



Introdução aos Algoritmos e Estruturas de Dados

(<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester>)

Guidelines

Compilação

- O programa deve ser compilado usando o comando
`gcc -O3 -Wall -Wextra -Werror -Wno-unused-result`.
- O compilador não deve apresentar erros ou avisos ("warnings").
- Recomenda-se usar a versão 7, ou superior, do compilador (consulte `gcc --version`).
- O sistema de avaliação automática usa a versão 12.3 do gcc.

Formatação

- A formatação do código deve ser **consistente**. Deve utilizar o tabulador e não espaços brancos. Os editores permitem definir quantos espaços devem ser usados para representar o tabulador. Recomenda-se a utilização de um editor de texto com indentação automática.
- Os caracteres acentuados devem ser representados em UTF-8.
- As linhas nunca não deverão exceder um limite de 80 caracteres: `wc -L`.
- Abrir cada chaveta na mesma linha do cabeçalho da função. Fechar cada chaveta numa linha nova, também alinhada com o cabeçalho. O mesmo aplica-se às chavetas associadas aos comandos `for`, `if`, `switch`, e `while`.

Comentários

- Deve utilizar o formato Doxygen para produzir os comentários do código. Inicialmente, para gerar o ficheiro de configuração **Doxyfile**, execute `doxygen -g`. Pode alterar os parâmetros em **Doxyfile**, nomeadamente **PROJECT_NAME** e **EXTRACT_ALL**. Para gerar a documentação executar `doxygen Doxyfile`. A documentação em **HTML** encontra-se em `html/index.html`.
- O código deve que ser adequadamente comentado. Comentários em excesso devem ser evitados. Por exemplo, não é necessário comentar sobre uma variável que representa um contador de um ciclo.
- Cada ficheiro deve ter um comentário, no início, com uma descrição sucinta e o(s) nome(s) do(s) autor(es).
- Cada função deve ter um comentário, antes do cabeçalho, com uma descrição sucinta.
- Cada variável global e/ou estática deve ter um comentário com uma descrição sucinta. O comentário pode ser colocado na mesma linha da declaração da variável, ou na linha anterior.

Organização do código

- Não deve haver repetição de código (código repetido deverá ser colocado numa função).
- Deve poupar memória, evitando repetições de valores.
- Evite funções demasiado longas (e pouco legíveis).
- Recomenda-se usar a ferramenta lizard (<http://www.lizard.ws/>), *pip install lizard*, para calcular a complexidade ciclométrica e o número de linhas de código. A ferramenta não deve emitir avisos.

```
lizard -L 50 -T nloc=30 -C 12 -m proj.c
```

Constantes

- Constantes (números, strings) nunca devem aparecer directamente no código. Deverá ser usada a directiva `#define` no início dos ficheiros ou num ficheiro `.h` separado.

Exemplo

```
/**
 * A program exemplifying the use of Doxygen comments in C.
 * @file ex.c
 * @author ist1xxxxx (first-name last-name)
 */
#include<stdio.h>

/** The maximum number of values stored. */
#define SZ 100

/** Array of values. */
int vals[SZ];

/** The number of values stored. */
int count;

/**
 * Adds a given value to the global vector returns the new count of values.
 * @param val value to add to global vector
 * @return The number of values in the global vector
 */
int add(int val) {
    vals[count] = val;
    return ++count;
}

/**
 * Reads n values from stdin and inserts them into the vector vals.
 * @return Always returns 0
 */
int main() {
    int n, v;
    scanf("%d", &n);
    while (n--) {
        scanf("%d", &v);
        add(v);
    }
    return 0;
}
```

Opção fsanitize

A opção `fsanitize` é uma ferramenta útil para analisar o projecto, em particular erros de memória.

Para analisar um teste que está a falhar, deverá efetuar os seguintes passos:

1. Compilar com as flags `-g -fsanitize=address`, e.g.
`gcc -g -fsanitize=address -Wall -Wextra -Werror -Wno-unused-result -o proj proj.c`
2. executar o projecto compilado usando o teste que está a falhar como standard input, e.g.

```
$ ./proj < testes_publicos/teste27.in
```

Podem também ignorar o output do projecto por forma a ver só os erros da seguinte forma:

```
$ ./proj < testes_publicos/teste27.in > /dev/null
```

Valgrind

É aconselhável analisar o projecto usando também a ferramenta Valgrind. Para instalar no Ubuntu:

```
$ sudo apt install valgrind
```

Para analisar com o valgrind um teste que está a falhar, deverá efectuar os seguintes passos:

1. Compilar com a flag `-g`, e.g. `gcc -g -Wall -Wextra -Werror -Wno-unused-result -o proj proj.c`.
2. executar o projecto compilado com o Valgrind usando o teste que está a falhar como standard input, e.g.

```
$ valgrind ./proj < testes_publicos/teste27.in
```

Podem também ignorar o output do projecto por forma a ver só os erros da seguinte forma:

```
$ valgrind ./proj < testes_publicos/teste27.in > /dev/null
```

Nota: O valgrind neste momento não tem suporte para Macs. Vejam a FAQ na página da disciplina sobre como instalar a aplicação leaks para Mac. Outras alternativas são utilizar uma máquina virtual com ubuntu, os computadores dos laboratórios, o nó sigma (`ssh ist1xxxxx@sigma.tecnico.ulisboa.pt` com password do fénix).

Página Inicial (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/2024-2025/2-semester/rss/announcement>)

Grupos (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/grupos>)

Avaliação (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/avaliacao>)

Bibliografia (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/bibliografia>)

Horário (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/horario>)

Métodos de Avaliação (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/metodos-de-avaliacao>)

Objectivos (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/objectivos>)

Planeamento (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/planeamento>)

Programa (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/programa>)

Turnos (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/turnos>)

Anúncios (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/turnos>)

Sumários (<https://fenix.tecnico.ulisboa.pt/disciplinas/IAED236/2024-2025/2-semester/rss/summary>)