

 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA</p> <p><i>Departamento de Engenharia Informática</i></p>	<p><b>Trabalho nº 0 de Algoritmos e Estruturas de Dados</b></p> <p><b>2013-2014 – 2º Semestre</b></p> <p><b>Texto de Apoio – Como usar o <i>Mooshak</i></b></p>
--	---

### Objectivos:

Este texto de apoio, enquadrado no TP0 de AED, tem por objectivo dar algum apoio aos alunos que ainda não tenham tido qualquer experiência com o *Mooshak*, a plataforma de submissão de problemas de programação que será usada nos diversos trabalhos práticos da disciplina.

### O que é o *Mooshak*?

O *Mooshak* é uma plataforma para gestão *on-line* de concursos de programação, desenvolvida pela Universidade do Porto. Para além do suporte de competições de programação, tem sido também usada para dar apoio a disciplinas de programação. É nesse contexto que o *Mooshak* será usado em AED.

### Como é usado o *Mooshak* em AED:

Foi instalado um servidor *Mooshak* de suporte as aulas de AED (<http://mooshak.dei.uc.pt/~aed>) e, usando esse servidor, os trabalhos práticos funcionarão da seguinte forma:

- Para cada Trabalho Prático os docentes criam um “concurso” (*contest*).
- Cada concurso pode ter vários problemas, cada um deles com um enunciado específico e com prazos de submissão específicos. O trabalho prático pode ser assim constituído por várias metas (por exemplo: *problemas A e B entregues obrigatoriamente numa data intermédia, problema C entregue 1 semana mais tarde, problema D opcional e sem prazo de entrega*).
- Os alunos, usando as contas previamente criadas, submetem ao *Mooshak* o código-fonte da sua solução para cada problema. O *Mooshak* testa esse código, usando um conjunto de *inputs* de teste previamente definido pelos docentes, e reporta ao aluno se o código funciona de acordo com as especificações do enunciado ou não.
- Os alunos podem fazer sucessivas submissões, até ter o seu código aprovado pelo *Mooshak*, e visualizar todas as submissões efectuadas.
- Os alunos poderão usar o *Mooshak* para colocar questões aos docentes (dúvidas sobre o enunciado ou sobre o trabalho, etc.), que responderão usando também o *Mooshak*. Espera-se que seja este o principal canal de suporte aos Trabalhos (para além das próprias aulas).
- Os docentes podem verificar, no *Mooshak*, o histórico de submissões de cada aluno, o respectivo código-fonte, e o resultado dos testes feitos pelo *Mooshak*. Esses elementos ajudam os docentes a avaliar o trabalho.

Deve no entanto ser referido que o *Mooshak* é apenas um dos instrumentos de avaliação dos Trabalhos Práticos, complementando os relatórios escritos, as defesas dos trabalhos e a avaliação da qualidade do código-fonte submetido (POO, estrutura, clareza, optimização, etc.).

Ter o código *aceite* pelo *Mooshak* não é por si só garantia de uma boa classificação, e não conseguir que o *Mooshak* aceite o código-fonte submetido indicia obviamente alguns problemas na implementação ou interpretação do enunciado mas não implica necessariamente uma avaliação negativa do trabalho do aluno.

### **Como posso obter uma conta no *Mooshak*?**

---

Todos os alunos devem registar-se no *Mooshak* de AED. Para isso deverão:

1. Aceder à página principal (<http://mooshak.dei.uc.pt/~aed>)
2. Seguir a opção “*Register for on-line contest*”
3. Preencher formulário:  
Contest: **TP0**  
Name: **login do aluno na máquina student**  
Email: **endereço do aluno na máquina student**  
Group: **AED**
4. O aluno receberá depois na sua caixa de correio a *password* da sua conta no *Mooshak*
5. Usando essa *password* poderá depois fazer *login* como participante no concurso TP0.

Para os trabalhos práticos (*contests*) seguintes este processo é desnecessário, pois as contas serão automaticamente criadas, por cópia das contas do TP0.

