

 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA <i>Departamento de Engenharia Informática</i></p>	<p>Trabalho nº 1</p> <p>Algoritmos e Estruturas de Dados</p> <p>2015/2016 – 2º Semestre</p> <p>“Números Especiais”</p>
---	--

Objectivos:

- Introdução ao sistema *Mooshak*.
- Análise de complexidade.

Exercícios

- Números Especiais (obrigatório)

Prazo limite de submissão no *Mooshak*: 14/02/2016, 23h00

Os alunos deverão consultar previamente o texto de apoio sobre o *Mooshak* que acompanha este enunciado, e proceder ao seu registo no servidor Mooshak de AED: <http://mooshak.dei.uc.pt/~aed>

Enunciado dos Problemas A, B, C, D

Os problemas A, B, C e D correspondem todos ao mesmo enunciado: dada uma sequência de números inteiros positivos e um valor inteiro positivo (VALOR), indicar se existe na sequência de números alguma subsequência cuja soma seja exatamente igual a VALOR.

Input

O *input* consistirá em sequências de valores inteiros positivos (cada um deles com um valor entre 1 e 1000).

A **primeira linha de cada subsequência** terá o tamanho da subsequência e o VALOR a pesquisar. As linhas seguintes apresentam os elementos da subsequência (um por linha).

Uma linha com o par “0 0” indica o final do *input*.

Output

Para cada sequência do *input* o programa deverá indicar, consoante exista ou não uma subsequência cuja soma seja igual a VALOR:

SUBSEQUENCIA NAO ENCONTRADA

SUBSEQUENCIA NA POSICAO i (sendo i a posição a partir da qual está a subsequência)

Exemplo de Input

```
3 8
10
5
1
7 5
1
2
3
4
5
6
7
0 0
```

Exemplo de Output

```
SUBSEQUENCIA NAO ENCONTRADA
SUBSEQUENCIA NA POSICAO 2
```

Problema A: Abordagem Exaustiva

Na primeira resolução do problema, deverá ser usado um algoritmo baseado na pesquisa directa e exaustiva de todas as sequências possíveis:

```
PARA i=1 ATÉ N {  
  PARA j=i ATÉ N  
    {  
      soma = 0;  
      PARA z = i ATÉ j  
        {  
          soma = soma + sequencia[z]  
        }  
      SE soma == VALOR ENTÃO  
        {  
          Subsequência existe na posição i  
          TERMINA  
        }  
    }  
  }  
} Subsequência não existe
```

Problema B: Reaproveitamento da Soma

Nesta resolução do problema, o algoritmo anterior deverá ser otimizado tendo em conta que o cálculo da soma ($\text{sequencia}[i] + \dots + \text{sequencia}[j-1] + \text{sequencia}[j]$) pode ser obtido adicionando simplesmente o valor “sequencia[j]” à soma anteriormente calculada ($\text{sequencia}[i] + \text{sequencia}[i+1] + \dots + \text{sequencia}[j-1]$).

```
PARA i=1 ATÉ N {  
  soma = 0;  
  PARA j=i ATÉ N  
    {  
      soma = soma + sequencia[j]  
      SE soma == VALOR ENTÃO  
        {  
          Subsequência existe na posição i  
          TERMINA  
        }  
    }  
  }  
} Subsequência não existe
```

Problema C: Interrupção da Soma

Na terceira resolução do problema, o algoritmo usado pode ser ainda mais otimizado, interrompendo o processo sempre que a soma obtida é já superior a VALOR.

```
PARA i=1 ATÉ N {
    soma = 0;
    j=i;
    ENQUANTO (soma < VALOR) e (j <= N)
    {
        soma = soma + sequencia[j]
        SE soma == VALOR ENTÃO
        {
            Subsequência existe na posição i
            TERMINA
        }
        j = j+1;
    }
    Subsequência não existe
```

Problema D: Otimização Dupla

Na última abordagem tentaremos otimizar o algoritmo conjugando o reaproveitamento dos algoritmos dos problemas C e D (ou seja, adicionar à última soma calculada o valor do elemento seguinte da sequência, em vez de recalcular toda a soma) com a “subtração” a essa soma do primeiro elemento da subsequência (ou seja, tomando partido do facto de sabermos que a soma dos elementos da sequência entre i+1 e j pode ser obtida subtraindo à soma dos elementos da sequência entre i e j (já estimada) o valor da sequência na posição i.

```
soma=0;
i=1;
j=1;
ENQUANTO (soma!=k) E (i<=n)
{
    ENQUANTO (soma<VALOR) E (j<=N)
    {
        soma = soma + sequencia[j];
        SE soma == VALOR ENTÃO
        {
            Subsequência existe na posição i
            TERMINA
        }
        j = j+1
    }
    soma = soma - sequencia[i];
    i = i+1;
}
SE soma == VALOR ENTÃO
    Subsequência existe na posição i+1
SENÃO Subsequência não existe
```