

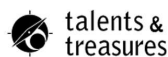
# Torneio Estudantil de Computação MultiLinguagem de Aveiro



Fase Final

Águeda, 20 de fevereiro de 2019

11h30





# Conteúdo

<b>Equipa</b>	<b>iv</b>
Comissão Organizadora . . . . .	iv
Comissão Científica . . . . .	iv
Estudantes Voluntários . . . . .	v
Responsáveis Locais . . . . .	vi
<b>Observações iniciais</b>	<b>viii</b>
Competição . . . . .	viii
Compilação . . . . .	ix
Linguagens . . . . .	ix
Restrições . . . . .	ix
Linha de comando . . . . .	ix
Versões . . . . .	x
Editores disponíveis . . . . .	x
Restrições de execução . . . . .	xi
Especificação dos dados de entrada . . . . .	xi
Especificação dos dados de saída . . . . .	xi
<b>Participantes</b>	<b>xii</b>
<b>Problemas</b>	<b>1</b>
A    Quantos Litros? . . . . .	2
B    Número de ouro . . . . .	4
C    Cifra de Substituição . . . . .	6
D    Vizinhos mais próximos . . . . .	8
E    Formação Profissional . . . . .	11
	iii

# Equipa

## Comissão Organizadora

- Ana Rita Calvão, ESTGA-UA
- Ana Rita Santos, ESTGA-UA
- Ciro Martins, ESTGA-UA
- Cristina Guardado, ESTGA-UA
- Fábio Marques, ESTGA-UA
- Hélder Gomes, ESTGA-UA
- Magda Monteiro, ESTGA-UA

## Comissão Científica

- Ana Rita Calvão, ESTGA-UA
- Ana Rita Santos, ESTGA-UA
- António Barbeito, ESTGA-UA
- Ciro Martins, ESTGA-UA
- Fábio Marques, ESTGA-UA
- Hélder Gomes, ESTGA-UA
- Luís Jorge Gonçalves, ESTGA-UA
- Magda Monteiro, ESTGA-UA
- Mário Rodrigues, ESTGA-UA
- Valter Silva, ESTGA-UA

## Estudantes Voluntários

- Nuno Oliveira, CTeSP em Programação de Sistemas de Informação
- Tiago Vieira, CTeSP em Programação de Sistemas de Informação

## Responsáveis Locais

- Abílio Silva, Escola Secundária de Felgueiras
- Alexandra Laranjeira, Agrupamento de Escolas D. Dinis
- Artur Freitas, Escola Secundária Lima-de-Faria
- Carlos António Oliveira Rainho, Agrupamento De Escolas De Albergaria-a-Velha
- Carlos Malta, Escola Secundária de Emídio Navarro
- Cristina Amaral, Agrupamento de Escolas Dr. Manuel Gomes de Almeida
- Deolinda Rosa Alves Barbosa, Agrupamento de Escolas de Lousada
- Graciano Fangueiro da Silva Torrão, Agrupamento de Escolas D. Afonso Sanches
- Helena Pereira, Agrupamento de Escolas Fontes Pereira de Melo
- Irene Maria dos Santos Baptista, Agrupamento de Escolas Ferreira de Castro
- Joana Marques, Escola Secundária Tomaz Pelayo
- João Sá, Escola Secundária de Avelar Brotero
- Jorge Carvalho, INETE - Instituto de Educação Técnica
- Jorge Duque, Escola Secundária Cacilhas-Tejo
- Jorge Henriques, Escola Secundária Adolfo Portela
- Júlio Magalhães, Agrupamento de Escolas A Beira Douro
- Luís Cerejeira, Agrupamento de Escolas D. Sancho I
- Manuel Marinho, Escola Profissional de Aveiro
- Manuel Teixeira, Agrupamento de Escolas Soares Basto
- Marco Silva, Escola Profissional de Carvalhais
- Margarida Lemos, Escola Secundária de Carvalhos
- Michael Teixeira, Escola Técnico Profissional de Cantanhede
- Miguel Ângelo, Colégio Internato dos Carvalhos
- Nélia Leitão, Agrupamento de Escolas da Lourinhã

- Nuno Oliveira, Escola Profissional de Tondela
- Paula Maria Vasconcelos da Silva, Agrupamento de Escolas de Padre Benjamim Salgado
- Paulo Filipe Abelheira Esteves, Agrupamento de Escolas de Vouzela e Campia
- Pedro Ferreira, Agrupamento De Escolas De Castelo De Paiva
- Rui Mesquita, Agrupamento de Escolas n.º 1 de Gondomar

# Observações iniciais

## Competição

O TECLA é uma competição organizada pela Escola Superior de Tecnologia e Gestão de Águeda, da Universidade de Aveiro, com o intuito de sensibilizar os alunos do ensino secundário ou equivalente para a área da programação de computadores.

