Pergunta 1

Requer avaliação
Nota: 1,00

Executar uma aplicação num servidor remoto requer que o ambiente de utilizador seja carregado antes que a aplicação possa ser executada.

Crie um script que prepara um ambiente em C:

- verifica se o compilador gcc está instalado e caso nao esteja instala-o
- instala a biblioteca: libhdf5-dev

Assuma que tem permissões para instalar pacotes e que o sistema operativo é Ubuntu

```
!#/bin/bash

if ! command -v gcc $> /dev/null then

sudo apt-get update

sudo apt-get install -y gcc

fi

sudo apt-get update

sudo apt-get install -y libhdf5-dev
```

Pergunta 2

Requer avaliação
Nota: 1,00

Executar uma aplicação num servidor remoto requer que o ambiente de utilizador seja carregado antes que a aplicação possa ser executada.

Crie um script que prepara um ambiente virtual em Python com os seguintes pacotes instalados:

- torch (pytorch)
- matplotlib

Nota: consultar pratica 5

```
#!/bin/bash

python3 -m venv pytorch_environment

source ./pytorch_environment/bin/activate

pip install torch matplotlib
```

Pergunta 1

Requer avaliação
Nota: 1,00

Tendo em conta que a organização de uma matrix em C é "row-major", implemente a soma de duas matrizes de forma a maximizar e minimizar o desempenho da cache, nas funções, soma_fast(...) e soma_slow(...) respectivamente.

Assuma que o tamanho das matrizes é definido pelas constantes N e M.

```
void somar_fast(int op1[ ][M], int op2[ ][M], int soma3[ ][M]) {
    int i, j;
    for(i=0; i<N; i++){
        for(j=0; j<M; j++){
            soma3[i][j]=op1[i][j]+op2[i][j];
        }
    }
}

void somar_slow(int op1[ ][M], int op2[ ][M], int soma3[ ][M]) {
    int i, j;
    for(j=0; j<M; j++){
        for(i=0; i<N; i++){
            soma3[i][j]=op1[i][j]+op2[i][j];
        }
    }
}
```

Q1

```bash
#!/bin/bash
if ! command -v gcc &> /dev/null
then
    sudo apt-get install -y gcc
fi
sudo apt-get install -y libhdf5-dev
```

Q2

```bash
#!/bin/bash
ENV_NAME="pytorch_env"
python3 -m venv $ENV_NAME
source $ENV_NAME/bin/activate
pip install torch matplotlib
```

Q3

```
void somar_fast(int op1[][M], int op2[][M], int soma3[][M]) {
    int i, j;
    for (i = 0; i < N; i++) {          // Itera pelas linhas
        for (j = 0; j < M; j++) {      // Itera pelas colunas
            soma3[i][j] = op1[i][j] + op2[i][j];
        }
    }
}

void somar_slow(int op1[][M], int op2[][M], int soma3[][M]) {
    int i, j;
    for (j = 0; j < M; j++) {// Itera pelas colunas primeiro
        for (i = 0; i < N; i++) {      // Itera pelas linhas
            soma3[i][j] = op1[i][j] + op2[i][j];
        }
    }
}
```