



Notas

Se não conseguires submeter uma solução via CMS, por favor faz o seguinte:

- Coloca a tua submissão numa pasta chamada `submit_<task name>` no Desktop (a pasta já deve existir) **antes do concurso terminar**.
 - Para os problemas de "output-only", os ficheiros devem chamar-se `01.out`, `02.out`, ..., `10.out`.
 - Para os outros problemas, deve existir **exatamente um** ficheiro na pasta correspondente, contendo o código fonte.
- Pede ao teu team leader para submeter um appeal.

Se achas que te deve ser dado tempo extra, por favor faz o seguinte:

- Envia um pedido de clarificação (através do CMS ou em papel) tão cedo quanto possível.
- Não saís da tua mesa de trabalho e não fales com nenhum outro concorrente quando o concurso terminar.

Cada problema tem um arquivo em anexo (ficheiro comprimido) disponível do qual podes fazer download a partir do CMS.

Existem algumas tarefas de "output-only", nas quais:

- O arquivo em anexo contém o casos de teste e os casos de exemplo. Cada caso de teste é uma subtarefa separada.
- Podes submeter múltiplos ficheiros de output como um arquivo zip. Neste caso, os teus ficheiros de output devem ser chamados de `??out`, onde `??` é o número do caso de teste (ex: `03.out`). Podes criar um arquivo zip com múltiplos ficheiros usando o seguinte comando: `zip output.zip *.out`
- Podes fazer até 100 submissões para os problemas de "output-only". Em cada submissão, podes submeter os ficheiros de output para qualquer subconjunto dos casos de teste.

Para os outros problemas:

- O arquivo em anexo contém os avaliadores exemplo, implementações exemplo, casos de teste exemplo e scripts de compilação.
- Podes fazer até 50 submissões para cada problema, e tens de submeter exatamente um ficheiro em cada submissão.
- O nome do ficheiro que deves submeter é dado no cabeçalho do enunciado. O

ficheiro deve implementar as funções descritas no enunciado correspondente usando as assinaturas dadas nas implementações exemplo.

- Podes implementar outras funções.
- As tuas submissões não devem ler do standard input, escrever para o standard output, ou interagir com qualquer outro ficheiro. No entanto, podem escrever para o standard error.
- As tuas submissões **não devem chamar `exit()` ou `System.exit()`.**
- Quando estiveres a testar com o avaliador exemplo, o teu input deve obedecer ao formato e restrições do enunciado, caso contrário comportamentos inesperados podem acontecer.
- Nos inputs do avaliador exemplo, dois "tokens" consecutivos numa linha são separados por um único espaço, a não ser que outro formato seja explicitamente dado.
- Quando testares o teu código na tua máquina local, recomendamos que uses os scripts de compilação dados no arquivo zip. De outro modo, especialmente no caso do C++, não te esqueças de adicionar a opção `-std=gnu++14` ao compilar.

Convenções

Os enunciados especificam as assinaturas de funções usando os tipos genéricos `void`, `int`, `int64`, e `int[]` (array).

Para cada uma das linguagens de programação suportada, os avaliadores usam os tipos de dados ou implementações apropriadas, tal como listadas em baixo:

Linguagem	<code>void</code>	<code>int</code>	<code>int64</code>	<code>int[]</code>	tamanho do array <code>a</code>
C++	<code>void</code>	<code>int</code>	<code>long long</code>	<code>std::vector<int></code>	<code>a.size()</code>
Java	<code>void</code>	<code>int</code>	<code>long</code>	<code>int[]</code>	<code>a.length</code>

Limites

Problema	Limite de Tempo	Limite de Memória
line	output-only	output-only
vision	1 seg	1024 MB
walk	4 seg	1024 MB