

---

**Trabalho Prático de Avaliação n.º 1**

---

**Descrição:**

O objetivo deste trabalho é a implementação de **um** algoritmo rápido para a ordenação de arrays de grandes dimensões utilizando a linguagem C (ISO) e o compilador *gcc* em ambiente Linux.

Poderão ser utilizados algoritmos que não foram apresentados na aula. **No entanto**, no relatório terá de explicar o algoritmo, quais as modificações realizadas para melhoria de performance, e a análise da complexidade assintótica do mesmo. O algoritmo de ordenação utilizado pelo estudante tem de ser claramente compreendido e explicado pelo próprio. Se não compreender o algoritmo não o utilize.

O programa deverá ser executado a partir da linha de comandos de acordo com o seguinte exemplo:

```
$ordena 10000 -500 500  
A ordenação está correta!
```

**ordena** é o nome dado ao programa. Portanto, de acordo com o exemplo, o programa irá gerar um array com 10.000 valores **aleatórios** entre -500 e 500. Obviamente, estes **valores são parâmetros**.

**NOTA:** \$ é a prompt no Linux. Em Windows seria: **c:\ordena 10000 -500 500**

Visto que estes valores numéricos são passados como strings ao *main()*, será necessário convertê-los. Sugere-se a utilização da função **atoi** da library **stdlib.h**.

Deve ser feita uma restrição para não permitir que o limites sejam menores que -N, ou maiores que N, onde N é o tamanho do array.

Após a ordenação, deverá ser feita a validação de que o array foi corretamente ordenado, e o utilizador deverá ser informado. De acordo com o exemplo anterior, após a ordenação verificou-se que a ordenação está correta, resultando no output da mensagem: **A ordenação está correta!** Esta verificação deverá ser feita por uma função que devolverá 1 caso o array esteja ordenado ou 0, caso contrário.

O programa deverá ter pelo menos três funções além da função *main()*. Uma onde se gera e preenche o array aleatoriamente, outra onde é feita a ordenação, e outra que faz a validação da ordenação (Atenção que aqui utiliza-se a palavra função, mesmo que não haja retorno).

O relatório não deverá conter código fonte. Poderá e deverá, no entanto, conter pseudocódigo

relativo a partes importantes do programa (por exemplo, o algoritmo de ordenação).

Caso seja oportuno, poderão ser apresentados testes de comparação com um outro algoritmo, ou com a versão normal do algoritmo que foi melhorada pelo estudante, sendo que, neste caso é necessário indicar as características importantes do computador onde os testes foram realizados.

Para utilizadores de *Latex* na realização do relatório, aconselha-se a utilização do pacote **clrscode** para a redação dos algoritmos.

### Prazos:

O trabalho deverá ser, preferencialmente, realizado individualmente. No entanto, são aceites grupos de dois estudantes. Quem desejar realizar o trabalho em grupos de dois deverá enviar um email ao docente até ao dia **16 de Maio de 2012** a informar a constituição do grupo. Após essa data não são aceites mais inscrições. Os alunos que pretendam realizar o trabalho de forma individual não precisam de enviar email.

Dúvidas **sobre o enunciado** que não sejam satisfeitas pelo enunciado, poderão ser colocadas por email ou pessoalmente até ao dia **20 de Maio de 2012**.

O relatório deverá ser entregue em papel até ao dia **24 de Maio de 2012**.

**Os trabalhos serão defendidos** no dia **28 de Maio**. No caso de trabalhos não individuais, cada um dos **dois membros** do grupo participarão obrigatoriamente em 50% da defesa, pelo que as notas serão individualizadas.

Trabalhos entregues fora dos prazos estabelecidos terão uma penalização de **10% por dia**, na nota final global.

### Avaliação:

O programa será testado no portátil do docente. O teste consistirá em correr o programa 4 vezes em arrays de tamanhos a definir pelo docente, e o tempo médio será o tempo considerado. Os 3 programas mais rápidos terão um bónus de 0,5 valores **sujeitos à qualidade do relatório/defesa**. Se o mais rápido for 3 vezes mais rápido do que os restantes, terá mais um bónus de 0,5 valores, também sujeitos à qualidade do relatório/defesa. O melhor trabalho será apresentado pelo estudante à turma, podendo auferir até mais 1 valor pela apresentação.

Caso sejam ultrapassados os 20 valores, os valores em sobra serão creditados no próximo

trabalho prático.

Na avaliação, além dos bónus, serão levados em conta os seguintes aspetos:

- Qualidade do relatório;
- Qualidade da defesa; Utilização de variáveis/constantes globais (quantas menos, melhor);
- Estruturação e qualidade do programa (bem “dividido” em funções);
- Comentários (muitos comentários não é sinónimo de bons comentários).

Bom trabalho!