Universidade da Beira Interior
Engenharia Informática — Laboratórios de Programação

Operações com Vetores em C

Rodrigo Marques

Conteúdo

Resumo

Este relatório descreve um programa em C que realiza operações com vetores. O programa foi desenvolvido separando as funções (no ficheiro `functions-PL403.c` e `funcions-PL403.h`) do programa principal (`main-PL403.c`). O programa solicita ao utilizador N números inteiros (onde N é definido pela constante `TAM`), validando a sua entrada entre -3 a 27. O sistema disponibiliza um menu interativo para manipulações do vetor, incluindo operações de ordenação, cálculos e transformações em matrizes. A documentação do programa foi gerada de forma automática utilizando o Doxygen e a compilação é feita através um makefile.

Palavras-chave: C, vetores, matrizes, funções, programação, validação de dados, Doxygen.

Acrónimos

TAM Tamanho do vetor

Capítulo 1

Introdução

1.1 Objetivos do Trabalho

O objetivo principal do trabalho consiste em desenvolver um programa que cumpra os seguintes requisitos técnicos e funcionais:

- Implementar um programa em C que solicita ao utilizador N (onde $N = \texttt{TAM} = 14$) números inteiros no intervalo de -3 a 27.
- Verificar a entrada para todos os números.
- Separar o código em dois ficherios diferentes (`.c` e `.h`).
- Apresentar um menu de operações interativo para manipulação do vetor.
- Incluir uma funcionalidade de ajuda acessível via menu (opção 7) ou através da flag `-help`.
- Gerar a documentação do código de forma automática através do Doxygen.

1.2 Estrutura do Relatório

Este relatório está estruturado da seguinte forma:

- O Capítulo 2 detalha a estrutura do projeto, as tecnologias utilizadas e as funcionalidades implementadas.
- O Capítulo 3 faça dos testes realizados para avaliar o programa.
- O Capítulo 4 resume os resultados.
- O Apêndice contém a documentação gerada automaticamente.

Capítulo 2

Planeamento e Execução

2.1 Tecnologias Utilizadas

- Linguagem de Programação: C;
- Compilador: GCC;
- `Makefile` para automatização da compilação e limpeza.
- Ferramenta de Documentação: Doxygen para geração de documentação HTML.

2.2 Estrutura do Projeto

O projeto tem uma estrutura que separa os ficheiros:

- `main-PL403.c`: Contém a função `main()`;
- `functions-PL403.h`: Ficheiro de cabeçalho que declara a constante `TAM` e as funções de todas que manipulam os vetores;
- `functions-PL403.c`: Ficheiro que contém o corpo das funções.

A compilação do projeto é automatizada pelo `Makefile`, que permite a compilação do executável e a geração da documentação com um único comando (`make`).

2.3 Funcionalidade (`-help`)

Em adição ao menu interativo, foi implementada uma funcionalidade de ajuda.

- A função `main()` foi modificada para aceitar argumentos (`argc` e `argv`).
- Se o utilizador executar o programa com o argumento `-help` (ex: `./main-PL403.exe -help`), o programa chama a função `ajuda()` e termina imediatamente, sem solicitar a introdução do vetor.

2.4 Detalhes de Implementação

O programa foi construído em torno de um vetor de tamanho `TAM`=14. As funcionalidades implementadas são:

- `ordenarvetor`: Ordena o vetor de inteiros por ordem crescente (Método Bubble Sort).
- `simetricovetor`: Calcula e apresenta o vetor que contém os valores simétricos dos elementos originais ordenando-os.
- `somavetor`: Calcula a soma dos elementos da primeira metade do vetor com os elementos da segunda metade.
- `multiplovetor`: Devolve os valores do vetor que estão em posições múltiplas de três.
- `aleatoriovetor`: Retorna um elemento aleatório do vetor.
- `matrizvetor`: Constrói uma matriz NxN, onde cada linha é composta por permutações dos valores do vetor.
- `raizvetor`: Calcula a raiz quadrada de cada elemento do vetor.
- `misturavetor`: Mistura a primeira metade do vetor original com um novo vetor de números inteiros introduzido pelo utilizador.
- `mdcvetor`: Calcula o Máximo Divisor Comum (`mdc`) de dois elementos consecutivos do vetor.
- `matriz2vetor`: Constrói uma matriz que resulta do produto de elemntos do vetor original com o vetor original ordenado de forma crescente.
- `transpostavetor`: Constrói e apresenta a matriz transposta da matriz gerada em `matriz2vetor`.
- `ajuda`: Mostra o menu de ajuda do programa.

2.5 Exemplos de Saída

Para um vetor de exemplo 10, -1, 5, 2, 7, 3}, a função `somavetor` retorna:

- $10 + 2 = 12$
- $-1 + 7 = 6$
- $5 + 3 = 8$

O resultado: $12, 6, 8$.

2.6 Dependências

O projeto depende das bibliotecas padrão de C (`stdio.h`, `stdlib.h`, `time.h`, `math.h`, `string.h`) e do `makefile` para compilação.

Capítulo 3

Testes

3.1 Detalhes dos Testes

Foram definidos testes para cada função:

- Validação de Entrada: Teste de valores válidos (dentro de [-3, 27]) e inválidos (fora do intervalo) para verificar a rejeição correta e a repetição do pedido.
- Verificação Matemática: Verificação da saída correta para operações como soma (`somavetor`), cálculo de `mdc` e raízes (`raizvetor`).
- Funcionalidade do programa: Teste de por exemplo; `ordenarvetor` como auxiliar e teste da flag `-help`.

3.2 Execução dos Testes

Cada função foi executada separadamente para garantir que todas as operações estavam corretas. A funcionalidade `-help` foi testada diretamente na linha de comandos.

3.3 Resultados

Os testes confirmaram que:

- A validação de entrada impede números fora do intervalo.
- As operações matemáticas e de manipulação (como ordenação) retornam resultados corretos.
- A funcionalidade `-help` está a funcionar.
- Os elementos aleatórios variam entre execuções, garantindo o uso correto da função `srand(time(NULL))`.

Capítulo 4

Conclusões

Todos os objetivos foram atingidos com sucesso.

4.1 Ficheiros

- `main-PL403.c`
- `functions-PL403.c`
- `functions-PL403.h`
- `makefile`
- `Doxyfile`

Bibliografia

- Kernighan, B. W., & Ritchie, D. M. (1988). *The C Programming Language.* Prentice Hall.
- Curso de C - Playlist no YouTube.: `https://www.youtube.com/watch?v=2w8GYzBjNj8&list=PLpaKFn4Q4GMOBAeq...`
- https://labex.io/pt/tutorials/cpp-how-to-use-compiler-flags-correctly-434220
- https://www.doxygen.nl/manual/markdown.html
- https://tableless.github.io/iniciantes/manual/html/estruturabasica.html