O TECLA segue, de uma forma geral, as regras do International Collegiate Programming Contest (ICPC). É uma prova que se realiza em duas fases de 2 horas cada. A fase preliminar teve lugar nas escolas de origem dos participantes e permitiu escolher as equipas que disputam a fase final. A fase final realiza-se nas instalações da ESTGA, em que participam as 25 equipas melhor classificadas na fase preliminar e sendo a participação limitada a três equipas por escola. Foram igualmente apuradas as 5 melhores equipas do 10º ano (constituídas exclusivamente por alunos do 10º ano), para além de eventuais equipas do 10º ano apuradas pelo critério anterior. A participação das equipas do 10º ano é limitada a uma equipa por escola.

Os problemas são elaborados pela Comissão Científica do TECLA. Durante a prova é utilizado o Mooshak, um juiz automático desenvolvido pela Universidade do Porto<sup>1</sup>. Este sistema permite a avaliação instantânea das submissões, devolvendo o resultado da avaliação: *Accepted*, *Presentation Error*, *Runtime Error*, *Compile Time Error*, *Time Limit Exceeded*, *Memory Limit Exceeded*, *Wrong Answer* e *Invalid Function*.

Durante a prova os membros da Comissão Científica poderão ser questionados relativamente aos problemas. As equipas classificam-se de acordo com o número de problemas aceites e pelo tempo que necessitaram para os resolver. Existe uma penalização de 5 minutos por cada submissão não aceite.

---

<sup>1</sup><http://mooshak.dcc.fc.up.pt/>



## Compilação

### Linguagens

- C
- C++
- C#
- Java
- Pascal
- Python
- VB.NET

### Restrições

- Tempo de compilação máximo: 5 segundos
- Tamanho máximo do código fonte: 100 kb
- O código fonte deve ser submetido num único ficheiro

### Linha de comando

- gcc \$file -lm
- g++ \$file
- mcs \$file
- javac -nowarn \$file
- fpc -v0 -oprog \$file
- python3 \$file (execução)

- vbnc \$file

## Versões

- (C, C++) gcc 7.3.0
- (Java) jdk 10.0.2
- (Pascal) FreePascal 3.0.4
- (C#) Mono C# Compiler 4.6.2.0
- (Python) Python 3.6.7
- (VB.Net) Visual Basic.Net Compiler 0.0.0.5943

## Editores disponíveis

- C, C++
  - CodeBlocks
  - Geany
  - Mono Develop
- C#
  - Geany
  - Mono Develop
- Java
  - Geany
  - NetBeans
- Python
  - Geany
  - Pycharm Edu
- Pascal
  - Geany
- VB.NET
  - Mono Develop

## Restrições de execução

- Tempo máximo de CPU: 1 segundo
- Memória disponível para variáveis dinâmicas e globais: 16 MB
- Memória disponível para variáveis locais e pilha de execução: 2 MB

## Especificação dos dados de entrada

Para cada problema existe uma secção específica que indica a estrutura precisa dos dados de entrada utilizados pelo Mooshak. Existem algumas regras gerais que são utilizadas em todos os conjuntos de dados de entrada:

- A última linha dos dados de entrada termina sempre com um "\n" (mudança de linha);
- O espaço em branco, exceto quando indicado explicitamente, é utilizado como separador. Uma linha nunca começa ou acaba com um espaço em branco.

Não é necessário verificar a validade dos dados de entrada. Os casos de teste verificam sempre as regras e restrições definidas para cada problema.

## Especificação dos dados de saída

Tal como para a especificação dos dados de entrada, a especificação dos dados de saída tem uma secção própria em cada um dos problemas. A última linha dos dados de saída termina sempre com um "\n" (mudança de linha).