### **Como verifica o *Mooshak* as minhas submissões?**

---

Usando os menus *on-line* o aluno envia para o *Mooshak* o seu código fonte (*.java*, *.c*, etc.) e o *Mooshak* compila esse código e executa-o usando um lote de *inputs* previamente definidos pelos docentes. Cada execução deverá produzir *outputs* exatamente iguais aos que foram previamente definidos pelos docentes (incluindo a formatação do *output*) para que a submissão seja aceite.

Os docentes definem vários pares *input/output* para poder testar todas as situações de funcionamento (casos “normais”, casos “extremos”, situações de exceção, etc.) e os alunos não têm acesso aos *inputs* usados pelo *Mooshak*, para evitar situações de fraude e, principalmente, para estimular uma análise mais cuidada dos enunciados e do código-fonte desenvolvido.

As respostas mais habituais do *Mooshak* às submissões dos alunos são as seguintes:

- **Accepted** – o programa funciona bem em todos os testes que foram feitos.
- **Compile Time Error** – o *Mooshak* não conseguiu sequer compilar o código, tendo sido gerado um erro ou um aviso.
- **Runtime Error** – a execução do programa terminou prematuramente, com uma excepção.
- **Time Limit Exceeded** – o programa não terminou dentro do limite de tempo dado (pode estar por exemplo “preso” num ciclo infinito ou ter um algoritmo excessivamente ineficiente, que penaliza de forma notória o tempo de execução e que deve ser optimizado).
- **Invalid Submission** – o ficheiro do código-fonte submetido tem uma extensão inválida ou correspondente a uma linguagem de programação não suportada.
- **Wrong Answer** – o *output* do programa não é igual ao esperado, pelo menos para um dos testes efectuados. Isso significa que o programa produz *outputs* errados ou, pelo menos, *outputs* cuja formatação não está de acordo com o definido no enunciado.

- **Presentation Error** – o programa produz *outputs* corretos mas com problemas de formatação (espaços a mais, por exemplo). Nem sempre o *Mooshak* consegue distinguir entre “*Presentation Errors*” e “*Wrong Answers*”, reportando nesses casos “*Wrong Answer*”.

### Como deve o meu programa ler o *input* e produzir o *output*?

---

Para cada execução o *Mooshak* lê o *input* na entrada *default* (*stdin*) e produz o *output* na saída *default* (*stdout*).

Um erro típico de principiante é o uso de menus interativos (por exemplo coisas do tipo “Digite o número primo a converter”) ou leitura explícita de ficheiros no código submetido (por exemplo “Indique o nome do ficheiro de *input*:”), que não são aceites pelo *Mooshak*.

Nos testes prévios que os alunos façam, antes de submeter o seu código no *Mooshak*, podem usar:

- *input* diretamente pelo teclado/console e *output* diretamente na consola de linha de comando, nos casos em que a dimensão do *input* seja reduzida.
- Usar a consola de linha de comando e redirecionar o *input* e *output* para ficheiros, por exemplo com comandos do tipo:

*c:\aed> java HelloWorld < fichInput.txt > fichOutput.txt*

- Ou, no IDE que estejam a usar, definir um ficheiro de texto como fonte *stdin*.

### Que linguagens de programação usar?

---

O *Mooshak* suporta várias linguagens de programação (C, C++, Java, *Python*, etc.). Nos trabalhos práticos, salvo indicações em contrário por parte dos docentes, os alunos são livres de escolher a linguagem de programação usada na resolução do problema, dado que a disciplina se foca essencialmente na algoritmia. No entanto, é importante notar que a qualidade do apoio prestado pelos docentes aos alunos poderá apresentar variações, consoante a maior ou menor experiência do docente com a linguagem que o aluno esteja a usar.

### Método *readLn*

---

Nos casos em que seja usado Java, para facilitar a leitura do *input* linha a linha, os alunos poderão integrar nas suas soluções o método estático *readLn()* que aqui se apresenta:

```
static String readLn (int maxLg){ //utility function to read from stdin
{
    byte lin[] = new byte [maxLg];
    int lg = 0, car = -1;
    String line = "";
    try {
        while (lg < maxLg){
            car = System.in.read();
            if ((car < 0) || (car == '\n')) break;
            lin [lg++] += car;
        }
    }
    catch (IOException e){
        return (null);
    }
    if ((car < 0) && (lg == 0)) return (null); // eof
    return (new String (lin, 0, lg));
}
```