

TORNEIO ESTUDANTIL DE COMPUTAÇÃO  
MULTI-LINGUAGEM DE AVEIRO



Escola Superior de Tecnologia e Gestão de Águeda  
Universidade de Aveiro

**TECLA 2009**

<http://tecla.estga.ua.pt/>

**Exemplos**

W – Exemplo de Leitura e Escrita

Z – Exemplo de Determinação de Números Mínimo e Máximo



# 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 e 30 minutos cada. A fase preliminar terá lugar nas escolas de origem dos participantes, e permitirá escolher as equipas que irão disputar a fase final. A fase final realizar-se-á nas instalações da ESTGA e terá a participação das 10 melhores equipas da fase preliminar. As equipas a concurso são compostas por dois elementos.

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++
- Java
- Pascal

### Restrições

- Tempo de compilação máximo: 30 segundos
- Tamanho máximo do código fonte: 20 kb
- O código fonte deve ser submetido num único ficheiro

### Linha de comando

- gcc -ansi -lm
- g++ -Wall
- javac -nowarn
- fpc -v0w -oprogram

### Versões

- gcc 4.2.3
- jdk 1.6.0\_07
- FreePascal 2.2.0

## Editores disponíveis

(Informação disponibilizada mais tarde)

---

## 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";
- O espaço em branco, excepto quando indicado explicitamente, é utilizado como separador. Uma linha nunca começa ou acaba com um espaço em branco.

## 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".

## Nota

Os problemas presentes neste enunciado são apenas para exemplificação de leitura dos dados de entrada e escrita dos dados de saída. Em nenhum dos casos exemplificam o grau de dificuldade dos problemas da prova.

As soluções propostas são apenas possíveis soluções. São as soluções que considerámos de mais fácil compreensão, não necessariamente as mais optimizadas e certamente não as únicas.

## Exemplo de Leitura e Escrita<sup>†</sup>

### Introdução

Este problema tem como objectivo exemplificar algumas instruções que podem ser utilizadas para ler e escrever aquando da utilização da ferramenta de validação automática Mooshak.

### Problema

Dada uma expressão, devolva a mesma expressão.

### Dados de entrada

A única linha dos dados de entrada contém a expressão a escrever.

### Dados de saída

Os dados de saída devem conter na primeira linha a expressão dada.

### Exemplo dos dados de entrada

Ola!

### Exemplo dos dados de saída

Ola!

### Resolução:

C

```
#include<stdio.h>
int main(){
    char s[10];
    scanf("%s",s);
    printf("%s\n",s);
    return 0;
}
```

---

<sup>†</sup>não exemplifica o nível de dificuldade dos problemas da prova

## C++

```
#include <iostream>
using namespace std;
int main(void)
{
char s[10];
cin >> s;
cout << s << endl;
return 0;
}
```

## Java

```
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.BufferedReader;

public class a {
    public static void main (String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        String s = in.readLine();
        System.out.println(s);
    }
}
```

## Pascal

```
program a;
var s : String;
begin
    readln(s);
    writeln(s);
end.
```

## Exemplo de Determinação de Números Mínimo e Máximo<sup>†</sup>

### Introdução

Este problema tem como objectivo exemplificar algumas instruções que podem ser utilizadas para ler e escrever aquando da utilização da ferramenta de validação automática Mooshak.

### Problema

Dado um conjunto de 5 números inteiros, determine os números mínimo e máximo.

### Dados de entrada

O ficheiro contém 5 linhas de entrada. Cada linha contém um número inteiro.

### Dados de saída

O ficheiro de saída deve conter na primeira linha o número mínimo e na segunda linha o número máximo.

### Exemplo dos dados de entrada

```
4
2
-1
5
9
```

### Exemplo dos dados de saída

```
-1
9
```

### Resolução:

C

```
#include<stdio.h>
int main()
{
```

---

<sup>†</sup>não exemplifica o nível de dificuldade dos problemas da prova



```
int i, max, min, numero;

for (i = 0; i < 5; i++) {
    scanf("%d", &numero);

    if (i == 0) {
        max = numero;
        min = numero;
    } else {
        if (numero < min) {
            min = numero;
        }
        if (numero > max) {
            max = numero;
        }
    }
}

printf("%d\n",min);
printf("%d\n",max);
return 0;
}
```

## C++

```
#include <iostream>
using namespace std;
int main(void) {
    int i, max, min, numero;
    for (i=0; i < 5; i++) {
        cin >> numero;
        if (i == 0) {
            max = min = numero;
        } else {
            if (numero < min) {
                min = numero;
            }
            if (numero > max) {
                max = numero;
            }
        }
    }
    cout << min << endl << max << endl;
}
```

**Java**

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;

public class z {

    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        int[] numero = new int[5];
        for (int i = 0; i < 5; i++) {
            String line = in.readLine();
            numero[i] = Integer.valueOf(line);
        }
        Arrays.sort(numero);
        System.out.println(numero[0]);
        System.out.println(numero[4]);
    }
}
```

**Pascal**

```
program a;
var i, min, max, numero : Integer;
begin
    for i := 1 to 5 do
        begin
            Readln(numero);
            if i = 1 then
                begin
                    min := numero;
                    max := numero;
                end
            else
                begin
                    if numero < min then
                        min := numero;
                    if numero > max then
                        max := numero;
                    end;
                end;
        end;
end;
```

```
    Writeln(min);  
    Writeln(max);  
end.
```