# Participantes

## **Agrupamento de Escolas D. Afonso Sanches**

Delta2

- Ricardo Gomes
- Pedro Castro

TeamRua

- Gonçalo Filipe dos Santos Garrido de Azevedo
- João Miguel Ramos Barbosa

WeFeed

- André Eiras
- Vasco Capela

WhatEver

- Rafael Pereira
- Rui Martins

## **Agrupamento de Escolas D. Sancho I**

AEDS1\_ONE

- Bruno Campos Gomes
- Mariana Campos da Silva

KB

- Alexandre Granja
- Francisco Carneiro

MOTARS

- João Mota
- Nelson Carneiro

**Agrupamento de Escolas de Padre Benjamim Salgado**

DoubleByte

- Sérgio Miguel Machado Oliveira
- Venâncio Filipe Sousa Carneiro

Madlads

- David Ferreira Sá Machado
- Nuno Miguel Carvalho Moreira

**Agrupamento de Escolas Soares Basto**

Bateitaz

- Gonçalo Brandão
- Alexandre Borges

JG

- João Correia
- Gabriel Florêncio

**Agrupamento de Escolas de Lousada**

SubToPewdiepie

- Miguel Marques Faria Barbosa
- Mariana Beatriz Moreira Marques

**Colégio Internato dos Carvalhos**

12IF

- Miguel Alves
- Gaspar Rocha

Best\_T1

- José Ribeiro
- Rui Costa

Haunted\_Comp

- Duarte Nobrega
- Pedro Miguel C. S. Pereira

UmaBoaPergunta

- Nuno Alves
- Paulo Costa

### **Escola Secundária de Avelar Brotero**

LMAO

- Joaquim Rodrigo Rodrigues Milheiro
- Paulo Xavier Matias Garcia

Os\_Patuscos

- Carlos André Mendes A. Roxo
- Nuno Gabriel da Silva T. Veloso

PRODE

- José António Proença Lopes
- Samuel dos Santos Carinhas

### **Escola Secundária de Carvalhos**

Sem\_mae

- Nuno Soares
- Rafael Monteiro

### **Escola Secundária Cacilhas-Tejo**

Sub2Pewds

- Diogo Silva
- Alexandre Pais

**Escola Secundária de Emídio Navarro**

Bruuutos

- Duarte Veiga Santos
- Carlos Pimentel Martins

NoName01

- André Miguel Correia Gonçalves
- Alexandre Valejo Moreira

sEND\_MEemes

- João Pedro Costa Galega
- Pedro Alexandre Santos Amaral

Unnamed

- Lucas de Almeida da Silva
- Pedro Miguel Gomes Carmo

**Escola Secundária Lima-de-Faria**

Dann610

- André Daniel Taipina Costa
- João Miguel Ferreira Castelo Branco Catré

iTeam

- Miguel Ângelo Dinis Barroso
- Simão Pedro Neto Gonçalves

WhileTrueWinner

- Bernardo Taipina Carvalho Ribeiro
- Bruno Emanuel Neves dos Santos

**INETE - Instituto de Educação Técnica**

ADB

- Tomás Evangelista Bota
- João Nuno Monteiro

include\_tgpsi\_h

- Tomás Vicente
- Pedro Vinhas



# Problemas

## A Quantos Litros?

### Introdução

Considere uma pista de automobilismo onde se vai realizar uma prova na qual está a pensar participar. Mas primeiro precisa de estimar a quantidade de combustível que necessita para poder concluir a prova.

### Problema

Dados o comprimento da pista, o número de voltas da prova, o consumo médio do seu carro e uma margem de segurança (valor extra de combustível para garantir que o carro não fica sem combustível em qualquer circunstância), calcule a quantidade de gasolina que necessita para realizar a prova.

### Dados de entrada

A entrada consiste em quatro linhas, em que a primeira contém o comprimento da pista, em quilómetros, a segunda contém o número de voltas da prova, a terceira contém o consumo médio do seu carro, em litros por 100 quilómetros, e a quarta contém a margem de segurança, em percentagem.

### Restrições

O comprimento da pista é um valor inteiro maior que zero (0) e menor ou igual a mil (1000). Os restantes parâmetros de entrada são inteiros com um valor maior que zero (0) e menor ou igual a cem (100).

### Dados de saída

Uma linha com a quantidade de gasolina necessária para realizar a prova, arredondada para duas casas decimais.

### Exemplo de dados de entrada e de dados de saída 1

Input	
1	5
2	30
3	20
4	12

```
1 | 33.60 | Output
```

### Exemplo de dados de entrada e de dados de saída 2

	Input
1	10
2	25
3	30
4	15

	Output
1	86.25

## B Número de ouro

### Introdução

O Número de Ouro ( $\phi = \frac{1+\sqrt{5}}{2}$ ) é um número irracional misterioso e enigmático que nos surge numa infinidade de elementos da natureza na forma de uma razão, sendo considerado por muitos como uma oferta de Deus ao mundo. Em geometria, o retângulo de ouro surge quando a divisão entre o seu comprimento ( $C$ ) e a sua largura ( $L$ ) é aproximadamente igual ao número de ouro.

