. As colunas estão ordenadas conforme especificado anteriormente. Cada coluna tem largura correspondente a cinco posições, e os números estão encostados à direita nessas cinco posições. Note que nem todas as colunas terão o mesmo número de números.

Exemplo:

```
P1.INP
20 32
```

```
P1.OUT
      28   32   24
      22   30
      21   20
      26
      27
      25
      23
      29
      31
```

Note bem:

Se não tiver tempo para resolver a parte da ordenação, escreva as colunas por ordem crescente. Se não tiver tempo

para escrever as três colunas como indicado escreva primeiro os números perfeitos, depois os defeituosos, depois os excessivos, um por linha. Se não tiver tempo para os números defeituosos e para os excessivos, resolva a parte dos números perfeitos.

Segundo problema: Acertando no “*site*”

Com tantos *sites* interessantes na Internet, alguns com nomes estranhos, é impossível lembrarmos do nome de todos eles exactamente, mesmo daqueles que visitámos há pouco. O que era bom era que os nossos *browsers* se lembrassem disso e nos ajudassem a reencontrar os *sites* de cujos nomes já não nos lembramos bem. Podia ser assim: o *browser* guardaria a lista dos nomes dos últimos 500 *sites* por onde passámos. Depois quando escrevêssemos o nome de um *site*, se o *browser* não o encontrasse, substituí-lo-ia por um *site* da lista com um nome “parecido”. O problema, portanto, é escolher na lista os *sites* cujo nome é parecido com o nome introduzido.

O critério de pareçença é simples: dada uma palavra *x* dizemos que uma outra palavra *y* é *parecida* (por definição) com *x* se todos os caracteres de *x* existirem em *y*, pela mesma ordem, mas não necessariamente em posições consecutivas. Por exemplo, a palavra “telecom” é parecida com “tlc”, com “eco”, com “elem”, com “teo”, etc.; a palavra “olimpiadas” é parecida com “olimp”, com “piadas”, com “lmpd”, etc. A palavra “futebol” é parecida com “ftbl”, “bol”, “teo”, etc.

Quando o *browser* encontrar vários *sites* com nomes parecidos com o nome indicado pelo utilizador (como “telecom” e “futebol”, ambos parecidos com “teo”), deveria mostrá-los todos, para o utilizador poder seleccionar o pretendido. Se o número de palavras parecidas encontradas for maior que um certo limite *N*, apenas devem ser apresentados os *N* correspondentes aos *N sites* visitados mais recentemente (de entre aqueles que têm nomes parecidos com o nome indicado).

Tarefa:

A sua tarefa neste problema é escrever um programa para encontrar numa lista de palavras representando *sites* na Internet as

palavras parecidas com certas palavras dadas.

Dados:

A lista de palavras com nomes de *sites* reside num ficheiro texto, de nome SITES.TXT, com um nome por linha, encostado à esquerda, sem espaços. O ficheiro tem no máximo 500 linhas, cada uma tem no máximo 31 caracteres. Os *sites* estão listados por ordem cronológica de visita, sem repetições, com os mais antigos no início e os mais recentes no fim.

Os dados para este estão no ficheiro de texto P2.INP. Na primeira linha vem o número *N*, que indica o número máximo de *sites* a apresentar. Na segunda linha vem outro número, *P*. Nas *P* linhas seguintes, vêm *P* palavras (uma em cada linha, encostadas à esquerda, sem espaços.). O ficheiro tem, portanto, *P*+2 linhas.

Em ambos os ficheiros as palavras são formadas apenas por letras maiúsculas (sem acentos ou cedilhas), algarismos e pontos, hífenes e sublinhados.

Resultado:

O resultado do processamento vem no ficheiro P2.OUT. O ficheiro será constituído por vários troços, um para cada palavra no ficheiro P2.INP. O primeiro troço contém as palavras presentes em SITES.TXT parecidas com a primeira palavra de P2.INP, o segundo troço as palavras parecidas com a segunda, etc. Cada troço termina com uma linha onde existe apenas um asterisco. Cada troço lista no máximo *N* palavras (que correspondem aos *sites* mais recentes, isto é, àqueles que estavam mais próximos do final do ficheiro).

Exemplo:

SITES.TXT

WWW.TELECOM.PT
WWW.FUTEBOL.PT
WWW.UNIVERSIDADENOVA.PT
WWW.OLIMPIADAS-NACIONAIS.PT
WWW.RADIOCOMERCIAL.PT
WWW.CRAZYWORLD.COM
WWW.SUMMERVACATIONS.COM
WWW.AMAZON.CO.UK
WWW.ALIENLIFE.COM
WWW.REVOLUCAOPOPULAR.PT

P2.INP

4
6
AAA
TEO
CO.
XYZ
UNIVERS
UK

P2.OUT

WWW.OLIMPIADAS-NACIONAIS.PT
*
WWW.FUTEBOL.PT
WWW.TELECOM.PT
*
WWW.REVOLUCAOPOPULAR.PT
WWW.AMAZON.CO.UK
WWW.SUMMERVACATIONS.COM
WWW.CRAZYWORLD.COM
*
*
WWW.UNIVERSIDADENOVA.PT
*
WWW.AMAZON.CO.UK
*