INSTITUTO FEDERAL
Paraná
Campus Pinhais

Bacharelado em ciência da computação
Sistemas de software livre — 27 de junho de 2025
Trabalho prático: Bash, git e make — v1.0

1 Introdução

Sua tarefa é implementar 3 programas simples em C++, usando também:

- Git para fazer o controle de versões e trabalho em grupo
- Make para automatizar a compilação dos programas
- Bash para criar e executar casos de teste dos seus programas

Você deve implementar uma solução separada, em C++, para cada um dos problemas:

- Um programa que leia um número inteiro e imprima se ele é primo ("Primo") ou não ("Não é primo") (arquivo `primo.cpp`)
- Um programa que lê um vetor de números inteiros e imprime o maior valor do vetor (arquivo `vetor.cpp`)
- Calculadora de matriz que permita soma, subtração e multiplicação de duas matrizes (arquivo `matriz.cpp`).

O trabalho pode ser feito individualmente ou em duplas.

1.1 O programa `primo.cpp`

O programa deve ler um número inteiro da entrada padrão (teclado) e imprimir na tela "Primo" ou "Não é primo".

```
$ ./primo
4
Não é primo
$

$ ./primo
5
Primo
$
```

Se existir um arquivo `test1-primo.in` com o número 7, você poderia executar seu programa assim:

```
$ ./primo < test1-primo.in
Primo
$
```

1.2 O programa `vetor.cpp`

O programa deve ler um número inteiro n que é o tamanho do vetor, e depois n números que são os elementos do vetor. Então, o programa deve imprimir o maior valor do vetor.

```
$ ./vetor
4
5 3 7 1
7
$

$ cat test1-vetor.in
5
3 9 2 1 5
$ ./vetor < test1-vetor.in
9
$
```

1.3 O programa `matriz.cpp`

O programa deve ler um número inteiro n que é o tamanho das matrizes, e depois ler 2 matrizes $n \times n$. Então, o programa deve ler uma letra que indica a operação:

- `a`dição
- `s`ubtração
- `m`ultiplicação

O programa deve fazer a operação e imprimir a matriz resultante.

```
$ ./matriz
2
2 3
3 3
1 4
6 5
a
3 7
9 8
$
```

2 Usando *make*

Você deve criar um, e apenas um, arquivo chamado `Makefile` para automatizar a compilação dos seus 3 programas.

Você pode criar quantas regras quiser, mas você deve conseguir compilar os 3 programas separadamente. Por exemplo: `make primo` compila apenas o programa `primo.cpp`, gerando o executável `primo`. Você também deve criar uma regra que compile todos os programas que necessitem de uma vez só.

O seu `Makefile` deve compilar os programas usando as opções (*flags*) `-Wall -Wextra`.

3 Usando *git*

Você deve criar um projeto no Github para as suas soluções. Você deve seguir as seguintes práticas:

- Cada grande funcionalidade deve ser uma *branch* diferente, que depois é unificada (*merge*) à *main*. Exemplos: adição e subtração da matriz; multiplicação da matriz; maior valor do vetor; identifica primo.
- Cada *commit* deve ser uma pequena mudança. Exemplo: soma de matrizes; subtração de matrizes; corrige erro ao encontrar maior valor do vetor.
- As mensagens de *commit* devem ser curtas e diretas, e devem começar com um verbo, respondendo à pergunta: "O que essa mudança faz?" Exemplo: "Implementa a multiplicação de duas matrizes".

4 Usando *scripts shell*

Você deve criar 3 scripts para gerar casos de teste para cada programa:

- `generate-test-primo.sh`
- `generate-test-vetor.sh`
- `generate-test-matriz.sh`

Cada script deve gerar uma entrada de teste para o respectivo programa. Você pode guardar o teste em um arquivo para usar com seu programa.

Opcionalmente, você pode criar mais scripts (ou adaptar o seu) para gerar a saída esperada para a entrada aleatória gerada.

Você também pode automatizar outras tarefas, se achar necessário.

5 O que entregar

Você deverá entregar, até 04/julho, um arquivo *.txt* contendo o link para o projeto *público* no Github. O projeto deve conter, no mínimo:

- Os três arquivos C++ com as soluções.
- O arquivo `Makefile`.
- Os scripts usados (arquivos `.sh`)
- Um arquivo chamado `README.md` no formato Markdown contendo o nome completo dos integrantes do grupo e qualquer outra informação que o grupo achar relevante (problemas conhecidos, dificuldades, como usar os scripts, ...).
- Outros arquivos de teste contendo exemplos de entrada e saída que o grupo usou para testar o trabalho.