### Problema

Dados o comprimento  $C$  e a largura  $L$  de um retângulo, o programa deverá indicar “sim” se o valor absoluto da diferença entre a razão  $C/L$  e o número de ouro não for maior do que 0,01 e “nao” no caso contrário.

### Dados de entrada

A entrada consiste em duas linhas, a primeira corresponde ao valor de  $C$  e a segunda ao valor de  $L$ .

### Restrições

- Os dados de entrada são números reais positivos
- $C > L$

### Dados de saída

Uma linha com a resposta encontrada.

### Exemplo de dados de entrada e de dados de saída 1

	Input
1	3.25
2	2

	Output
1	sim

### Exemplo de dados de entrada e de dados de saída 2

	Input
1	1.35
2	0.28
	Output
1	nao

## C Cifra de Substituição

### Introdução

Uma cifra de substituição é uma cifra em que cada caracter de um texto em claro é substituído por um caracter de um alfabeto de criptograma. Para o nosso caso, vamos considerar que a substituição é produzida pela aplicação da formula  $C=(M+K) \bmod 26$ , em que 26 é o número de letras do alfabeto, M um caracter da mensagem com um valor entre 0 e 25, K (a chave) um número inteiro positivo e C o caracter do criptograma resultante, com um valor entre 0 e 25.

### Problema

A sua tarefa é implementar um programa que cifre uma mensagem usando a cifra de substituição acima especificada. Sugestão: converta cada letra da mensagem num número entre 0 e 25 e faça a operação inversa, após a operação de cifra, para obter a letra do criptograma.

### Dados de entrada

A entrada consiste em duas linhas: a primeira com a chave K e a segunda com a mensagem a cifrar.

### Restrições

- A chave K é um número inteiro maior do que zero.
- A mensagem a cifrar tem um tamanho máximo de 50 caracteres e é composta exclusivamente pelos caracteres alfabéticos de A a Z, não contendo algarismos, minúsculas, caracteres com diacríticos (cedilhas, acentos, etc.) ou caracteres de pontuação ou separação (espaços, vírgulas, etc.).

### Dados de saída

O texto da mensagem cifrada (o criptograma resultante).

### Exemplo de dados de entrada 1

		Input	
1	3		
2	TECLAEMAGUEDA		

### Exemplo de dados de saída 1

		Output	
1	WHFODHPDJXHGD		

### Exemplo de dados de entrada 2

		Input	
1	25		
2	ESCOLASUPERIORDETECNOLOGIAEGESTAODEAGUEDA		

### Exemplo de dados de saída 2

		Output	
1	DRBNKZRTODQHNQCDSDBMNKNFHZDFDRSZNCZFTDCZ		

## D Vizinhos mais próximos

### Introdução

As moradias de uma urbanização estão dispostas em forma de matriz do tipo  $m \times n$ , formando  $(m - 1) \times (n - 1)$  ruas.

Os vizinhos mais próximos de uma habitação são todas as habitações que a rodeiam. No mínimo uma habitação tem 3 vizinhos mais próximos, que ocorre quando a habitação está nas pontas da primeira e última ruas, e no máximo tem 8 vizinhos mais próximos, que ocorre quando a habitação em causa está no centro de 9 habitações. Por exemplo, na figura abaixo, o bairro tem duas ruas e as habitações 1 e 12 têm apenas 3 vizinhos mais próximos enquanto que as habitações 5 e 8 têm 5 vizinhos mais próximos e as habitações 6 e 7 têm 8 vizinhos mais próximos.

1	2	3	4
5	6	7	8
9	10	11	12

### Problema

O presidente da junta pretende ter a informação dos vizinhos mais próximos de cada habitação de uma urbanização que se dispõe em forma de matriz e para tal pede a vossa ajuda.

A matriz deve ter exatamente  $m$  linhas e  $n$  colunas. Pretende-se que construam um programa que, dados as dimensões  $m$  e  $n$  e a matriz com o número de cada habitação, retorne para cada habitação os números das habitações que são seus vizinhos mais próximos.

### Dados de entrada

A entrada consiste em  $m + 1$  linhas. A primeira linha é composta pelas dimensões  $m$  e  $n$ , separadas por um espaço, da matriz.

As  $m$  linhas seguintes, com  $n$  valores, correspondem aos números das habitações da urbanização.



## Restrições

- $2 \leq m, n \leq 15$

## Dados de saída

A saída é constituída por  $(m \times n)$  linhas. A primeira linha começa com o menor número da habitação seguida de ":" e um espaço. Os restantes números dessa linha correspondem aos números (ordenados por ordem crescente) das habitações que são os vizinhos mais próximos. As restantes linhas têm a mesma formatação e correspondem às restantes habitações ordenadas por ordem crescente.

## Exemplo de dados de entrada e de dados de saída 1

	Input
1	3 4
2	1 2 3 4
3	5 6 7 8
4	9 10 11 12

	Output
1	1: 2 5 6
2	2: 1 3 5 6 7
3	3: 2 4 6 7 8
4	4: 3 7 8
5	5: 1 2 6 9 10
6	6: 1 2 3 5 7 9 10 11
7	7: 2 3 4 6 8 10 11 12
8	8: 3 4 7 11 12
9	9: 5 6 10
10	10: 5 6 7 9 11
11	11: 6 7 8 10 12
12	12: 7 8 11

## Exemplo de dados de entrada e de dados de saída 2

	Input
1	4 3
2	21 23 25
3	22 24 26
4	27 29 31
5	28 30 32

	Output
1	21: 22 23 24
2	22: 21 23 24 27 29
3	23: 21 22 24 25 26
4	24: 21 22 23 25 26 27 29 31
5	25: 23 24 26
6	26: 23 24 25 29 31
7	27: 22 24 28 29 30
8	28: 27 29 30
9	29: 22 24 26 27 28 30 31 32
10	30: 27 28 29 31 32
11	31: 24 26 29 30 32
12	32: 29 30 31

## Exemplo de dados de entrada e de dados de saída 2

	Input
1	2 5
2	52 56 58 62 64
3	53 55 59 61 63

	Output
1	52: 53 55 56
2	53: 52 55 56
3	55: 52 53 56 58 59
4	56: 52 53 55 58 59
5	58: 55 56 59 61 62
6	59: 55 56 58 61 62
7	61: 58 59 62 63 64
8	62: 58 59 61 63 64
9	63: 61 62 64
10	64: 61 62 63

## E Formação Profissional

### Introdução

O Instituto de Emprego e Formação Profissional (IEFP), numa estratégia de diminuição da taxa de desemprego, decidiu implementar o seguinte procedimento de seleção de candidatos para frequência de formação profissional: todos os  $N$  desempregados inscritos no IEFP são colocados num círculo, voltados para dentro. Um desses desempregados é escolhido arbitrariamente como número 1, sendo os restantes numerados no sentido horário até  $N$  (os quais estarão à esquerda do 1º).

Partindo do 1º e movendo-se no sentido horário, um funcionário do IEFP conta  $k$  posições e retira um candidato, enquanto outro funcionário começa a partir de  $N$  e move-se no sentido anti-horário, contando  $m$  posições e retirando outro candidato. Os dois candidatos escolhidos são então enviados para um determinado programa de formação profissional. Se ambos os funcionários escolherem o mesmo candidato, é enviado somente um candidato. Cada funcionário, em seguida, começa a contar novamente a partir do próximo candidato disponível e o processo continua até que não restem mais candidatos. De notar que os dois candidatos selecionados em cada ronda deixam o círculo ao mesmo tempo. Por isso, é possível que o segundo funcionário conte o candidato já selecionado pelo primeiro funcionário.

### Problema

Pretende-se que implemente um programa que leia três números ( $N$ ,  $k$  e  $m$ ) e determine a ordem pela qual os desempregados são selecionados para formação.

### Dados de entrada

A entrada do programa é constituída por uma linha contendo os três números inteiros  $N$ ,  $k$  e  $m$ .

### Restrições

- $0 < N < 20$
- $0 < k, m < 500$

### Dados de saída

A saída deverá mostrar uma linha de números especificando a ordem pela qual os de-

sempregados foram selecionados. Liste o par escolhido partindo do desempregado escolhido pelo funcionário do sentido horário. Os pares sucessivos são separados pelo carácter # (mas não deverá existir # após a última escolha), existindo uma vírgula entre cada um dos números do par.

### Exemplo de dados de entrada e de dados de saída 1

	Input
1	10 4 3
	Output
1	4,8#9,5#3,1#2,6#10#7

### Exemplo de dados de entrada e de dados de saída 2

	Input
1	6 2 8
	Output
1	2,5#4,6#3,1

### Exemplo de dados de entrada e de dados de saída 3

	Input
1	2 2 1
	Output
1	2#1